

ZASTOSOWANIE WIELOPOPULACYJNEGO ALGORYTMU EWOLUCYJNEGO DO PROBLEMU WYZNACZANIA ŚCIEŻKI PRZEJŚCIA

Łukasz KUCZKOWSKI, Roman ŚMIERZCHAŁSKI

Politechnika Gdańska, Wydział Elektrotechniki i Automatyki, ul. G. Narutowicza 11/12, 80-233 Gdańsk, tel: 58 348 63-11, e-mail: lkuczowski@ely.pg.gda.pl

Streszczenie: Artykuł przedstawia zastosowanie wielopopulacyjnego algorytmu ewolucyjnego do wyznaczenia ścieżki przejścia. Zastosowany algorytm oparty jest na modelu z częściowo wymienianą populacją oraz migracją między niezależnie ewoluującymi populacjami. W pracy porównano działanie algorytmu wielopopulacyjnego z jednopopulacyjnym oraz zbadano wpływ poszczególnych parametrów na jakość uzyskanego rozwiązania. Wykazano, że zastosowanie kilku niezależnie ewoluujących populacji prowadzi do uzyskania rozwiązania lepszego w stosunku do wariantu jednopopulacyjnego. Jako problem testowy zostało wybrane zagadnienie unikania kolizji na morzu.

Słowa kluczowe: algorytmy ewolucyjne, algorytmy wielopopulacyjne, planowanie ścieżki przejścia.

1. WSTĘP

Jednym ze sposobów rozwiązywania zagadnień optymalizacji jest zastosowanie algorytmu ewolucyjnego z wieloma populacjami. Wariantem takiego algorytmu jest wyspowa metoda genetyczna (DGA), który został po raz pierwszy przedstawiony przez Reiko Tanese [1] jako metoda zrównoleglenia obliczeń w prostym algorytmie genetycznym [2]. Zaproponowane rozwiązanie polegało na podzieleniu globalnej populacji na kilka podpopulacji. Następnie na każdej podpopulacji uruchamiano prosty algorytm genetyczny z wykorzystaniem osobnego procesora, który wykonywał pojedynczą pętlę ewolucji. W każdej kolejnej generacji wymieniano kilka osobników pomiędzy podpopulacjami (synchronicznie bądź asynchronicznie). Przeprowadzone badania wykazały, że tak zdefiniowany algorytm genetyczny uzyskiwał lepsze wyniki pod względem czasu obliczeń oraz jakości rozwiązania w porównaniu z algorytmem jednopopulacyjnym.

Jednym z problemów w których można zastosować algorytmy wielopopulacyjne jest zadanie sterowania obiektem ruchomym (np. robotem mobilnym, autonomicznym pojazdem nawodnym). Polega ono na przeprowadzeniu obiektu własnego z punktu początkowego do danego punktu docelowego lub obszaru wykonywania zadania (misji). W tym celu należy wyznaczyć optymalną ścieżkę przejścia względem określonego kryterium np. najkrótszego czasu dojazdu do punktu docelowego. Ścieżka ta musi omijać przeszkody:

ograniczenia statyczne i dynamiczne środowiska. Zwykle ograniczenia dynamiczne interpretowane są jako inne obiekty ruchome, poruszające się po określonej trajektorii i z pewną prędkością. Zagadnienie może być rozpatrywane w dwóch trybach: off-line i on-line. W trybie off-line poszukiwanie ścieżki przejścia dokonywane jest w środowisku z uwzględnieniem parametrów ruchu innych obiektów, które są niezmiennie. W trybie on-line planowanie ścieżki związane jest ze zmiennością środowiska i niepewnością ruchu pozostałych obiektów. W związku z tym wymagana jest ciągła kontrola zmian w środowisku i parametrów ruchu obiektów. W wypadku ich zmian następuje modyfikacja wcześniej wyznaczonej ścieżki w trybie off-line.

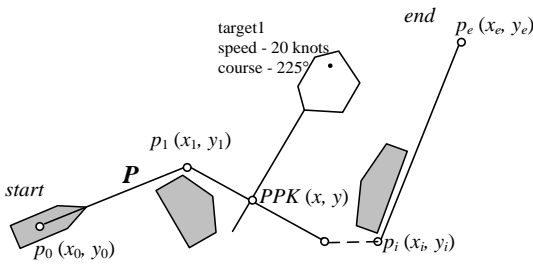
Problem sprowadzono do zadania optymalizacji dynamicznej z ograniczeniami statycznymi (łądy, obszary ograniczenia ruchu, itp.) i dynamicznymi (obce statki, zmienne warunki hydrometeorologiczne) [3]. Do rozwiązania postawionego problemu została zastosowana adaptacyjna metoda ewolucyjna bazująca na systemie wyznaczania ścieżek przejść Evolutionary Planner/Navigator (vEP/N++) [4]. Pracę systemu vEP/N++ oparto o bibliotekę algorytmów ewolucyjnych GALib [5]. Na podstawie dostępnych komponentów biblioteki zbadano oraz porównano działanie wielopopulacyjnego algorytmu ewolucyjnego w ewolucyjnej metodzie poszukiwania ścieżki przejścia w stosunku do algorytmu jednopopulacyjnego.

