Autoreferat

1. **Imię i Nazwisko: Aleksander Skakovski.**
2. **Posiadane dyplomy, stopnie naukowe - z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej:**
   - dyplom magistra inżyniera o specjalności „Elektroniczne urządzenia obliczeniowe" uzyskany na Politechnice Kowieńskiej w Kownie na Litwie, w roku 1983,
   - dyplom/stopień doktora nauk technicznych w zakresie informatyki nadany uchwałą Rady Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej z dnia 27 maja 1997r.

     Rozprawa doktorska napisana w języku angielskim pod tytułem:
     „Efektywne i odporne na uszkodzenia strategie odwzorowania wykorzystywane w przetwarzaniu potokowym"
     (*„Efficient and Fault-Tolerant Assignment Strategies for Pipeline Processing"*).
3. **Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych:**
   - 1983 - 1991 - projektant systemów i programów mikroprocesorowych sterujących napędami dysków twardych w Instytucie Badawczym Inżynierii Komputerowej i Informatyki w Wilnie, Litwa,
   - 1997 - 1998 – starszy wykładowca w Zakładzie Technik Informatycznych na Wydziale Oceanotechniki i Okrętownictwa Politechniki Gdańskiej,
   - 1998 - 2000 - adiunkt w Katedrze Informatyki Akademii Morskiej w Gdyni,
   - 2000 - 2018 adiunkt w Katedrze Nawigacji Akademii Morskiej w Gdyni.
   - 2007 – 2019 - prowadzenie wykładów z Badań operacyjnych w Wyższej Szkole Zarządzania w Gdańsku.
   - Od 2019 - adiunkt w Katedrze Systemów Informacyjnych Uniwersytetu Morskiego w Gdyni.

A. Skakowski

4. Wskazanie osiągnięcia* wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2017 r. poz. 1789):

a) tytuł osiągnięcia naukowego:

Population-Based Approaches to the Discrete-Continuous Scheduling
(Zastosowanie metod opartych na populacji do szeregowania dyskretno-ciągłego)

b) (autorzy, tytuł publikacji, rok wydania, nazwa wydawnictwa, recenzenci wydawniczy):

A. Skakovski, *"Population-Based Approaches to the Discrete-Continuous Scheduling"*, 2018, samodzielny autor części II monografii: E. Ratajczak-Ropel and A. Skakovski, "Population-based approaches to the resource-constrained and discrete-continuous scheduling," Part I & Part II, Studies in Systems, Decision and Control, vol. 108, J. Kacprzyk, (red.), 2018, Springer, str. 101-236, DOI: 10.1007/978-3-319-62893-6.

c) omówienie celu naukowego ww. pracy i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania.

Przedłożona do postępowania habilitacyjnego cz. II wyżej wymienionej monografii jest poświęcona ewolucyjnym metodom i algorytmom opartym na populacji do rozwiązywania NP-trudnego problemu szeregowania dyskretno-ciągłego (*Discrete-Continuous Scheduling Problem - DCSP*).

Głównym celem opisanych w monografii badań naukowych było opracowanie efektywnych ewolucyjnych algorytmów opartych na populacji do rozwiązania złożonego obliczeniowo problemu szeregowania dyskretno-ciągłego oraz opracowanie metod wykorzystujących potencjał zawarty w tego typu algorytmach do podniesienia ich efektywności. Osiągnięcie tego celu wymagało eksperymentalnego zbadania czynników i zależności między tymi czynnikami, oraz ustalenia zakresów wartości parametrów, które zapewniłyby podwyższoną efektywność opracowanych algorytmów.

DCSP, który został opisany w rozdziale 8 monografii, jest szczególnym przypadkiem bardziej ogólnego problemu znanego w literaturze jako *Resource-Constrained Project Scheduling Problem* (RCPSP) (problem szeregowania (harmonogramowania) zadań projektu z ograniczonymi zasobami). W praktyce DCSP pojawia się w przemyśle, w sytuacjach gdy zestaw urządzeń działających równolegle jest zasilany ze wspólnego źródła mocy: elektrycznej, hydraulicznej, pneumatycznej. Takimi urządzeniami mogą być urządzenia szlifujące, mieszające, elektrolizery, pompy paliwa lub gazu, piece hutnicze lub młoty hydrauliczne. W tych przykładach urządzenia reprezentują zasób dyskretny (podzielne w sposób dyskretny), a współdzielona moc – zasób ciągły (podzielna w sposób ciągły). W

A. Skakovski

systemach komputerowych, wirtualna stronicowana pamięć operacyjną współdzielona przez wiele procesorów może być traktowana jako zasób ciągły. W systemach przetwarzania sieciowego (*grid computing*), duża ilość procesorów może być traktowana jako zasób ciągły, a rolę zasobu dyskretnego mogą pełnić napędy dysków twardych, drukarki lub urządzenia służące do badań naukowych.

