

**RADA NAUKOWA DYSCYPLINY
INFORMATYKA TECHNICZNA I TELEKOMUNIKACJA POLITECHNIKI WARSZAWSKIEJ**

zaprasza na
PUBLICZNĄ OBRONĘ ROZPRAWY DOKTORSKIEJ

mgr. inż. Leszka Ciopińskiego

która odbędzie się w dniu **15 czerwca 2023 roku**, o godzinie **14:30** w trybie stacjonarnym

Temat rozprawy:

„Synteza samoadaptacyjnych systemów czasu rzeczywistego za pomocą rozwojowego programowania genetycznego”

Promotor: dr hab. inż. Roman Stanisław Deniziak – Politechnika Świętokrzyska

Recenzenci: dr hab. inż. Rafał Różycki – Politechnika Poznańska

prof. dr hab. inż. Andrzej Obuchowicz – Uniwersytet Zielonogórski

Obrona odbędzie się **w sali nr 116** Gmachu Elektroniki Politechniki Warszawskiej (Warszawa, ul. Nowowiejska 15/19).

Z rozprawą doktorską i recenzjami można zapoznać się w Czytelni Biblioteki Głównej Politechniki Warszawskiej, Warszawa, Plac Politechniki 1.

Streszczenie rozprawy doktorskiej i recenzje są zamieszczone na stronie internetowej: <https://bip.pw.edu.pl/Postepowania-w-sprawie-nadania-stopnia-naukowego/Doktoraty/Wszczete-do-30-kwietnia-2019-r/Dyscyplina-informatyka-techniczna-i-telekomunikacja-dziedzina-nauk-inzynieryjno-technicznych/mgr-inz.Leszek-Ciopinski>

Przewodniczący Rady Naukowej Dyscypliny
Informatyka Techniczna i Telekomunikacja
Politechniki Warszawskiej
dr hab. inż. Jarosław Arabas, prof. uczelni

Synteza samoadaptacyjnych systemów czasu rzeczywistego za pomocą rozwojowego programowania genetycznego

Streszczenie

mgr inż. Leszek Adam Ciopiński

W ostatnich latach jednym z kierunków rozwoju systemów informatycznych jest adaptacyjność. W systemach adaptacyjnych możliwa jest zmiana pewnych własności (np. realizowana funkcja, architektura) w celu dostosowania działania systemu do zmieniających się warunków zewnętrznych. Szczególnym przypadkiem jest tu samoadaptacyjność, gdzie zmiany te zachodzą samoistnie jako automatyczna adaptacja systemu do zmiany otoczenia. W przypadku gdy samoadaptacyjność dotyczy zmiany parametrów jakościowych systemu, mówimy wtedy o tzw. samooptrymalizacji. Praca dotyczy problemu projektowania samooptrymalizującego się oprogramowania wbudowanego, dla systemów wbudowanych implementowanych z wykorzystaniem procesorów wielordzeniowych.

W odniesieniu do systemów wbudowanych celem optymalizacji jest minimalizacja zużycia energii, przy jednoczesnym zapewnieniu odpowiedniej wydajności. Realizowane jest to poprzez wykorzystanie zasobów o wysokiej wydajności, umożliwiających szybkie przetworzenie informacji, oraz zasobów o wysokiej efektywności energetycznej, które zużywają wyraźnie mniej energii, pomimo dłuższego czasu realizacji poszczególnych zadań. Powstaje więc problem optymalnego przyporządkowania zasobów do realizacji poszczególnych funkcji systemu wbudowanego.

Optymalizacja oprogramowania w systemach rozproszonych polega na odpowiednim uszeregowaniu zadań na poszczególnych zasobach obliczeniowych. W przypadku gdy zasobami są rdzenie procesorów o różnych parametrach wydajnościowo-energetycznych, optymalizacja ta dotyczy również problemu minimalizacji poboru energii. Istnieje wiele metod szeregowania zadań, mających na celu zwiększenie wydajności lub minimalizację poboru energii. Jednak nie istnieją metody optymalizacji systemu pod kątem możliwości samoadaptacyjnych.

W pracy przedstawiono nowatorskie rozwiązanie umożliwiające syntezywanie oprogramowania wbudowanego w sposób zapewniający efektywne wykorzystanie zasobów, przy założeniu, że system będzie pracował w dynamicznym środowisku, w którym czasy wykonania zadań mogą się zmieniać w określonym zakresie. Rozwiązanie to zakłada możliwość formalnego opisu założeń systemu i możliwych zmian parametrów zadań oraz wykorzystaniu tych informacji do wykonania syntezy oprogramowania wraz z mechanizmem umożliwiającym samooptrymalizację mającą na celu minimalizację poboru energii przy jednoczesnym zapewnieniu spełnienia ograniczeń czasu rzeczywistego.