Artykuł został zorganizowany następująco: w punkcie drugim przedstawiono problem planowania ścieżek przejścia w środowisku morskim, metoda ewolucyjna oraz jej wariant wielopopulacyjny został opisany w części trzeciej, czwartej oraz piątej. Część szósta prezentuje badania symulacyjne. W części końcowej znajduje się podsumowanie.

2. PLANOWANIE ŚCIEŻKI PRZEJŚCIA W SYTUACJI KOLIZYJNEJ

Problem unikania kolizji polega na wyznaczeniu ścieżki przejścia P jako części danej trasy którą pokonuje statek od obecnej pozycji (punktu startowego) (x_0, y_0) do faktycznego punktu końcowego (x_e, y_e) . Ścieżka złożona jest z sekwencji segmentów liniowych p_i ($i = 1, \dots, n$), połączonych między sobą punktami zwrotu (x_i, y_i) . Wybór punktu startowego i końcowego zależy od operatora. Biorąc pod uwagę powyższe, ścieżka P jest dopuszczalna (należy do zbioru ścieżek bezpiecznych), jeżeli każdy jej segment p_i ($i = 1, \dots, n$) pozostaje w granicach

środowiska oraz nie następuje przecięcie z żadnym ograniczeniem, tak statycznym jak i dynamicznym. Ścieżki, których fragment znajduje się na obszarach zakazanych, generowanych przez ograniczenia statyczne i dynamiczne, uważane są za niedopuszczalne [3].



Rys. 1. Potencjalna sytuacja kolizyjna

Gdy napotkany obiekt znajduje się w obszarze obserwacji oraz kiedy jego kurs przecina kurs własnego statku w niebezpiecznej odległości, to obiekt taki określa się jako groźący kolizją (target 1, punkt PPK, rysunek 1). Odległość bezpieczna od statku własnego zależy od przyjętego poziomu stopnia zagrożenia kolizją (najczęściej przyjmuje się odległość 5-8 mil morskich na wprost od dziobu i 2-4 mile morskie za rufą).

W przypadku rozwiązywania problemu w ujęciu ewolucyjnym obiekty stanowiące zagrożenie interpretuje się jako ruchome obszary (obce statki), poruszające się z określoną prędkością i o określonym kształcie.

Zgodnie z planem transportowym statek własny powinien pokonać określoną trasę w zakładanym czasie. Z drugiej strony, musi poruszać się bezpiecznie wzdłuż wyznaczonej trasy w celu unikania obiektów groźących kolizją. Planowanie trasy przejścia statku w sytuacji kolizyjnej musi być kompromisem pomiędzy odchyleniem od przyjętego kursu i bezpieczeństwem żeglugi. Dlatego też problem ten przyjęto definiować jako zadanie optymalizacji wielokryterialnej, w którym bierze się pod uwagę zarówno bezpieczeństwo jak i ekonomię ruchu statku. Każda ścieżka oceniana jest na podstawie funkcji przystosowania (1).

$$Total_Cost(\mathbf{P}) = Safe_Cost(\mathbf{P}) + Econ_Cost(\mathbf{P}) \quad (1)$$

$$Safe_Cost(\mathbf{P}) = w_c * clear(\mathbf{P}) \quad (2)$$

gdzie: $clear(\mathbf{P}) = \max_{i=1}^n c_i$, w_c – współczynnik wagowy, c_i – odległość przeszkody od najbliższego węzła ścieżki \mathbf{P}

$$Econ_Cost(\mathbf{P}) = w_d dist(\mathbf{P}) + w_s smooth(\mathbf{P}) + w_t time(\mathbf{P}) \quad (3)$$

gdzie: w_d, w_s, w_t – współczynniki wagowe

Na ogólny koszt przystosowania ścieżki do środowiska składa się zarówno koszt bezpieczeństwa (2), jak i koszt związany z ekonomią ruchu statku wzdłuż przejętej trasy (3).

Koszt bezpieczeństwa jest obliczany na podstawie odległości wyznaczonej ścieżki od przeszkód ($clear(\mathbf{P})$), natomiast koszt związany z ekonomią ruchu statku uwzględnia: całkowitą długość ścieżki \mathbf{P} ($dist(\mathbf{P})$), maksymalny kąt zwrotu pomiędzy poszczególnymi segmentami p_i ($smooth(\mathbf{P})$), oraz czas potrzebny do przebycia trasy ($time(\mathbf{P})$).