Szybkość wykonania zadania jest zależna od chwilowej nieznanej z góry ilości zasobu ciągłego, przydzielonego do zadania, i jest opisana za pomocą funkcji ciągłej i rosnącej. Problem polega na jednoczesnym przydziale do zadań zasobów dyskretnych i ciągłych w taki sposób, aby zminimalizować dane kryterium szeregowania. W literaturze są rozpatrywane 4 kryteria szeregowania: C_{max} - długość uszeregowania, $\bar{F}$ - średni czas przepływu (czas przebywania zadania w systemie), L_{max} - maksymalne opóźnienie, U_{max} - poziom zużycia zasobu ciągłego. W przedłożonej monografii jako kryterium szeregowania była stosowana długość uszeregowania C_{max}. Ogólna metodyka rozwiązywania DCSP zakłada dekompozycje problemu wyjściowego na dwa powiązane ze sobą pod-problemy: przydziału zasobów dyskretnych (jeden zasób to zbiór m maszyn) oraz przydziału zasobu ciągłego (jeden dodatkowy zasób dostępny w ilości 1). Wyznaczenie uszeregowania optymalnego polegałoby więc na: (i) ustaleniu wszystkich możliwych przydziałów zasobu dyskretnego do zadań oraz (ii) określeniu optymalnego przydziału zasobu ciągłego, do każdego z ustalonych wcześniej przydziałów zasobu dyskretnego. Przydział zasobu ciągłego do danego przydziału zasobu dyskretnego nazywamy rozwiązaniem semi-optymalnym. Rozwiązanie semi-optymalne wyznaczone na etapie drugim, w ogólności, uzyskuje się poprzez rozwiązanie odpowiednio sformułowanego problemu programowania matematycznego. Etap pierwszy metodyki wymaga ustalenia zbioru S wszystkich możliwych przydziałów zasobu dyskretnego do zadań. Moc takiego zbioru S rośnie wykładniczo ze wzrostem liczby zadań. Zasadne jest więc ograniczenie procesu przeszukiwania do jak najmniejszego podzbioru, który zawiera przynajmniej jeden dopuszczalny przydział zasobu dyskretnego odpowiadający uszeregowaniu optymalnemu. Taki zbiór nazywa się potencjalnie optymalnym (POS). Niestety, moc POS również rośnie wykładniczo ze wzrostem liczby zadań. Dla DCSP z wklęsłymi funkcjami szybkości wykonania zadań zostały udowodnione właściwości uszeregowań optymalnych dla kryterium C_{max}, pozwalające na redukcje POS. Podjęto również próbę eliminacji uszeregowań o takiej samej wartości C_{max}, jednak w obu przypadkach moc POS nadal rośnie wykładniczo ze wzrostem liczby zadań. W literaturze są rozpatrywane dwa podstawowe rodzaje funkcji szybkości wykonania zadań: wklęsłe i wypukłe. Chociaż dla funkcji wypukłych uszeregowanie otrzymuje się w sposób łatwy, DCSP z takimi funkcjami szybkości wykonania zadań nie mają większego znaczenia praktycznego. W praktyce najczęściej występują DCSP z wklęsłymi funkcjami szybkości wykonania zadań. W przypadku wklęsłych

A. Skałkowski

funkcji szybkości wykonania zadań długość uszeregowania semi-optymalnego można obliczyć analitycznie. W tym celu często wykorzystuje się specjalizowane solvery.

Ze względu na złożoność wyznaczenia rozwiązania semi-optymalnego, jak i długi czas obliczeń solvera stosuje się dyskretyzacje zasobu ciągłego. W tym podejściu zakłada się, że ilość zasobu ciągłego jest dzielona na pewne przydziały, odpowiadające trybom wykonania zadania. Na skutek dyskretyzacji zasobu ciągłego ilość trybów, jak i ilości zasobu ciągłego odpowiadające tym trybom są z góry znane. Zadanie wykonywane w takim trybie jest wykonywane przy użyciu stałej, niezerowej ilości zasobu ciągłego. Należy określić tryby wykonania zadań oraz ustalić sekwencje tych zadań na maszynach tak, aby minimalizowane było zadane kryterium szeregowania, a chwilowe zużycie zasobów przez wykonywane zadania było dopuszczalne. Przy takich założeniach otrzymany problem, oznaczony jako Θ_Z, jest szczególnym przypadkiem *Multi-Mode Resource-Constrained Project Scheduling Problem* (MRCPSP) i jest NP-trudny. Wszystkie przeprowadzone przeze mnie i opisane w monografii badania polegały na opracowaniu efektywnych ewolucyjnych algorytmów do rozwiązywania problemu Θ_Z.