Opracowana metoda bazuje na rozwojowym programowaniu genetycznym. Na podstawie specyfikacji systemu w postaci grafu zadań oraz określonych parametrów czasowo-energetycznych zadań, metoda generuje oprogramowanie dla systemu wielordzeniowego, z wbudowanym mechanizmem automatycznego przeszerzegowania zadań, w przypadku gdy czasy wykonania zadań okażą się inne niż zakładano. Wykonane eksperymenty wykazały, że zaproponowana metoda tworzy oprogramowanie znacznie efektywniejsze niż istniejące metody szeregowania zadań.

prof. dr hab. inż. Andrzej Obuchowicz

Zielona Góra, 19 luty 2023

Uniwersytet Zielonogórski

Instytut Sterowania i Systemów Informatycznych

e-mail: a.obuchowicz@issi.uz.zgora.pl

RECENZJA

rozprawy doktorskiej Pana mgr inż. Leszka Ciopińskiego

*Synteza samoadaptacyjnych systemów czasu rzeczywistego z pomocą
rozwojowego programowania genetycznego*

opracowana na zlecenie

Rady Dyscypliny Naukowej Informatyka Techniczna i Telekomunikacja
Politechniki Warszawskiej

1. Problem badawczy i jego znaczenie

Nieustanny wzrost, niemalże wykładniczy, aplikacji systemów budowanych realizowanych w realiach czasu rzeczywistego w otaczających nas produktach technologicznych wymusza na badaczach poszukiwanie coraz to bardziej zaawansowanych metod syntezy oprogramowania tychże systemów. Szczególnie istotnym obszarem stanowią zastosowania systemów w układach izolowanych energetycznie, bez stałego dostępu do zewnętrznych źródeł energii, bazujące głównie, lub jedynie, na mobilnych magazynach energii. W takich przypadkach istotnym elementem jest optymalna gospodarka energią. Praca w realiach czasu rzeczywistego, w warunkach zmiennych, trudnych do przewidzenia, wymaga nadania systemowi cech elastyczności, samoadaptacji do zmiennych wymagań czasowo-proceduralnych. Kreacja modeli syntezy oprogramowania dla takich systemów staje się niezwykle inte-

resującym zagadnieniem. Wobec powyższych faktów cele i teza pracy przedstawione w trzecim rozdziale rozprawy doktorskiej pana mgr inż. Leszka Ciopińskiego uznać należy za słuszne i zasadne.

2. Wkład autora

Zasadniczy oryginalny wkład doktoranta to opracowanie metody syntezy samoadaptacyjnych systemów czasu rzeczywistego spełniających zadane wymagania czasu realizacji i poprawiające parametry oszczędnościowe pobieranej energii. W odróżnieniu od dotychczasowych propozycji, w których nacisk położono na gwarancję wymaganego terminu, w związku z tym harmonogramowanie zadań uwzględniało najbardziej niekorzystne warunki realizacji zadań, a co za tym idzie, są to rozwiązania wysoko energetyczne, w proponowanym rozwiązaniu harmonogram zadań dostosowano do najbardziej prawdopodobnych warunków. W przypadku zajścia sytuacji niekorzystnej, system dynamicznie poprawia przydział zadań na zasoby w celu uniknięcia zagrożenia naruszenia terminu realizacji i przy możliwie najniższym koszcie energetycznym.

Uzyskanie powyższego modelu wymagało opracowania wielu szczegółowych zagadnień, wśród nich wyróżnić należy:

- propozycja miar jakości do oceny samoadaptacyjności uszeregowania czasu rzeczywistego (SURT) i samooptymalizacji poboru energii (SPE), oraz ich wersji uproszczonych;
- opracowanie, dla rozważanej klasy zagadnień, reprezentacji osobnika w rozwojowym algorytmie genetycznym i dopasowanie do jego postaci operacji rekombinacji i mutacji;
- przeprowadzenie badań eksperymentalnych potwierdzających skuteczność proponowanego narzędzia.

3. Poprawność

Opiniowana praca, zawarta na 137 stronach, składa się z 7 rozdziałów, w tym wstępu i podsumowania, uzupełnionych streszczeniami w języku angielskim i polskim,

spisem symboli i oznaczeń, oraz bibliografią zawierającą 113 pozycji cytowanej literatury, zawierającej wyczerpujący zbiór publikacji obejmujący stan badań prowadzonych w obszarze zainteresowania pracy.

Zawartość wstępu, stanowiącego rozdział pierwszy, stanowi w pewnym sensie motywację podjęcia zagadnienia, uwypuklając istotność systemów samoadaptacyjnych we współczesnych rozwiązaniach technologicznych. W rozdziale drugim doktorant szczegółowo rozwinął podjętą myśl wstępu przedstawiając przegląd stosowanych metod samoadaptacyjnych, szczególnie w systemach czasu rzeczywistego. Cele oraz teza pracy została zdefiniowana w rozdziale trzecim. W szczególności wskazano w nim na cechy rozważanej klasy systemów, oczekiwanych własności budowanego modelu, a także tok postępowania prowadzącego do osiągnięcia przedstawionych celów i udowodnienia postawionej tezy. Podstawowe narzędzie użyte do szeregowania zadań – rozwojowe programowanie genetyczne – zostało opisane w rozdziale czwartym. W szczególności przedstawiono technikę kodowania osobnika populacji, powiązane z nim operatory genetyczne: mutację i rekombinację, a także wskazano metodę selekcji. Opis pełnej metody syntezy samoadaptacyjnego oprogramowania systemów wbudowanych czasu rzeczywistego z wykorzystaniem rozwojowego programowania genetycznego zawarto w rozdziale piątym. W rozdziale szóstym przedstawiono badania eksperymentalne mające na celu wykazanie skuteczności proponowanej metody w odniesieniu do zaproponowanych wcześniej kryteriów w zagadnieniach szeregowania zadań z ograniczoną dostępnością i silnymi ograniczeniami. Rozdział siódmy zawiera podsumowanie całej rozprawy.