3. ALGORYTM EWOLUCYJNY

Wraz z rozwojem robotyki i technologii komputerowej wykonano szereg prac celem opracowania nowych, efektywnych metod planowania ścieżki przejścia obiektu ruchomego w określonym środowisku. Analizując istniejące metody można zauważyć następujące problemy i trudności, które mają znaczenie dla rozwiązania tego zagadnienia: niemożność otrzymania wyników w czasie zbliżonym do rzeczywistego (co związane jest z długim czasem obliczeń) oraz trudność w dostosowywaniu się do zmiennych kryteriów oceny ścieżki przejścia (dynamiczne zmiany środowiska).

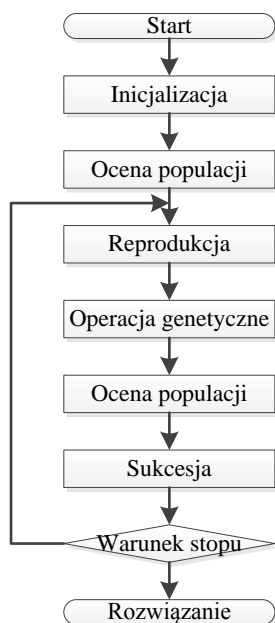
W celu rozwiązania tych problemów podjęto badania nad systemem planowania ścieżek przejść z wykorzystaniem technik ewolucyjnych. Uwzględniając zalety tych technik oraz wykorzystując specjalistyczną wiedzę z dziedziny algorytmów ewolucyjnych, opracowano system EP/N. Ważnym elementem systemu jest: reprezentacja rozwiązań przez unikalną strukturę chromosomu i konstrukcja operatorów genetycznych modyfikujących osobniki. Do modyfikacji ścieżek przejść wykorzystano następujące operatory genetyczne: mutację, mutację miękką, mutację prędkości, dodawanie genu, wymiana pozycji genu, krzyżowanie, wygładzanie, usuwanie genu. Dzięki temu system EP/N nie wymaga dyskretyzacji mapy środowiska poszukiwań, czego wymagają zazwyczaj inne systemy planowania. Obiekty środowiska można zamodelować za pomocą wielokątów zarówno dla przeszkód stałych, wprowadzanych z nośników informacji, jak również obiektów identyfikowanych za pomocą urządzeń pomiarowych. Bazując na systemie EP/N opracowano rozwiniętą jego wersję o nazwie vEP/N++. W systemie uwzględniono dodatkowo obiekty dynamiczne środowiska. Niestacjonarne obiekty zostały scharakteryzowane poprzez wprowadzenie do obliczeń parametru czasu i parametrów ruchu.

Dalsze prace nad programem skupiły się na wprowadzeniu biblioteki algorytmów genetycznych GALib [5]. Zastosowanie biblioteki pozwala w prosty sposób zmieniać parametry algorytmu ewolucyjnego, takie jak skalowanie funkcji przystosowania [6] oraz metody selekcji [7]. W oparciu o wbudowane mechanizmy biblioteki przeprowadzono badania: mechanizmu niszowania [8], wpływu nacisku selektywnego [9], wariantów operatorów genetycznych [10], wariantów algorytmów genetycznych [11] oraz opracowano koncepcję hierarchicznego algorytmu wyznaczania ścieżki przejścia [12].

4. ALGORYTM JEDNOPOPULACYJNY Z CZĘŚCIOWO ZASTĘPOWANĄ POPULACJĄ

Działanie algorytmu ewolucyjnego polega na przetwarzaniu zbioru rozwiązań nazwanych populacją. Środowisko w jakim działa algorytm jest zdefiniowane na podstawie rozwiązywanego problemu (funkcja celu, ograniczenia). Każdy osobnik (pojedynczy element populacji) reprezentuje rozwiązanie problemu. Na podstawie funkcji celu do osobnika przypisuje się wartość nazwaną przystosowaniem, które określa jakość reprezentowanego przez osobnika rozwiązania. W momencie startu algorytmu następuje ustalenie warunków początkowych (inicjalizacja populacji), a każdemu osobnikowi zostaje przypisana losowa wartość. Następnie zostają wykonane etapy: reprodukcji, operacji genetycznych, oceny oraz sukcesji (rys. 2). Reprodukacja polega na losowym wprowadzeniu do populacji tymczasowej, osobników z populacji bazowej. Możliwe jest wprowadzenie więcej niż jednej kopii danego osobnika. Lepsza wartość przystosowania osobnika zapewnia większe prawdopodobieństwo powielenia