W podrozdziałach 9.1 – 9.2 przedłożonej monografii został przedstawiony przegląd aktualnego stanu badań nad problemami szeregowania-dyskretno-ciągłego. Szczególny przypadek DCPS dotyczący minimalizacji zużycia zasobu ciągłego został omówiony w podrozdziale 9.5. W podrozdziale 9.6, natomiast, został opisany problem współdzielenia zasobu ciągłego (CRSharing). Podrozdziały 9.3 – 9.4 zawierają przegląd algorytmów heurystycznych oraz meta-heurystycznych do rozwiązywania problemów szeregowania dyskretno-ciągłego opracowanych przez innych badaczy. Ponieważ moje badania dotyczyły wielopopulacyjnych algorytmów ewolucyjnych (a w szczególności modelu wyspowego) oraz ich efektywności, podrozdział 9.7 zawiera przegląd badań nad modelem wyspowym, a podrozdział 9.8 – przegląd prac o zapobieganiu zjawisku przedwczesnej zbieżności w algorytmach ewolucyjnych i genetycznych.

Ze względu na złożoność obliczeniową problemu Θ_Z, do jego rozwiązania zostały zaproponowane ewolucyjne algorytmy wielopopulacyjne. Ponieważ algorytmy te były postrzegane jako główne narzędzie do rozwiązywania NP-trudnego DCSP, cały wysiłek badawczy został podzielony między dwa główne kierunki: opracowanie nowych efektywnych wielopopulacyjnych algorytmów ewolucyjnych rozwiązujących ten problem oraz zwiększenie efektywności tych algorytmów poprzez określenie czynników i najbardziej korzystnych wartości parametrów o niej decydujących.

W rozdziale 10 monografii zostały opisane wielopopulacyjne algorytmy ewolucyjne opracowane do rozwiązywania problemu Θ_Z: Island-Based Evolutionary Algorithm (IBEA) w

A. Skakowski

podrozdziale 10.1, Population Learning Algorithm (PLA) w podrozdziale 10.2, Cross-Entropy-Based Population Learning Algorithm (PLA2) w podrozdziale 10.3, Population Learning with Differential Evolution Algorithm (PLA3) w podrozdziale 10.4, Island-Based Differential Evolution Algorithm (IBDEA) w podrozdziale 10.5. Struktura wszystkich tych algorytmów opiera się o kanoniczny model wyspowy, który często wykorzystuje się w systemach rozproszonych w celu zwiększenia efektywności przetwarzania. Kanoniczny model wyspowy zakłada podział ogólnej populacji na wyspy o jednakowych rozmiarach (tej samej ilości osobników/rozwiązań na wyspie), na których działa ten sam algorytm. Takie wyspy można określić jako homogeniczne. Wyspy współpracują ze sobą według określonego schematu określonego przez topologię połączeń, wymieniając się osobnikami zgodnie z określoną polityką migracji. Po spełnieniu kryterium stopu algorytm podaje najlepsze pośród wszystkich wysp rozwiązanie jako rozwiązanie problemu. Chociaż wszystkie algorytmy omawiane w monografii są oparte na kanonicznym modelu wyspowym, jednak są między nimi pewne różnice. IBEA był zbudowany na homogenicznych wyspach ulokowanych na skierowanym pierścieniu, które realizowały ten sam ewolucyjny algorytm. Dzięki zastosowanemu w IBEA modelu wyspowemu udało się znacząco zmniejszyć średni błąd znajdywanych rozwiązań względem najlepszych znanych. Średni błąd względny rozwiązań znalezionych przez IBEA był kilkakrotnie, a w niektórych przypadkach nawet ponad dwudziestokrotnie, mniejszy niż średni błąd względny rozwiązań znalezionych przez algorytm genetyczny bazujący na pojedynczej populacji. Jednak wadą IBEA był zbyt duży maksymalny błąd względny w porównaniu do jednopopulacyjnego algorytmu genetycznego, co uwarunkowało opracowywanie bardziej efektywnego algorytmu, którym był PLA. W PLA zostało wykorzystane nowatorskie podejście do ewolucji populacji naśladujące społeczny system edukacji oraz została wdrożona idea heterogeniczności wysp poprzez wprowadzenie jednej dodatkowej wyspy, na której był wykonywany algorytm przeszukiwania tabu. Zastosowanie koncepcji heterogeniczności wysp w PLA poprzez wprowadzenie tylko jednej heterogenicznej wyspy wywołało efekt synergii, co w konsekwencji spowodowało polepszenie 165 (55%) z 300 najlepszych znanych rozwiązań. Sukces PLA określił kierunek dalszych badań nad rozwojem algorytmów wyspowych. Na podstawie PLA powstał PLA2, w którym w celu zwiększenia efektu synergii została dodana kolejna heterogeniczna wyspa. Na wyspie tej były generowane rozwiązania początkowe za pomocą algorytmu entropii skrośnej (Cross-Entropy). Właśnie tym zmianom PLA2 zawdzięcza polepszenie 241 (80,33%) z 300 najlepszych znanych rozwiązań. W algorytmie PLA3, który był następnikiem PLA2, dalej była wykorzystywana idea synergii heterogenicznych wysp. Oprócz kolejnej dodanej wyspy realizującej ewolucję różniczkową (Differential Evolution), dotychczasowa topologia połączeń bazująca na skierowanym pierścieniu została zamieniona na topologię losową, tzn. taką, w której migracja osobników odbywała się między losowo dobieranymi parami wysp. Te zmiany pozwoliły PLA3 na polepszenie 123 (41%) z 300