Z merytorycznego punktu widzenia przeprowadzone badania i analizy są właściwe i poprawnie zrealizowane. Stwierdzenia w większości są uzasadnione, bądź podane są odnośniki do literatury, gdzie odpowiednie uzasadnienie można odnaleźć. Tok głównej myśli rozprawy jest właściwy i dobrze przemyślany. Na szczególne wyróżnienie zasługuje przejrzystość przeprowadzanych rozważań i analiz, jak również niezwykła dbałość o postać edycyjną pracy.

Jednakże poniżej podaję kilka uwag merytorycznych.

1. W rozdziale drugim wśród omawianych metod występują także metody, których doktorant jest współautorem – wskazują na to odwołania do publikacji doktoranta. Należałoby wyróżnić w tekście fakt, że są to prace zawierające

wyniki omawiane dalej w rozprawie. Bez tej informacji czytelnik ma wrażenie, że prace te zawierają tylko pewne poprzedzające rezultaty nie objęte w rozprawie.

2. Rys. 4.1 i komentarz do niego. Algorytmy ewolucyjne można podzielić na dwie klasy: przeszukiwania genotypowe i fenotypowe. W pierwszym przypadku operujemy na dwóch przestrzeniach, przestrzeni fenotypowej, na której określona jest optymalizowana funkcja dopasowania, i przestrzeni genotypowej, na której przebiega proces ewolucyjny, czyli określone są operacje reprodukcji, mutacji, rekombinacji, czy sukcesji. W tym przypadku genotyp zawiera zakodowaną wersję fenotypu (cech), na podstawie których oceniany jest osobnik. Trzeba nadmienić, że kodowanie fenotypu w genotyp zwykle jest nieizomorficzne i zwykle stratne. Najbardziej znanymi reprezentantami tego typu przeszukiwań są algorytmy genetyczne i programowanie genetyczne. W drugim przypadku nie kodujemy fenotypu, tylko ocena osobnika i proces ewolucyjny realizowany jest bezpośrednio na fenotypach. Reprezentantami tego typu przeszukiwań są np. programowanie ewolucyjne, czy strategie ewolucyjne. Rysunek 4.1b sugeruje, że mamy do czynienia z przeszukiwaniem genotypowym, a rysunek 4.1a przeszukiwaniem fenotypowym, do której to klasy algorytmy genetyczne nie należą.
3. Str. 42 trzeci punkt listy punktowanej. Nie rozumiem dlaczego mamy unikać przypadku uzyskania lepszego osobnika jako wyniku przypadkowego losowania, tylko musi to być efekt krzyżowania.
4. Rozdział 4. W opisie proponowanego algorytmu rozwojowego programowania genetycznego brakuje ogólnego schematu całego procesu. Np. niejasne jest miejsce selekcji. Czy, w przypadku n -elementowej populacji, selekcji turniejowej rzędu m używamy w celu wygenerowania n osobników rodzicielskich (zwycięzców n turniejów grup liczebności m), a następnie po mutacjach i krzyżowaniach otrzymamy n -elementową populację nowej generacji? Czy z aktualnej n -elementowej populacji wybieramy metodą turniejową k -elementową podpopulację rodzicielską, która jest zastępowana przez utworzonych z niej za pomocą mutacji i krzyżowań potomków, bądź wybieramy nową n -elementową populację nowej generacji jako najlepszych osobników $n + k$ -elementowej populacji poprzedniej generacji uzupełnionej k potomkami? Opcji jest wiele. Nie

podano jaką użyto w pracy i dlaczego.

5. Wybór metody selekcji w postaci selekcji turniejowej uzasadniono powołując się na badania opublikowane w pracy [47], której doktorant jest współautorem. Dla pełności raportu badawczego, opis tych badań powinien znaleźć się w rozprawie, chociażby w postaci dodatku.
6. Definicje od 5.3 do 5.7 bardzo zbliżone są do założeń znanej z obszaru badań operacyjnych probabilistycznej metody planowania i kontroli projektu z analizą kosztową (PERT-COST, ang. Project Evaluation and Review Technique), czy elementy tej metody były wykorzystywane w budowie modelu? np. przy oszacowaniu maksymalnych opóźnień realizacji danego zadania na danym zasobie nie zagrażającym końcowemu reżimowi czasowemu.
7. Pojęcie definicji używane jest do pełnego określenia pojęcia. Czy definicje 5.9 i 5.10 spełniają ten warunek, tzn. czy możliwe są inne wyrażenia definiujące j^{SURT} i j^{SPE} niż wyrażenia (5.7) i (5.11)? Jeżeli nie, to wyrażenia (5.7) i (5.11) powinny być zawarte odpowiednio w definicjach 5.9 i 5.10.
8. Podobnie rozumując jak wyżej wyrażenie (5.2) powinno być objęte definicją 5.14, (5.3) powinno być objęte definicją 5.15 i wyrażenie (5.4) powinno być objęte definicją 5.16.
9. Strona 62. Słuszne jest zdanie, w którym oczekuje się, że ograniczona liczba scenariuszy testowych powinna być reprezentatywna dla całej przestrzeni możliwych przypadków. Jak zapewnić tę reprezentatywność wybierając zbiór testowy? Bądź jak uzasadnić, że wybrany zbiór jest reprezentatywny? Zwłaszcza, kiedy przestrzeń wszystkich scenariuszy jest przestrzenią nieciągłą.

Jakkolwiek praca została napisana z dużą dbałością edycyjną, to pojawiło się kilka niedociągnięć.

1. Brak mi podsumowań poszczególnych rozdziałów, w których wyróżniono by istotne wnioski i fakty ułatwiające czytelnikowi śledzenie myśli autora i rozeznanie etapu procesu przeprowadzanego dowodu tezy i osiągnięcia celów wskazanych we rozdziale trzecim.