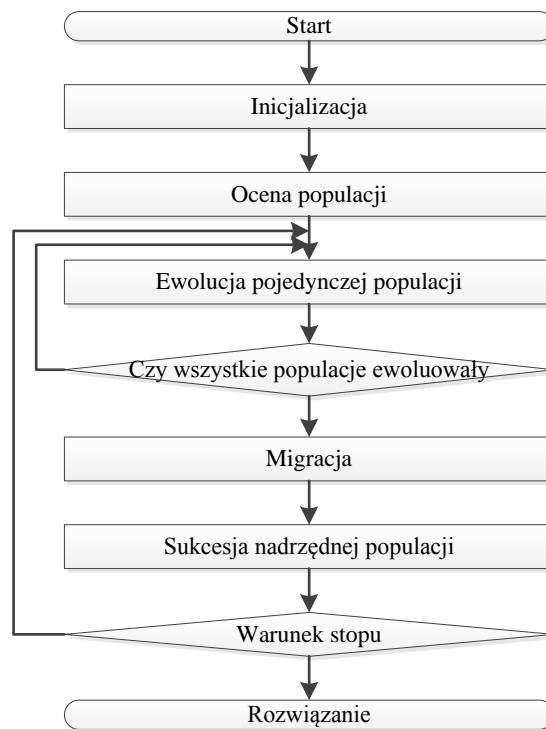
jego kopii. Następnie populacja tymczasowa poddawana jest operacjom genetycznym, które dokonują modyfikacji na osobnikach. Powstały w ten sposób zbiór rozwiązań nazywamy populacją potomną, która podlega ocenie. Etap sukcesji polega na stworzeniu nowej populacji bazowej. W algorytmie jednopopulacyjnym z częściowo zastępowaną populacją nowa populacja bazowa zawiera osobniki z populacji potomnej oraz starej populacji bazowej. Liczba osobników dodawanych do populacji bazowej z populacji potomnej jest ustalana przez użytkownika. Przeniesienie osobnika nastąpi tylko wtedy, kiedy będzie on posiadał lepszą wartość przystosowania niż najgorszy osobnik w starej populacji bazowej. Powyższe fazy algorytmu są powtarzane w pętli do momentu spełnienia warunku zatrzymania (liczba generacji lub dowolnie inaczej zdefiniowany warunek stopu). W bibliotece GALib algorytm przyjmuje nazwę Steady-State GA.



Rys. 2. Schemat działania jednopopulacyjnego algorytmu ewolucyjnego

5. ALGORYTM WIELOPOPULACYJNY

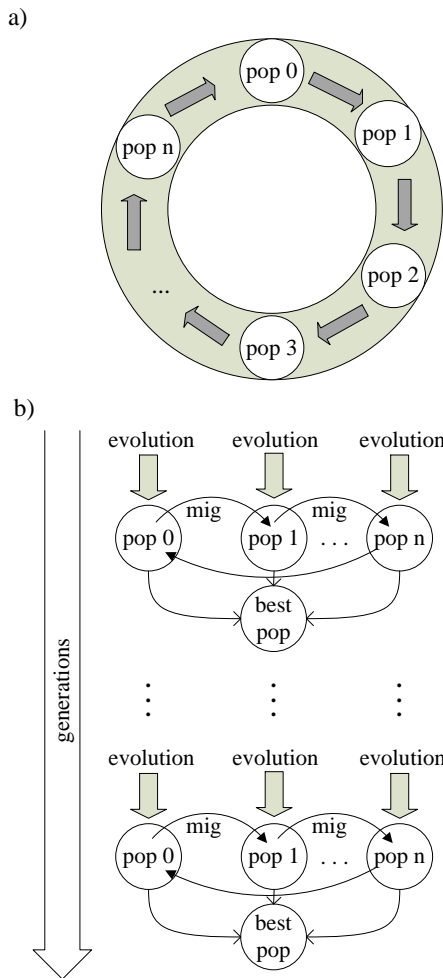
Zastosowanie algorytmu wielopopulacyjnego pozwala na jednoczesną, niezależną ewolucję wielu populacji. W tym celu program vEPN++ wykorzystuje algorytm ewolucyjny z częściowo wymienianą populacją. Schemat ogólny działania algorytmu wielopopulacyjnego został przedstawiony na rysunku 3. W pierwszej fazie działania programu następuje inicjalizacja określonej przez użytkownika liczby populacji przez losowo generowane osobniki. Następnie przeprowadzany jest proces ewolucji dla każdej populacji osobno. Proces ten, podobnie jak w algorytmie jednopopulacyjnym, składa się z następujących etapów: reprodukcja, operacje genetyczne, ocena osobników, selekcja. Po zakończeniu ewolucji wszystkich populacji następują etapy migracji oraz sukcesji nadrzędnej populacji.