A. Skakowski

najlepszych znanych rozwiązań. Chociaż każdy kolejny algorytm wyspowy bazujący na heterogenicznych wyspach polepszał najlepsze znane rozwiązania, jednak ułomnością tych algorytmów pozostawał kilkunastoprocentowy maksymalny błąd względny. W IBDEA, natomiast, został zrealizowany homogeniczny model wyspowy bazujący na topologii losowej, w którym na każdej wyspie był realizowany ten sam algorytm ewolucji różniczkowej. IBDEA był zbudowany na 19 wyspach, po 200 osobników na każdej wyspie. Chociaż IBDEA nie polepszył najlepszych znanych rozwiązań w tak znaczący sposób jak jego poprzedniki, jednak był lepszy od nich pod względem maksymalnego błędu względnego. Orientacyjne ilości wysp, których użyto do zbudowania opisanych wyżej algorytmów w zależności od algorytmu wahały się w przedziałach: 12-27 wysp (w tym do trzech wysp heterogenicznych), po 24-200 osobników na każdej wyspie. Aby znaleźć rozwiązanie problemu każdy z wymienionych algorytmów wykonywał 720 000 ewaluacji funkcji przystosowania.

Wszystkie zaproponowane algorytmy były poddane ocenie w celu wyłonienia najbardziej z nich efektywnego. W podrozdziale 11.1 monografii została przeprowadzona ich ewaluacja za pomocą nieparametrycznego statystycznego testu rang Friedmana. Test ten udowodnił hipotezę, że nie wszystkie badane metaheurystyki są jednakowo efektywne, niezależnie od rozwiązywanych instancji problemu. Dodatkowo został obliczony współczynnik Kendalla (*W* Kendalla) do ewaluacji stopnia zgodności pomiędzy uzyskanymi w teście Friedmana ocenami. Otrzymana wartość *W* Kendalla wskazała na średni stopień zgodności pomiędzy uzyskanymi ocenami. Chociaż taka wartość *W* Kendalla nie pozwala na wiarygodne wnioskowanie o wyższości jednych metaheurystyk nad innymi, jednak średnie wartości rang, obliczone w teście Friedmana dla testowanych algorytmów, pozwoliły wyłonić dwie grupy algorytmów pod względem ich efektywności. Do grupy bardziej efektywnych algorytmów weszły (w nawiasach są podane średnie rang): PLA3 (4,65), IBDEA (4,45), DEA (4,4), do grupy mniej efektywnych: PLA2 (3,58), PLA (2,28), IBEA (1,64).

Podczas eksperymentów nad opracowanymi algorytmami nieraz powstawał problem strojenia (tuningu) algorytmu, tzn. wyboru takiej struktury i takich ilości wysp, ilości osobników na wyspie, częstotliwości migracji oraz rozmiaru migracji, które by zapewniły wysoką efektywność algorytmów. Okazało się że kwestia ta wymaga osobnych badań, dlatego zostały zainicjowane gruntowne eksperymenty obliczeniowe mające na celu ustalenie najbardziej korzystnej pod względem efektywności struktury algorytmu oraz wartości badanych parametrów.

W rozdziale 11 monografii został przedstawiony kolejny etap prac, na którym były przeprowadzone eksperymentalne badania nad efektywnością opracowanych algorytmów oraz ustalenie czynników wpływających na ich efektywność. Badania, które miały na celu

A. Sleakowicz

określenie najkorzystniejszej pod względem efektywności struktury algorytmu (rodzaj wysp oraz topologia połączeń między wyspami) oraz polityki migracji osobników między wyspami, zostały opisane w rozdziale 11.2 monografii. Badania nad strukturą algorytmów wyspowych były przeprowadzone na 6 wersjach PLA2 o różnej budowie z wyspami homogenicznymi i heterogenicznymi. Uzyskane wyniki jednoznacznie wskazały na topologię losową jako najbardziej korzystną pod względem efektywności. Z tego powodu topologia losowa została wykorzystana w algorytmach PLA3 i IBDEA.