2. Niektóre skróty literowe pojawiają się przed późniejszym ich rozwinięciem, lub wyjaśnieniem. Na przykład QoS pojawia się na stronie 14, wyjaśnione jest na stronie 20.
3. Strona 19 podrozdział 2.3. Zapis w nawiasie (*SZODZ, ang. Resource-Constrained Project Scheduling Problem*) sugeruje angielskie wyjaśnienie skrótu, który jest skrótem polskim. Polski termin podany jest wcześniej, ale poszczególne słowa terminu powinny być tam zapisane poczynając od dużych liter.

4. Wiedza kandydata

Podjęty przez doktoranta w rozprawie problem naukowy wymaga wiedzy i umiejętności z wielu obszarów badawczych, w szczególności z zakresu programowania rekonfigurowalnych systemów wbudowanych, szeregowania zadań, czy technik ewolucyjnych. Zawartość treści w pierwszych dwóch rozdziałach pracy zapoznającą czytelnika z aktualnym stanem wiedzy dziedzinowej, a także opis jej realizacji w rozdziałach następnych wskazuje na szeroką wiedzę autora rozprawy w powyższych obszarach i umiejętność z niej korzystanie. Ponadto doktorant wykazał się poprawnością terminologiczną i szeroką znajomością literatury przedmiotu.

5. Podsumowanie

Biorąc pod uwagę opinie zaprezentowane w poprzednich punktach, w tym nieliczne uwagi krytyczne, w części natury dyskusyjnej i edytorskiej, i wymagania zdefiniowane przez artykuł 187 ustawy z dnia 20 lipca 2018r. Prawo o szkolnictwie wyższym i nauce (z późniejszymi zmianami) **stwierdzam, że:** rozprawa doktorska pana mgr inż. Leszka Ciopińskiego zawiera oryginalne rozwiązanie problemu naukowego, kandydat posiada ogólną wiedzę teoretyczną w dyscyplinie *informatyka techniczna i telekomunikacja* i posiada umiejętność samodzielnego prowadzenia pracy naukowej.

W związku z powyższym **wnoszę** o przyjęcie rozprawy przez odpowiednią Komisję powołaną przez Radę Dyscypliny Naukowej Informatyka Techniczna i Telekomunikacja Politechniki Warszawskiej i dopuszczenie jej do publicznej obrony.



Dr hab. Rafał Różycki, prof. PP
Instytut Informatyki
Politechnika Poznańska
ul. Piotrowo 2, 60-965 Poznań
rafal.rozycki@put.poznan.pl

Rada Naukowa Dyscypliny
INFORMATYKA TECHNICZNA
I TELEKOMUNIKACJA
Sekretariat
Data wpływu 25.04.23r.
Numer.....



Poznań, 19.04.2023

Recenzja rozprawy doktorskiej

mgr. inż. Leszka Adama Ciopińskiego

p.t.: *Synteza samoadaptacyjnych systemów czasu rzeczywistego
za pomocą rozwojowego programowania genetycznego*

będącej podstawą o ubieganie się o nadanie stopnia naukowego
doktora w dyscyplinie naukowej informatyka techniczna i telekomunikacja.

Niniejsza recenzja została przygotowana w odpowiedzi na uchwałę nr 402/2022 Rady Naukowej Dyscypliny Informatyka Techniczna i Telekomunikacja Politechniki Warszawskiej z dnia 13.12.2022 r. w sprawie powołania komisji doktorskiej, określenia zakresu egzaminów doktorskich oraz powołania komisji przeprowadzających egzaminy doktorskie w przewodzie doktorskim mgr. inż. Leszka Ciopińskiego.

Postępowanie odbywa się w trybie przewidzianym w Ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz.U. poz.1668, z późn. zm.).

1. Ogólna charakterystyka rozprawy

Do recenzji przedstawiono rozprawę doktorską wydaną w roku 2022 nakładem wydawnictwa Politechniki Warszawskiej. Jedynym autorem pracy jest Doktorant – mgr inż. Leszek Adam Ciopiński. Strona tytułowa wskazuje, że treść pracy zawiera wyniki badań prowadzonych w dziedzinie nauk inżyniersko-technicznych, w dyscyplinie naukowej informatyka techniczna i telekomunikacja.

Praca napisana jest po polsku, zawiera też obszernie i wyczerpujące streszczenie w języku angielskim.

2. Cel i zakres rozprawy

Przedstawione do oceny materiały obejmuje wyniki badań nad opracowaniem innowacyjnej metody syntezy oprogramowania systemów czasu rzeczywistego. W systemach czasu rzeczywistego nakłada się na uruchamianie w nich oprogramowanie silne ograniczenia czasowe tak, by uzyskać gwarancję terminowej realizacji funkcji spełnianych przez taki system. W najprostszej sytuacji zakłada się tzw. linię krytyczną, czyli graniczny czas działania wszystkich komponentów oprogramowania, którego przekroczenie jest zabronione.

W pracy jako warstwę sprzętową systemu czasu rzeczywistego rozważa się nowoczesne architektury komputerowe, w których podstawowym elementem wykonawczym jest pojedynczy procesor wielordzeniowy lub zbiór równoległych procesorów. Istotną cechą takich systemów jest to, że podczas wykonywania tego samego zadania obliczeniowego poszczególne rdzenie procesora/procesory mogą mieć różną wydajność obliczeniową oraz zużywać różne ilości energii.