Rys. 3. Schemat działania wielopopulacyjnego algorytmu ewolucyjnego

Badany algorytm wykorzystuje migrację typu wyspowego (ang. *stepping-stone*) (rys. 4a) [1]. Z każdej populacji, zaczynając od zerowej, następuje przekazanie do populacji sąsiedniej (kolejnej z rzędu) określonej liczby najlepszych osobników. Proces ten powtarzany jest do momentu, aż ostatnia w kolejności populacja przekaże część swoich osobników do populacji zerowej.

Następnym etapem działania algorytmu jest sukcesja nadrzędnej populacji (rys. 4b). Populacja ta służy do przechowywania najlepszych rozwiązań i nie podlega ewolucji. Z każdej populacji podrzędnej (czyli takiej na której została przeprowadzona ewolucja) zostaje wybrana określona liczba najlepiej przystosowanych osobników. Następnie algorytm sprawdza, czy którykolwiek z wybranych osobników jest lepiej przystosowany niż najgorszy osobnik w populacji nadrzędnej. Jeżeli tak, następuje zastąpienie gorszego osobnika lepszym. Za wynik końcowy przyjmuje się najlepiej przystosowanego osobnika z nadrzędnej populacji.



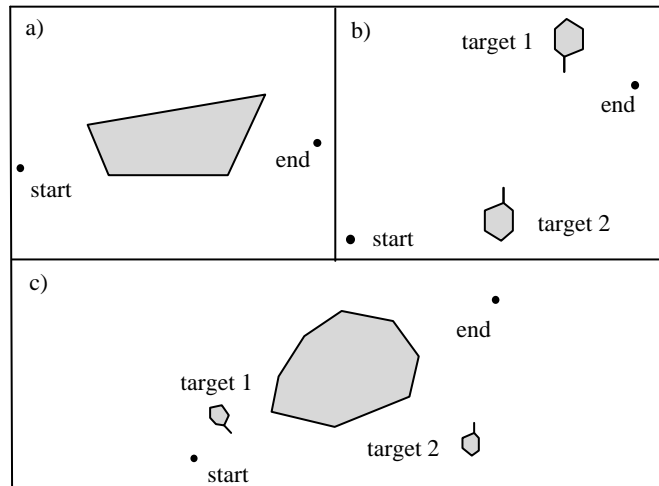
Rys. 4. a) Migracja typu wyspowego, b) sukcesja populacji elitarnej

6. BADANIA SYMULACYJNE

Wielopopulacyjny algorytm ewolucyjny został wykorzystany do rozwiązania problemu wyznaczenia ścieżki przejścia na morzu. Badania algorytmu wymagają dobrania odpowiednich zadań testowych. Zaproponowano 3 środowiska odwzorujące zbliżone do rzeczywistych sytuacje morskie. Zastosowano następujące oznaczenia ψ – kurs, v – prędkość.

Środowisko 1 (rys. 5a) przedstawia problem omijania statycznych obszarów niedozwolonych. Środowisko 2 (rys. 5b) przedstawia problem unikania kolizji z dwoma poruszającymi się w przeciwnych kierunkach obiektami dynamicznymi, które reprezentują obce statki odpowiednio o parametrach: target 1 $\psi - 180^\circ$, $v - 12$ węzłów, target 2 $\psi - 0^\circ$, $v - 8$ węzłów. Środowisko 3 (rys. 5c) jest połączeniem obu poprzednich problemów, zawiera więc przeszkody statyczne oraz dynamiczne odpowiednio o parametrach: target 1 $\psi - 140^\circ$, $v - 8$ węzłów, target 2 $\psi - 0^\circ$, $v - 12$ węzłów.

Celem przeprowadzonych badań było porównanie działania algorytmu SSGA z DGA oraz, dla algorytmu DGA, określenie wpływu liczby populacji oraz migrujących osobników na jakość rozwiązania oraz czas obliczeń. Do symulacji wykorzystano komputer klasy PC z procesorem Intel Core i7 950 3,07 GHz działający pod kontrolą systemu Windows 7.