Badania nad parametrami decydującymi o efektywności kanonicznego modelu wyspowego były przeprowadzone na algorytmie IBDEA bazującym na wyspach homogenicznych, na których była wykonywana ewolucję różniczkową. W celach porównawczych, efektywność IBDEA dla różnych wartości parametrów była porównywana do efektywności algorytmu jednopopulacyjnego DEA, również wykonującego ewolucję różniczkowa. Eksperymenty nad takimi podstawowymi parametrami kanonicznego modelu wyspowego jak ilość wysp K, rozmiar populacji na wyspie x_P, częstotliwość migracji osobników między wyspami ex były przeprowadzone w znacznie szerszym zakresie niż dotychczas zgłaszane w literaturze na ten temat. Dla przykładu, ilość wysp K wahała się w przedziale od 2 do 2150 wysp, rozmiar populacji na wyspie x_P wahał się od 5 do ponad 14000 osobników, a migracja osobników odbywała się z częstotliwością co 1 do 17 pokoleń wygenerowanych na każdej wyspie. Dla przyjętych wartości parametrów K oraz x_P ilość osobników w ogólnej populacji IBDEA wynosiła od 10 do ponad 28000. Oba algorytmy, IBDEA i DEA, wykonywały 10^6 ewaluacji funkcji przystosowania. W celu zapewnienia wiarygodności i miarodajności wyników każdy z algorytmów był testowany na zestawie z 54 instancji problemu Θ_Z. Sumaryczna wartość funkcji przystosowania uzyskana dla takiego zestawu była obliczana dla każdej pojedynczej wartości każdego z badanych parametrów. Oprócz tego, testowane algorytmy wykorzystywały generator liczb pseudolosowych, który zawsze był inicjowany za pomocą tego samego ziarna. W taki sposób jedynym czynnikiem, który mógł zmienić końcowe rozwiązanie algorytmu była wartość badanego parametru. W wyniku przeprowadzonych eksperymentów udało się ustalić wartości parametrów, dla których efektywność testowanych algorytmów wyraźnie wzrastała. Okazało się, że wbrew panującej w literaturze opinii o przewadze modelu wielopopulacyjnego nad jednopopulacyjnym pod względem efektywności są jednak pewne wartości badanych parametrów, dla których DEA działający na pojedynczej populacji wykazywał taką samą efektywność jak IBDEA działający na wielu populacjach. Właściwość ta przemawia za wyborem implementacji jednopopulacyjnej zamiast wielopopulacyjnej jako łatwiejszej w realizacji w sytuacjach, gdy model wyspowy nie jest narzucony przez specyfikę zadnia obliczeniowego. Przewaga implementacji jednopopulacyjnej nad wielopopulacyjną polega na tym, że projektant/programista jest

A. Skakowski

zwolniony z pracy nad wyborem najkorzystniejszej topologii połączeń między wyspami, organizacją migracji rozwiązań oraz ustaleniem polityki migracyjnej. Ustalenia te są potrzebne do zapewnienia wysokiej efektywności algorytmu wielopopulacyjnego i wymagają przeprowadzenia dodatkowych testów, które zazwyczaj są praco i czasochłonne. Oprócz tego, jednopopulacyjny algorytm oszczędnej korzysta z zasobów komputerowych, potrzebuje co najwyżej jednej jednostki obliczeniowej i może zajmować mniej pamięci niż algorytm wielopopulacyjny.

Kolejne badania nad podwyższeniem efektywności algorytmów ewolucyjnych skupiły się na zapobieganiu zjawisku przedwczesnej zbieżności oraz wykorzystaniu możliwości jakie daje różnorodność rozmiarów populacji, na których te algorytmy działają.

Zjawisko przedwczesnej zbieżności pojawia się gdy populacja zostaje wypełniona przez bardzo podobne albo identyczne rozwiązania, a operatory ewolucyjne już nie potrafią wyprodukować różniącego się od nich rozwiązania. W takiej sytuacji proces ewolucji więźnie w obszarze optimum lokalnego bez perspektywy przeniesienia się do innych obszarów przeszukiwań. Aby rozwiązać problem przedwczesnej zbieżności stosuje się metody i techniki podtrzymujące różnorodność rozwiązań populacji na odpowiednim poziomie. W podrozdziale 11.4 przedłożonej monografii została opisana procedura decloning'u zaproponowana do utrzymania różnorodności populacji na pewnym poziomie, co w konsekwencji przyczyniło się do zwiększenia efektywności algorytmu ewolucyjnego. Działanie procedury decloning'u polegało na cyklicznym zastępowaniu „klonów" (takich samych, albo bardzo podobnych rozwiązań) w populacji nowymi losowo wygenerowanymi rozwiązaniami. Największe polepszenie efektywności jednopopulacyjnego algorytmu DEA realizującego ewolucję różniczkową (około 7%) było obserwowano gdy algorytm działał na niedużych populacjach (20 rozwiązań).