W pracy przyjmuje się, że na uruchamianie w systemie oprogramowania składa się skończony zbiór niepodzielnych zadań obliczeniowych o znanych typowych, minimalnych i maksymalnych wartościach czasów wykonywania na poszczególnych procesorach (rdzeniach). Każdemu czasowi wykonania zadania odpowiada pewne zużycie energii, którego wartość zależy zarówno od zadania jak i procesora. Dodatkowo zakłada się, że zadania nie mają określonego czasu gotowości do wykonania, ale między poszczególnymi zadaniami mogą występować relacje kolejnościowe.

Podstawową zaletą proponowanego w pracy systemu ma być zdolność uruchamianego w nim oprogramowania do adaptacji do zmiennych warunków zewnętrznych wpływających na działanie oprogramowania. Adaptacja ta uwzględniać ma przy tym konieczność dotrzymania przyjętej w systemie czasu rzeczywistego linii krytycznej, a zarazem – w razie możliwości - minimalizować zużycie energii.

Tak określony zakres badań uważam za bardzo interesujący i wychodzący naprzeciw potrzebom opracowywania wydajnych, odpornych na zakłócenia i oszczędnych energetycznie systemów czasu rzeczywistego. Systemy o takich cechach są szczególnie pożądane w zyskujących na popularności koncepcjach Internetu rzeczy i przetwarzania brzegowego.

W szczególności autor postawił sobie za cel zabudowanie systemu, którego działanie nie jest projektowane na bazie najgorszych (pesymistycznych) scenariuszy wykonania komponentów oprogramowania, lecz opiera się ono na scenariuszach typowych – co w zamyśle autora ma pozwolić na bardziej oszczędne wykorzystanie energii.

3. Teza rozprawa

Auto stawia w swojej pracy następującą tezę:

„Synteza oprogramowania systemów wbudowanych czasu rzeczywistego metodą RPG pozwala na uzyskanie większych możliwości samoadaptacyjnych systemu niż w przypadku stosowania istniejących metod przeszerzgowania zadań”

4. Układ rozprawy

Oprócz streszczenia praca zawiera siedem rozdziałów, spis wykorzystanych oznaczeń oraz bibliografię.

We Wstępie (Rozdział 1) zarysowany jest kontekst prowadzonych badań, na wielu przykładach systemów rzeczywistych wskazane są różne aspekty samoadaptacyjności.

W Rozdziale 2 autor przedstawia pojęcie samoadaptacyjności w kontekście systemów informatycznych, opisuje klasę samoadaptacyjnych zorientowanych obiektowo systemów czasu rzeczywistego, wskazuje na analogie do ogólnego problemu rozdziału ograniczonych zasobów systemu komputerowego.

Rozdział 3 dedykowany jest przedstawieniu celu i przyjętej tezy rozprawy. Wymienia się w nim przyjęte założenia i zakres pracy badawczej.

Oryginalną wersję metody Rozwojowego Programowania Genetycznego (RPG) w zastosowaniu do rozważanego problemu optymalizacyjnego zaprezentowano w Rozdziale 4.



Aspekt zdolności do samoadaptacji proponowanego systemu do warunków zewnętrznych rozważany jest szczegółowo w Rozdziale 5. Wymieniono tu założenia dotyczące parametrów zadań, opisano algorytmy generowania harmonogramów ich wykonywania na podstawie danych zaszytych w genotypie rozwiązania, zaproponowano też sposoby oceniania uzyskanych rozwiązań.

Opisane wcześniej miary oceny uzyskiwanych rozwiązań stanowią ważny komponent procesu weryfikacji efektywności zaproponowanego podejścia. Dokonano jej na podstawie eksperymentów obliczeniowych opisanych w Rozdziale 6.

Ostatni rozdział podsumowuje wyniki przeprowadzonych badań.

Umieszczony na końcu rozprawy skorowidz użytych symboli i nazw poprawia wygodę zapoznawania się z jej treścią.

5. Dyskusja merytoryczna

A) Teza pracy i sposób jej dowodzenia

Autor postawił przed sobą zadanie udowodnienia postawionej tezy (str.27 – „Teza ta zostanie udowodniona...”), która jest dość śmiała. Twierdzi się w niej, że zaproponowane rozwiązanie jest w przypadku ogólnym obiektywnie lepsze (w sensie przyjętej „miary samoadaptacyjności”) od każdej innej metody, w której realizowane jest przeszeregowywanie zadań. W mojej ocenie przedstawiony w pracy sposób dowodzenia – m.in. na podstawie bardzo ograniczonego eksperymentu obliczeniowego, z formalnego punktu widzenia jest niewystarczający.

Ponadto wątpliwość budzi sama teza, w której uwypukla się znaczenie metody RPG. Wydaje się jednak, że jeżeli poszukiwać źródła raportowanej efektywności proponowanego podejścia, to tkwi ona w sposobie reprezentowania rozwiązania (budowa chromosomu) i przyjętej metodzie jego dekodowania (algorytmy mapowania). Podobną efektywność prawdopodobnie dałoby się uzyskać wykorzystując inne podejścia metaheurystyczne operujące na tej samej reprezentacji i metodzie dekodowania rozwiązania.