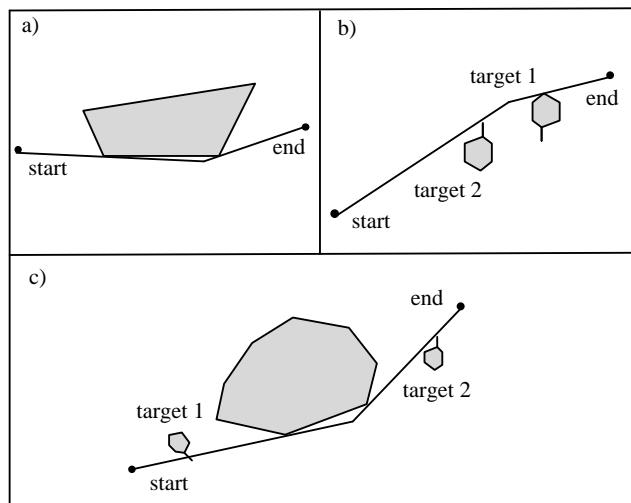


Rys. 5. Środowisko a) 1, b) 2, c) 3

Zestawienie wyników w tabelach zrealizowano w oparciu o rodzaj środowiska oraz populacje początkowe, określone jako a oraz b . Zmianie podlegały jedynie badane parametry. W prezentowanych wynikach zastosowano następujące skróty: SSGA - algorytm genetyczny z częściowo wymienianą populacją (ang. *Steady-State Genetic Alorith*), DGA - wielopopulacyjny algorytm genetyczny (ang. *Deme Genetic Algorithm*), env - rodzaj zastosowanego środowiska, $init$ - wariant zastosowanej populacji początkowej, pop - liczba populacji, mig - liczba migrujących osobników, t - czas obliczeń, $best$ - wartość funkcji przystosowania dla najlepszego osobnika, F - wartość funkcji przystosowania. Niezależnie od symulacji przyjęto następujące parametry algorytmu:

- wielkość pojedynczej populacji - 30 osobników;
- prawdopodobieństwo krzyżowania - 0,8;
- prawdopodobieństwo mutacji - 0,15;
- liczba wymienianych osobników w populacji - 6;
- selektor - ruletka proporcjonalna;
- zakończenie działania algorytmu dla SSGA - 1000 generacji, DGA - 200 generacji;
- początkowa prędkość własna statku - 20 węzłów.

Pozycje obiektów dynamicznych wyświetlane są zawsze dla najlepszego osobnika. Najlepsze rozwiązanie uzyskane dla każdego środowiska w formie graficznej zostało przedstawione na rysunku 6.



Rys. 6. Najlepsze rozwiązanie uzyskane dla każdego środowiska w formie graficznej

6.1. Porównanie algorytmu jedno- i wielopopulacyjnego

W pierwszej fazie badań porównano algorytm SSGA z DGA. W przypadku rozważanych symulacji algorytm DGA realizował obliczenia w oparciu o następujące parametry: $pop = 5$, $mig = 3$. Zastosowanie algorytmu DGA w porównaniu do SSGA spowodowało zwiększenie czasu obliczeń średnio o 2,5 razy. Algorytm DGA, pomimo krótszej ewolucji w porównaniu z SSGA (SSGA - 1000 generacji, DGA - 200 generacji), dokonuje większej liczby obliczeń związanych z wyznaczeniem wartości funkcji przystosowania każdej generacji. Silny nacisk selektywny w algorytmie SSGA powoduje, że w populacji znajduje się dużo kopii najlepszego osobnika. Nie jest więc konieczne ponowne obliczanie jego funkcji przystosowania. Algorytm DGA poprzez zastosowanie kilku niezależnych populacji oraz wymiany osobników pomiędzy nimi, wprowadza większe zróżnicowanie rozwiązań. Wzrasta tym samym prawdopodobieństwo wprowadzenia nowych ścieżek do populacji. Powstałe w ten sposób osobniki wymagają oceny, co prowadzi do wydłużenia czasu obliczeń. Otrzymany czas (średnio 130 s) jest akceptowalny w przypadku problemu unikania kolizji na morzu i mieści się w granicach czasu pozwalającego na podjęcie decyzji przez operatora w warunkach rzeczywistych [4]. Porównując algorytm jednopopulacyjny z wielopopulacyjnym stwierdzono, że niezależnie od zastosowanego środowiska oraz populacji początkowej uzyskano lepsze rozwiązanie w przypadku zastosowania algorytmu DGA (tab. 1). Średnio wynik uległ poprawie o 18%.

Tablica 1. Porównanie SSGA z DGA

env	init	SSGA		DGA $pop = 5, mig = 3$	
		t [s]	best	t [s]	best
1	a	18	236,58	55	192,38
	b	16	216,56	44	199,47
2	a	42	198,89	129	192,68
	b	32	277,02	130	175,85
3	a	54	316,35	217	260,99
	b	59	315,35	218	248,01

6.2. Liczba populacji

Podczas przeprowadzonych symulacji parametr $mig = 3$. Wyniki badań przedstawiono w tablicy 2. Algorytm poprawił lub utrzymywał na stałym poziomie wartości najlepszego rozwiązania wraz ze wzrostem liczby populacji. Zasada ta nie odnosi się do wszystkich przypadków. W symulacjach w których użyto $pop = 5$ otrzymano rozwiązania o najmniejszej wartości przystosowania, a w przypadku symulacji gdzie $pop = 8$ otrzymano wynik o największej wartości funkcji przystosowania. Biorąc pod uwagę powyższe, zwiększenie liczby populacji w znaczny sposób wydłuża czas obliczeń, a nie daje pewności poprawy jakości rozwiązania. Na podstawie przeprowadzonego eksperymentu stwierdzono, że dla badanego problemu zastosowanie $pop = 3$ do 5 daje najlepsze rezultaty.