Dalsze badania nad czynnikami decydującymi o efektywności algorytmów ewolucyjnych dotyczyły rozmiaru populacji, na której działa algorytm oraz ilości ewaluacji funkcji przystosowania. Przeprowadzone testy pozwoliły ustalić przedział rozmiarów populacji, dla których algorytm wykazywał najwyższą efektywność. Tak na przykład jakość znajdywanych rozwiązań można było polepszyć o ponad 9% zwiększając rozmiar populacji z 20 osobników do 1000. Podobny efekt można było uzyskać dla niektórych rozmiarów populacji zwiększając ilość wykonanych przez algorytm ewaluacji funkcji przystosowania. Tak algorytm działający na populacji o 1000 osobnikach polepszył jakość rozwiązań o ponad 8% gdy ilość wykonanych ewaluacji funkcji przystosowania zwiększyła się z 37800 do 720000. Natomiast gdy populacja składała się z 20 oraz 50 osobników zwiększenie ilości wykonanych przez algorytm ewaluacji funkcji przystosowania nie dało żadnego efektu.

A. Skakowski

Ustalony eksperymentalnie sposób w jaki efektywność jednopopulacyjnego algorytmu DEA zależy od takich czynników jak rozmiar populacji, ilość wykonanych ewaluacji funkcji przystosowania oraz zróżnicowanie populacji, pozwoliły na opracowanie metody mającej na celu zwiększenie efektywności DEA. Metoda ta wykorzystuje fakt iż efektywność algorytmu jest zależna od rozmiaru populacji oraz dostępnej ilości ewaluacji funkcji przystosowania. Dlatego rozmiar populacji powinien być dopasowany do dostępnej ilości ewaluacji funkcji przystosowania. Wybór najbardziej korzystnego pod względem efektywności rozmiaru populacji opiera się na uzyskanych w eksperymentach krzywych wartości funkcji przystosowania dla różnych rozmiarów populacji określonych względem ilości ewaluacji tej funkcji. Oprócz tego metoda zakłada możliwość skrócenia czasu działania algorytmu poprzez zmianę rozmiaru populacji w trakcie wykonywania algorytmu. Przyśpieszenie może być osiągnięte poprzez ustalenie przedziałów ilości ewaluacji funkcji przystosowania i odpowiadającym tym przedziałom najbardziej korzystnych pod względem efektywności rozmiarów populacji. W taki sposób w każdym momencie swojego funkcjonowania algorytm będzie działał na populacji, rozmiar której będzie najbardziej korzystny pod względem jego efektywności. Oprócz tego, jeżeli dopuszczalne jest tolerowanie nieznacznego pogorszenia jakości wyników czas działania algorytmu może być skrócony jeszcze bardziej. Według wstępnych szacunków, zaproponowana metoda w zależności od przyjętych założeń może przyśpieszyć znalezienie rozwiązania o około 1,29 razy, a gdy jest dopuszczalne pogorszenie jakości wyników rzędu 0,48%, nawet do 2,5 razy. W celu zapewnienia wiarygodności i miarodajności wyników DEA był testowany na zestawie z 54 instancji problemu Θ_Z. Średnia sumaryczna wartość funkcji przystosowania dla takiego zestawu była obliczana z 200 przebiegów algorytmu.

Najistotniejsze osiągniecia naukowe przedstawione w monografii (będącej głównym osiągnięciem naukowym):

1. Efektywne rozwiązanie złożonego obliczeniowo NP-trudnego problemu szeregowania dyskretno-ciągłego. Polepszono ponad 80% najlepszych znanych rozwiązań.
2. Ustalenie czynników mających wpływ na efektywność algorytmów opartych na populacji oraz jakość znajdywanych przez te algorytmy rozwiązań.
3. Opracowanie efektywnych ewolucyjnych wielopopulacyjnych algorytmów rozwiązujących problem szeregowania dyskretno-ciągłego z wyłonieniem grupy najbardziej efektywnych wśród nich.
4. Propozycja metody na podwyższenie efektywności ewolucyjnych algorytmów jedno i wielopopulacyjnych, z możliwością kilkakrotnego skrócenia czasu znalezienia rozwiązania.