B) Sposób funkcjonowania systemu

Główną wątpliwość budzi sposób funkcjonowania projektowanego systemu. Jak rozumiem, procedura RPG jest wykorzystywana do znalezienia metody konstruowania energooszczędnego uszeregowania zadań, która byłaby odporna na (nieprzewidywalne) zakłócenia występujące podczas realizacji tych zadań. Odrębna metoda zaszyta jest w każdym osobniku populacji algorytmu RPG i przyjętych algorytmach dekodowania osobnika do finalnego uszeregowania zadań na procesorach (rdzeniach procesorów). Załóżmy, że RPG dla danej instancji problemu (danego oprogramowania) znalazł najlepszego (w sensie przyjętych kryteriów) osobnika. Autor na str.22 wspomina, że „w systemach czasu rzeczywistego jedynie szybkie metody szeregowania zadań mogą być używane do harmonogramowania lub przeszeregowania zadań w trakcie działania systemu”. W tym kontekście pojawiają się następujące więc pytania:

- czy algorytmy dekodujące osobnika realizowane są na tych samych zasobach obliczeniowych (procesorach/rdzeniach procesorów) co docelowe oprogramowanie uruchamiane w systemie czasu rzeczywistego (w streszczeniu deklaruje się prace nad systemem z wbudowanym mechanizmem przeszeregowania)?
- jeśli odpowiedź na powyższe pytanie jest twierdząca, to czy dokonano oceny czasowej złożoności obliczeniowej algorytmów dekodujących oraz ilości zużywanej przez te algorytmy energii?
- jaki proces i w jaki sposób kontroluje odstępstwa od przyjętego harmonogramu realizacji zadań obliczeniowych składających się na oprogramowanie systemu czasu rzeczywistego?

- czy ewentualna akcja przeszerzegowania zadań podejmowana jest natychmiast po wykryciu odstępstwa, czy np. dopiero po zakończeniu zadania, którego wykonywanie trwa dłużej niż zaplanowano?
- jak poradzi sobie system z sytuacją, gdy opóźnione jest (ponad bezpieczną wartość buforową) zakończenie ostatniego zadania w zaplanowanym harmonogramie?
- a może akcja przeszerzegowania zadań podejmowana jest dopiero na początku następnego cyklu uruchomienia oprogramowania w systemie? Jak długo ona trwa i jakie parametry zadań są brane pod uwagę – zaktualizowane na podstawie danych z poprzedniego cyklu?

Precyzyjnej odpowiedzi na powyższe przykładowe pytania trudno znaleźć w pracy (poza enigmatycznymi wyjaśnieniami na str. 19, 26), a są one kluczowe dla zrozumienia podstawowych zasad działania finalnej wersji całego systemu.

We Wstępie (str.12-13) wymienione i opisane są etapy działania systemu samoadaptacyjnego (Monitorowanie, Analiza, Planowanie i Wykonanie) – niestety w dalszym tekście nie pojawia się już odniesienie do tych terminów w kontekście prezentowanych wyników badań.

C) Sformułowanie problemu

W pracy brakuje rozdziału, który zawierałby pełny model projektowanego systemu, wszystkie przyjęte kryteria optymalizacji, występujące w nim ograniczenia, parametry oraz zmienne decyzyjne. Taki formalny zapis ułatwiłby śledzenie prezentowanych wyników badań.

Dobłą praktyką jest sformułowanie problemu w postaci problemu programowania matematycznego – tu tego zabrakło.

D) Klasyfikacja rozważanego problemu szeregowania

Rozważany w pracy problem szeregowania autor klasyfikuje jako *Resource Constrained Project Scheduling Problem* (RCPSP), co w mojej ocenie jest założeniem błędnym. W RCPSP zakłada się występowanie wielu różnych typów i kategorii zasobów, o które współubiegają się zadania (czynności) wchodzące w skład całego projektu. W klasycznym RCPSP każde zadanie charakteryzuje zapotrzebowanie (być może zerowe) na każdy typ zasobu występujący w systemie i nie ma możliwości wykonania zadania w alternatywny sposób (czyli przy zapotrzebowaniu na inne zasoby).

W recenzowanej pracy z zasobami utożsamia się procesory (rdzenie procesorów), co do których zakłada się w ogólności, że nie są identyczne i dlatego nie mogą być traktowane jako zasób tego samego typu. Przydzielając zadanie do procesora o innej charakterystyce, de facto realizujemy inny (alternatywny) sposób jego wykonywania.

W celu zamodelowania takiej sytuacji w kontekście problemów rozdziału zasobów (ang. *Project Scheduling*) należałoby przyjąć wariant ogólniejszy tej klasy problemów, czyli tzw. *Multi-Mode Project Scheduling Problem* (MM RCPSP). W mojej ocenie jest to jednak zbędne, gdyż przy przyjętych w pracy założeniach rozważany problem optymalizacyjny jest pewnym szczególnym przypadkiem klasycznego problemu szeregowania zadań zależnych, niepodzielnych na maszynach (procesorach) równoległych dowolnych, bez żadnych dodatkowych zasobów. Specyfika rozważanego maszynowego problemu szeregowania wynika jedynie z charakterystyki maszyn oraz przyjętego kryterium (-ów) optymalizacji. Sugestię tę potwierdza w pewnym sensie autor wskazując - jako referencyjne metody szeregowania - algorytmy LLF i EDF, czyli heurystyki dedykowane do rozwiązywania problemów szeregowania zadań na maszynach.

E) Używana terminologia

W pracy brakuje elementarnego rygoryzmu w używaniu przyjętych terminów, który winien charakteryzować pracę naukową. Ze zbyt dużą swobodą, zamiennie używa się na przykład pojęć: program – oprogramowanie – aplikacja; uszeregowanie – harmonogram; generacja – pokolenie – iteracja.