Tablica 2. Wpływ liczby populacji na algorytm DGA

env	init	DGA $pop = 2, mig = 3$		DGA $pop = 5, mig = 3$		DGA $pop = 8, mig = 3$	
		t [s]	best	t [s]	best	t [s]	best
1	a	23	205,2	55	192,3	75	201,5
	b	24	201	44	199,5	91	197,1
2	a	54	223,1	129	192,7	201	192,9
	b	51	176,4	130	175,9	212	176,5
3	a	82	258,8	217	261	334	272,7
	b	90	255,4	218	248	330	252,6

6.3. Liczba migrujących osobników

Dokonano analizy wpływu liczby migrujących osobników na algorytm DGA. Przeprowadzono symulacje z $mig = 3, 6$ oraz 15 osobników (tab. 3). Wyniki badań nie wskazują aby liczba migrujących osobników miała wpływ na jakość rozwiązania. W przypadku *init b* zarówno dla *env 1* oraz 3 otrzymano podobne wartości funkcji przystosowania. Zastosowanie *init a* daje bardziej zróżnicowane rozwiązania. Mimo poszukiwania zależności nie wykazano trendu przedstawiającego wpływ tego czynnika na otrzymane rozwiązanie. Zauważono, że zwiększanie liczby migrujących osobników wpływa na wygładzenie charakterystyki zbieżności algorytmu. Eliminuje to skokowy charakter zmian wartości najlepiej przystosowanego osobnika.

Tablica 3. Wpływ liczby migrujących osobników na algorytm DGA

env	init	DGA $pop = 5, mig = 3$		DGA $pop = 5, mig = 6$		DGA $pop = 5, mig = 15$	
		t [s]	best	t [s]	best	t [s]	best
1	a	55	192,4	48	203	59	202,5
	b	44	199,5	52	195,1	48	199
2	a	129	192,7	127	163,3	134	163,3
	b	130	175,9	126	175,9	132	176
3	a	217	261	217	308,1	288	240
	b	218	248	234	250,6	223	252,4

6.4 Analiza czasu obliczeń

Następnym etapem badań była analiza czasu działania programu w zależności od środowiska, populacji początkowej, liczby populacji oraz liczby migrujących osobników. Stwierdzono, że czasy obliczeń zależne są od poziomu skomplikowania środowiska i w znacznie mniejszym stopniu od postaci populacji początkowej. Obliczenia dla *env 1*, które zawiera jedynie przeszkodę statyczną, trwają najkrócej. Wydłużenie czasu obliczeń dla *env 2* oraz 3 związane jest z wyznaczeniem położenia przeszkód dynamicznych oraz obliczeniem potencjalnych punktów kolizji.

W tablicy 4 przedstawiono procentowe porównanie czasu obliczeń z wartością funkcji przystosowania najlepszego osobnika. Wszystkie dane przedstawione są w odniesieniu do algorytmu jednopopulacyjnego. Przykładowo, stosując algorytm DGA z $pop = 2$ oraz *init a* (3 wiersz tabeli) dla *env 1*, czas obliczeń zwiększył się o 27%, a rozwiązanie uległo poprawie o 13% względem wyniku otrzymanego przy zastosowaniu SSGA.

Pod względem czasu obliczeń najlepsze rozwiązania uzyskujemy dla algorytmu SSGA. Zastosowanie algorytmu DGA z $pop = 2$ skutkuje średnim polepszeniem rozwiązania o 14% oraz wzrostem czasu obliczeń o 45%. Analogicznie dla $pop = 5$ otrzymujemy polepszenie rozwiązania o 18%, wzrost czasu o 260% oraz dla $pop = 8$ 16% i 450%. Z powyższych danych wynika, że najlepiej zastosować 2 lub 3 populacje. Zwiększenie liczby populacji przynosi wzrost czasu obliczeń niewspółmierny względem poprawy jakości rozwiązania.