A. Skakowski

5. Określenie na drodze eksperymentu zakresów wartości parametrów zapewniających podwyższoną efektywność algorytmów opartych na populacjach. Eksperymenty zostały przeprowadzone w sposób bardziej gruntowny i w znacznie szerszym zakresie niż dotychczas zgłaszano w literaturze na ten temat.
6. Potwierdzenie na drodze eksperymentu możliwości uzyskania efektu synergii poprzez wykorzystanie heterogeniczności populacji w algorytmach wielopopulacyjnych.
7. Ustalenie najbardziej efektywnej topologii połączeń między populacjami w algorytmach opartych na kanonicznym modelu wyspowym.
8. Propozycja procedury decloning'u zapobiegającej przedwczesnej zbieżności, a przez to zwiększającej efektywność ewolucyjnych algorytmów jedno i wielopopulacyjnych.
9. Ustalenie na drodze eksperymentu warunków równorzędnej efektywności implementacji jedno i wielopopulacyjnej, ze wskazaniem na wybór w tych warunkach implementacji jednopopulacyjnej jako łatwiejszej w realizacji i wymagającej mniej zasobów obliczeniowych.

Omówienie ewentualnego wykorzystania uzyskanych wyników

Wyniki uzyskane w przedstawionych w monografii badaniach mogą być wykorzystane na dwa sposoby:

- W przemyśle efektywne rozwiązanie problemu szeregowania dyskretno-ciągłego może być wykorzystane do usprawnienia procesu technologicznego oraz zoptymalizowania korzystania z zasobów produkcyjnych. W sytuacjach gdy proces technologiczny polega na wielokrotnym wykonaniu tych samych operacji przez dłuższy czas szczególnie ważne jest ustalenie takiego przydziału operacji do maszyn oraz podziału mocy między tymi maszynami, który zapewni oszczędne wykorzystanie i maszyn i mocy. W komputerowych systemach przetwarzania sieciowego (*grid computing*) z duża ilością procesorów rozwiązanie problemu szeregowania dyskretno-ciągłego może służyć do zoptymalizowanego wykorzystania jednostek obliczeniowych. W takich systemach procesory mogą być traktowane jako zasób ciągły, a rolę zasobu dyskretnego mogą pełnić napędy dysków twardych, drukarki lub urządzenia służące do badań naukowych.
- Zaproponowane rozwiązania i ustalenia dotyczące funkcjonowania i właściwości jedno i wielopopulacyjnych algorytmów ewolucyjnych poszerzają wiedzę na ich temat i mogą być wykorzystane jak do zwiększenia efektywności już istniejących, tak i do opracowywania nowych bardziej efektywnych algorytmów. Uzyskane wyniki określają czynniki i zachodzące między nimi zależności oraz wskazują wartości parametrów, które zapewniają zwiększoną efektywności tych algorytmów. Wartości tych parametrów można wykorzystać w celu zoptymalizowanego wykorzystania dostępnych zasobów obliczeniowych, co w konsekwencji prowadzi do poprawienia efektywności algorytmów. Na podstawie

A. Skakowski

uzyskanych podczas badań danych projektant algorytmów będzie mógł wybrać odpowiedni do istniejących warunków obliczeniowych typ algorytmu (jedno lub wielopopulacyjny) oraz dopasować ilość osobników w populacji, jak i ilość populacji w taki sposób, aby zaprojektowany algorytm wykazał wysoką efektywność. Szczególnie obiecującym obszarem zastosowań uzyskanych wyników mogą być systemy rozproszone, wieloprocesorowe, agentowe oraz systemy przetwarzania sieciowego (*grid computing*).

5. Omówienie pozostałych osiągnięć naukowo - badawczych.

Przed uzyskaniem stopnia doktora.