Chociaż w pracy energia nie jest traktowana jako zasób podwójnie ograniczony, jej zużycie jest uwzględniane w przyjętym kryterium optymalizacji. Na pewną krytykę zasługuje więc fakt iż w pracy nie rozróżnia się pojęć: moc vs. energia (lub pobór mocy vs. zużycie energii), zamiast tego używa się na przykład dość sztucznego pojęcia „jednostkowy koszt energetyczny energii” (patrz Definicja 5.7).

F) Problem numerowania zadań (węzłów) w grafie

W przedstawionej w pracy metodzie dekodowania genotypu w fenotyp, podstawowe znaczenie ma numer zadania, który utożsamiany jest z jego priorytetem. W pracy proponuje się prosty algorytm numerowania zadań (pkt. 4.2.4). W ogóle nie rozważa się jednak możliwości, że poprawna numeracja (uwzględniająca porządek zadań wynikający z relacji kolejnościowych między nimi) nie musi być jedyna. Konsekwencje tego przeoczenia są bardzo poważne. Najlepsze uszeregowanie może być bowiem w ogóle nieosiągalne, jeśli nie uwzględni się wszystkich dopuszczalnych porządków numerów wierzchołków w grafie zadań.

G) Założenia eksperymentu obliczeniowego

Jeśli przyjęte w eksperymentach założenie, że strojenie algorytmu RPG wykonywane jest dla każdej rozwiązywanej instancji problemu, należy przypuszczać, że podobną efektywność osiągnęłyby inne podejścia metaheurystyczne strojone w podobny sposób.

Cytowany w rozprawie algorytm GA autorstwa S. Hartmanna [61] podczas procesu ewolucji również posługuje się genotypem, który w finalne uszeregowanie (harmonogram) przekształcany jest za pomocą dedykowanych algorytmów, tzw. SGS (ang. *Schedule Generation Scheme*). Wykorzystanie w eksperymentach GA Hartmanna (lub innej metody metaheurystycznej), przy założeniu podobnych parametrów zadań (uwzględniających bufor czasu), przyniosłoby ciekawy materiał porównawczy (abstrahując od faktu, że GA Hartmanna w wersji podstawowej nie nadaje się do maszynowych problemów szeregowania). Porównywanie rozwiązania bazującego na RPG, który jest strojony do rozwiązywania konkretnej instancji (!), z wynikiem działania prostej heurystyki konstrukcyjnej (LPLLF) nie pozwala na sformułowanie rzetelnych wniosków.

H) Uproszczony sposób oceny rozwiązania

Autor sam sygnalizuje problem z rzetelną oceną generowanych przez proponowaną metodę rozwiązań. Wymagałoby to złożonej analizy statystycznej uwzględniającej różne rozkłady prawdopodobieństwa zmiany parametrów zadań obliczeniowych wchodzących w skład oprogramowania. Ze względu na konieczność częstej oceny rozwiązań – podejście takie jest nierealne ze względu na niezbędny czas obliczeń. Autor proponuje więc uproszczony sposób oceny rozwiązań, zakładając w eksperymentach możliwość wystąpienia tylko bardzo ograniczonego zbioru zakłóceń realizacji harmonogramu. Chociaż intencje autora wydają się zrozumiałe, takie podejście podważa jednak sens wyciągania wniosków ogólnych z eksperymentu, w którym używa się procedury szacowania jakości rozwiązania na bazie niereprezentatywnego podzbioru zakłóceń realizacji harmonogramu.

I) Inne uwagi ogólne

- Nie do końca jest jasne dla jakiego systemu zaproponowano opisane podejście. Jeśli jest to system operacyjny (w szczególności – system operacyjny czasu rzeczywistego), w pracy przydałby się rozdział dotyczący systemów operacyjnych i stosowanych w nich mechanizmach zarządzania zasobami.
- Chociaż w pracy jawnie wykorzystywane są różne kryteria oceny poszukiwanego rozwiązania, trudno odnaleźć w niej jakiekolwiek odniesienie do prac z zakresu optymalizacji wielokryterialnej. Prezentacja własnych wyników badań w odniesieniu do znanych podejść rozwiązywania problemów wielokryterialnych, pozwoliłaby wypełnić tę lukę.
- Nigdzie w pracy nie można znaleźć opisu działania operatora reprodukcji w proponowanym algorytmie RPG.
- Na energię nie są nałożone żadne ograniczenia, dlatego zarówno jej chwilowa dostępność (czyli moc) jak i zużycie są nieograniczone, co wydaje się założeniem mało praktycznym.
- Autor często posługuje się terminem „poprawny harmonogram” nie definiując wcześniej formalnie, co to oznacza. Domyślać się należy, że chodzi tu o uszeregowanie, w którym ostatnie zadanie kończy się przed zadaną linią krytyczną, czyli dopuszczalne.