Tablica 4. Zależność jakości rozwiązania od czasu obliczeń

init	pop	env 1		env 2		env 3	
		t	best	t	best	t	best
-	-	[%]	[%]	[%]	[%]	[%]	[%]
a	2	28	13,2 8	29	-12,19	52	18,19
b	2	50	7,17	59	36,31	53	19,00
a	5	200	15,7 6	210	12,98	346	14,75
b	5	200	8,65	304	36,50	281	20,62
a	8	317	14,8 3	379	3,03	519	13,81
b	8	469	9,00	561	36,28	459	19,89

7. WNIOSKI

Przeprowadzone badania wykazały, że zastosowanie wielopopulacyjnego algorytmu ewolucyjnego pozwala na poprawę jakości rozwiązania w porównaniu z algorytmem operującym na pojedynczej populacji. Efekt ten otrzymujemy niezależnie od zastosowanej populacji początkowej oraz rodzaju środowiska. Wykazano również, że algorytm DGA wymaga znacznie mniejszej liczby generacji, aby osiągnąć rezultat porównywalny bądź lepszy od algorytmu SSGA. Zwiększanie liczby ewoluowanych populacji przyczynia się do poprawy jakości rozwiązania. Wykazano istnienie krytycznej liczby populacji, której przekroczenie powodowało brak poprawy rozwiązania, a w znaczny sposób zwiększało złożoność obliczeniową problemu. Zmiany współczynnika migracji nie miały wpływu na jakość rozwiązania. Po zestawieniu wyników wszystkich symulacji stwierdzono, że największy wpływ na czas obliczeń ma złożoność zastosowanego środowiska oraz liczby populacji. Uznano również, że stosowanie więcej niż 2 - 3 populacji przynosi niesatysfakcjonującą poprawę rozwiązania w porównaniu ze zwiększeniem czasu obliczeń.

8. BIBLIOGRAFIA

1. Tanese R.: Distributed Genetic Algorithms, Proc. of 3rd Int. Conf. Genetic Algorithms, 1989, s. 432-439, ISBN 1-55860-066-3
2. Goldberg D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Boston: Addison-Wesley Longman Publishing Co., Inc., 1989, ISBN 0201157675

3. Śmierzchalski R.: Trajectory planning for ship in collision situations at sea by evolutionary computation, In Proc. of the IFAC MCMC'97, Brijuni, Croatia, 1997, s. 1-10, ISBN 0080429343
4. Śmierzchalski R., Michalewicz Z.: Modeling of Ship Trajectory in Collision Situations by an Evolutionary Algorithm. IEEE Transaction on Evolutionary Computation, Vol.4, No.3, 2000, s. 227 – 241, ISSN 1089-778X
5. Wall M. GALib: A C++ Library of Genetic Algorithm Components. MIT, 1996
6. Kolendo P., Śmierzchalski R., Jaworski B.: Experimental research on evolutionary path planning algorithm with fitness function scaling for collision scenarios. In: Methods and algorithms in navigation Marine navigation and safety of sea transportation, Gdynia Maritime University, 2011, s. 85-91, ISSN 2083-6473
7. Kolendo P., Jaworski B., Śmierzchalski R.: Comparison of Selection Schemes in Evolutionary Method of Path Planning, LNAI: Computational Collective Intelligence: Technologies and Applications Vol. 6923, Part II, 2011, s. 241-250, ISBN 978-3-642-23937-3
8. Śmierzchalski R., Kolendo P., Kuczkowski Ł., Jaworski B., Witkowska A.: The niching mechanism in the evolutionary method of path planning, LNCS: Artificial Intelligence and Soft Computing, Vol. 7895, Part II, 2013, s. 101-112, ISBN 978-3-642-38609-1
9. Kuczkowski Ł., Śmierzchalski R.: Selection Pressure in the Evolutionary Path Planning Problem, Advances in Intelligent Systems and Computing: Intelligent Systems in Technical and Medical Diagnostics, Vol. 230, 2014, s. 523-534, ISSN 2194-5357
10. Kuczkowski Ł., Kolendo P., Jaworski B., Śmierzchalski R.: Mean Crossover in Evolutionary Path Planning Method for Maritime Collision Avoidance, Scientific Journals Maritime University of Szczecin, nr 30 (102), 2012, s. 70-77, ISSN 1733-8670
11. Śmierzchalski R., Kuczkowski Ł., Kolendo P., Jaworski B.: Distributed Evolutionary Algorithm for Path Planning in Navigation Situation, TransNav - The International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 7, nr 2, 2013, s. 293-300, ISSN 2083-6473
12. Jaworski B., Kuczkowski Ł., Śmierzchalski R.: Extinction Event Concepts for the Evolutionary Algorithms, Przegląd Elektrotechniczny, nr 10b, 2012, s. 252-255, ISSN 0033-2097

APPLICATION OF MULTI-POPULATION EVOLUTIONARY ALGORITHM FOR PATH PLANNING PROBLEM

Key-words: evolutionary algorithms, multi-population algorithm, path planning

In this paper an application of multi-population evolutionary algorithm for path planning problem is presented. A steady-state genetic algorithm is used for evolution of single population. An applied migration mechanism is based on island topology. In the article the comparison of single and multi-population algorithm has been made and the influence of changing algorithm parameters has been studied. It has been shown that using several independently evolving population leads to a better solution than when the single population is used. For a testing problem a sea collision avoiding problem has been chosen.