Przed uzyskaniem stopnia doktora moja praca naukowa dotyczyła problemu odwzorowania zadań (*Assignment Problem*) na systemy potokowe, które są szczególnym przypadkiem systemów komputerowych równoległych bądź rozproszonych. Problem odwzorowania jest szczególnym przypadkiem bardziej ogólnego problemu szeregowania i różni się od ostatniego tym, że dotyczy jedynie przydziału zadań (programów) do procesorów, podczas gdy w problemie szeregowania dodatkowo określa się momenty wykonania zadań. W problemie odwzorowania programy mogą komunikować między sobą jeden lub więcej razy w trakcie ich wykonywania, a na kolejność ich wykonania są nałożone ograniczenia kolejnościowe, które są reprezentowane poprzez skierowany ważony acykliczny graf przepływu danych. Topologia połączeń komunikacyjnych między procesorami również jest przedstawiana w postaci grafu. Problem odwzorowania polega na tym, aby znaleźć taki przydział programów do procesorów, który minimalizuje dane kryterium optymalizacji. Kryterium optymalizacji może być: minimalne koszty komunikacji między procesorami, minimalne koszty wykonania programów i komunikacji między procesorami, zrównoważone obciążenie procesorów i inne kryteria. Gdy na wykonanie programów oprócz minimalnego czasu wykonania dodatkowo nakłada się wymagania dotyczące odporności na uszkodzenia, wówczas ma się do czynienia z odwzorowaniem tolerującym błędy. W tym przypadku, są tworzone kopie programów, które następnie są odwzorowywane na procesory. W przypadku ogólnym problem odwzorowania jest NP-trudny. Z tego powodu zostały zaproponowane efektywne heurystyczne algorytmy wykonujące odwzorowanie z kryterium minimalnego czasu wykonania aplikacji oraz tolerujące uszkodzenia procesorów. Wyniki badan naukowych nad problemem odwzorowania zostały opublikowane oraz wygłoszone na międzynarodowych i krajowych konferencjach naukowych i są wymienione w załączniku 6, punkcie II E (prace E15-E20). Zwieńczeniem prac naukowych nad tym problemem była rozprawa doktorska napisana w języku angielskim.

Po uzyskaniu stopnia doktora

Po uzyskaniu stopnia doktora moja praca naukowa dotyczyła problemów szeregowania następujących rodzajów: szeregowanie programów tolerujących błędy na systemach

A Skalkowski

wieloprocesorowych, szeregowanie projektów z aktywnościami wykonywanymi w wielu trybach i z ograniczeniami zasobów (*Multi-Mode Resource-Constrained Project Scheduling Problem with Minimal and Maximal Time Lags* – MRCPSP/max), oraz szeregowanie dyskretno-ciągłe (*Discrete-Continuous Scheduling Problem* - DCSP).

Programy tolerujące błędy były rozpatrywane w wersji wielowariantowej i wieloprocesorowej. Ponieważ problemy szeregowania tego typu należą do klasy problemów NP-trudnych do ich rozwiązania zostały opracowane efektywne algorytmy ewolucyjne oparte na populacji, które rozwiązywały te problemy w sposób przybliżony. Wyniki prac naukowych nad tymi problemami zostały opublikowane w czasopismach zagranicznych, w tym znajdujących się w bazie Journal Citation Reports (JRC), lub wygłoszone na międzynarodowych i krajowych konferencjach naukowych. Prace z tej tematyki zostały wymienione w załączniku 6, punkcie II A (prace w JCR: A6 – A8) oraz II E (prace E8 - E13, w tym prace E9, E11 są zawarte w bazie Web of Science).

Problem szeregowania MRCPSP/max dotyczy zoptymalizowanego przydziału czynnościom składającym się na projekt ograniczonych zasobów z uwzględnieniem ograniczeń czasowych nakładanych na wykonywanie tych czynności. Zależności między czynnościami są przedstawiane w postaci acyklicznych sieci, np. *Activity-on-Node Network* - AoN. Każda czynność może być wykonana w jednym z wielu trybów, które są określane przez dostępne ilości zasobów przydzielanych do czynności. Celem jest określenie przydziału trybów do czynności oraz czasów rozpoczęcia czynności, tak aby wszystkie ograniczenia zostały spełnione, a czas trwania projektu został zminimalizowany. Do rozwiązania tego NP-trudnego problemu w sposób przybliżony został zaproponowany algorytm uczenia populacji oparty na entropii skrośnej (*A Cross-Entropy Based Population Learning Algorithm – PLA2*). Artykuł zawierający opis wykonanych badań został opublikowany w czasopiśmie z bazy JCR (załącznik 6, punkt II podpunkt A, praca A4).

Kolejne prace naukowe dotyczyły problemu szeregowania dyskretno-ciągłego z dyskretyzacją zasobu ciągłego. Problem ten został opisany wyżej w punkcie 4c Autoreferatu. Do rozwiązania tego NP-trudnego problemu zostały zaproponowane algorytmy ewolucyjne oparte na populacji. Prace zawierające opisy algorytmów zaproponowanych do rozwiązania tego problemu zostały opublikowane w czasopismach znajdujących się w bazie JCR: załącznik 6, punkt II podpunkt A, prace A1, A2, A3, A5, lub w wydawnictwach zagranicznych indeksowanych w Web of Science: załącznik 6, punkt II podpunkt E, prace E1, E2, E4, lub wygłoszone na międzynarodowych konferencjach naukowych indeksowanych w Web of Science: załącznik 6, punkt II podpunkt E, prace E3, E5, E7.

A. Skakowski