J) Uwagi szczegółowe

- Na str. 13 autor wymienia technologię DVFS jako alternatywny sposób na osiągnięcie założonych w pracy celów. Jedynym przytoczonym argumentem za projektowaniem systemów samoadaptacyjnych jest stwierdzenie, że „(DVFS) ma jednak różne ograniczenia”. Jest to oczywiście stwierdzenie zbyt ogólne i wymagające przynajmniej skrótowego rozwinięcia.
- Na str. 19 pada stwierdzenie, że problem SZODZ (ang. RCPSP) „jest problemem NP-zupełnym”, choć wcześniej problem ten przedstawiany jest jako problem optymalizacyjny. W teorii złożoności obliczeniowej klasę problemów NP-zupełnych definiuje się jednak w kontekście problemów decyzyjnych, a nie optymalizacyjnych.
- Pierwsze zdanie na str.20 („Podejmowane wiele...”) demonstruje pewną niefrasobliwość autora w precyzyjnym formułowaniu przekazu. Niezgodnie z intencją autora (tak się domyślam!), zdanie to bowiem sugeruje, że podejmowane są próby rozwiązania problemu RCPSP jako takiego, a nie opracowania efektywnych metod poszukiwania rozwiązań różnych klas jego instancji.
- Na str. 20 pojawia się również termin „wielozadaniowe SZODZ” – bez dodatkowego wyjaśnienia, trudno dociec, co on oznacza.
- Na stronie 20 autor wspomina o klasie problemów SO SZODZ, czyli takich, w których występują „silne ograniczenia”. Nie sprecyzowano jednak kiedy ograniczenia mogą być interpretowane jako „silne” i czy jest to własność samej instancji problemu, czy też jest ona ściśle powiązana z algorytmem zastosowanym do rozwiązywania problemu.
- W punkcie 4.3.2 znajduje się opis reprezentacji rozwiązania (genotypu). W kontekście opisu ze str.53 (wprowadza się tam dodatkowy chromosom) okazuje się on jednak niepełny.

- Na str. 40 znajduje się odnośnik do Rys.4.4, którego interpretacja na tym etapie przedstawiania metody dekodowania genotypu w fenotyp (bez referencji od odpowiednich algorytmów) jest niemożliwa.
- Na str. 52 autor zauważa, że „im większe rezerwy czasowe, tym (...) zwykle większa energochłonność całego układu”. Stwierdzenie to stoi w sprzeczności z Definicją 5.6.
- Na str. 75 (pkt. 6.1.2) pada stwierdzenie, że „(...)wszystkie algorytmy zostaną dostrojone zgodnie z opisem z podrozdziału 6.1.1.”. Jako, że podrozdział 6.1.1 dotyczy strojenia metody RPG, zacytowane zdanie sugeruje, że w opisywanym eksperymencie obliczeniowy, algorytm RPG był ręcznie dostrajany do rozwiązywanej instancji problemu. W takiej sytuacji trudno utrzymać argument, że proponowany w pracy system ma własność samoadaptacji.

7. Strona redakcyjna rozprawy

Niektóre fragmenty pracy zmuszają czytelnika do uciążliwego wertowania kart – na przykład interpretacja Rys. 4.4 wymaga jednoczesnego dostępu do stron: 37, 40, 41, 46. W pracy występuje też dużo drobnych błędów redakcyjnych, błędy zawierają też niektóre rysunki (np. Rys. 6.7). Zdarzają się niedokończone zdania (np. koniec str. 93). Używane są akronimy, których znaczenie nie jest nigdzie wyjaśnione (DVFS – str.13, FPGA, LUT – str.22,...).

Zaletą pracy jest bez wątpienia poprawność stylistyczna prezentowanego materiału.

8. Wniosek końcowy

Jak wykazano powyżej, recenzowana rozprawa doktorska ma wiele mankamentów, zawiera sporo niejasności – raportowane w niej wyniki są trudne do interpretowania. Rzeczowa odpowiedź na większość (jeśli nie wszystkie) z uwag krytycznych w pełni potwierdziłaby predyspozycje Doktoranta do prowadzenia badań naukowych.

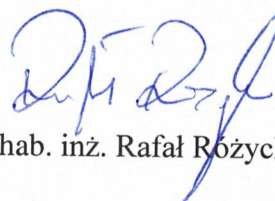
Mimo wielu uwag krytycznych rozprawa doktorska potwierdza ogólną wiedzę teoretyczną kandydata w dyscyplinie informatyka techniczna i telekomunikacja oraz podstawowe umiejętności samodzielnego prowadzenia pracy naukowej. Jej przedmiotem jest oryginalne rozwiązanie problemu naukowego, mającego potencjałe zastosowanie praktyczne.

Na podstawie materiału zgromadzonego w recenzowanej pracy, za największe osiągnięcia Doktoranta uważam:

- zaproponowanie ogólnego podejścia do syntezy energooszczędnego oprogramowania spełniającego twarde wymagania czasowe nowoczesnego systemu czasu rzeczywistego, bazującego na reaktywnej procedurze korygowania proaktywnego harmonogramu wykonywania poszczególnych zadań wchodzących w skład oprogramowania;
- opracowanie oryginalnego sposobu reprezentowania i dekodowania metody konstruowania harmonogramu zadań obliczeniowych;
- dokonanie analizy efektywności zaproponowanego rozwiązania na podstawie wyników eksperymentu obliczeniowego.

Doceniając duży wkład Doktoranta w przygotowanie proponowanego podejścia do syntezy oprogramowania systemów czasu rzeczywistego, eksperymentów obliczeniowych i samej rozprawy, uwzględniając, że jest On dopiero na początku swojej pracy naukowej, składam się ku pozytywnej ocenie przedstawionej do recenzji rozprawy.

W związku z powyższym uznaję recenzowaną rozprawę za osiągnięcie wystarczające do skierowania wniosku mgr. inż. Leszka Adama Ciopińskiego, o nadanie stopnia naukowego doktora w dziedzinie nauk inżynieryjno-technicznych w dyscyplinie informatyka techniczna i telekomunikacja do kolejnych etapów postępowania.



Dr hab. inż. Rafał Różycki, prof. PP