

Recenzja rozprawy doktorskiej mgr. inż. Adriana Woźniaka pt. „Metoda efektywnego rozmieszczania komponentów oraz sterowania ich obciążeniem w systemach klasy SOA”

1) Wymogi prawne

Zgodnie z art. 13 ust. 1 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz stopniach i tytule w zakresie sztuki (Dz. U. Nr 65, poz. 595, z późn. zm.) rozprawa doktorska, przygotowana pod opieką promotora, powinna stanowić oryginalne rozwiązanie problemu naukowego oraz wykazywać ogólną wiedzę teoretyczną kandydata w danej dyscyplinie naukowej, a także umiejętność samodzielnego prowadzenia pracy naukowej. Zgodnie zaś z § 6 ust. 4 rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 19 stycznia 2018 r. w sprawie szczegółowego trybu przeprowadzania czynności w przewodach doktorskim i habilitacyjnym oraz w postępowaniu o nadanie tytułu profesora (Dz. U. 2018, poz. 261) – recenzja rozprawy doktorskiej ma zawierać szczegółowo uzasadnioną ocenę, czy rozprawa ta spełnia warunki określone w art. 13 cytowanej ustawy; recenzja może też zawierać wnioski dotyczące ewentualnego uzupełnienia lub poprawienia rozprawy (ust. 6).

2) Tematyka

Głównym tematem recenzowanej rozprawy doktorskiej jest optymalizacja systemów zorientowanych na usługi (SOA - ang. Service Oriented Architecture) z punktu widzenia realizacji procesów biznesowych. Celem optymalizacji jest minimalizacja czasu realizacji procesów, zużywanych przez procesy zasobów oraz liczby niezrealizowanych procesów. Zmiennymi decyzyjnymi są w tym przypadku typy algorytmów harmonogramowania i kompozycji usług zastosowane w SOA a także przydział komponentów programowych implementujących usługi do serwerów. Usługi w systemach SOA są składowymi procesów biznesowych natomiast algorytmy harmonogramowania i kompozycji usług odnoszą się do kolejności obsługi usług w komponentach oraz wyboru komponentów do realizacji określonej usługi. Zakłada się w tym przypadku, że istnieje wiele komponentów implementujących tą samą usługę a sposób implementacji wpływający na czas realizacji usługi oraz wykorzystywane zasoby może być różny w poszczególnych komponentach.

W celu optymalizacji SOA zastosowano oryginalne rozwiązanie będące połączeniem czterech metod, tj. modelowania matematycznego, symulacji komputerowej, metody „brute force” oraz algorytmu genetycznego. Model matematyczny SOA zaproponowany w pracy i jego programowa implementacja pozwalają na symulację działania dowolnego systemu i wyznaczenie jego parametrów wydajnościowych, czasowych i niezawodnościowych. Parametry te są niezbędne w procesie oceny różnych wariantów SOA generowanych podczas działania algorytmu optymalizacji metodą „brute force” oraz przez algorytm genetyczny. Metoda „brute force” służy do wyboru algorytmów harmonogramowania i kompozycji usług poprzez pełny przegląd wszystkich możliwych opcji w tym zakresie, natomiast algorytm genetyczny jest zastosowany do określenia optymalnego przydziału komponentów do serwerów.

3) Zawartość rozprawy

Przedłożona do recenzji rozprawa liczy 92 strony. Składa się z: wstępu, 7 rozdziałów zasadniczych, wniosków końcowych, wykazu tabel i rysunków i spisu literatury. Spis literatury liczy 131 pozycji, głównie w języku angielskim.

Pierwszy zasadniczy rozdział rozprawy to wprowadzenie w tematykę systemów SOA. Autor przedstawia ogólną charakterystykę tych systemów, a także omawia ich podstawowe komponenty takie jak procesy biznesowe, usługi oraz korporacyjną szynę usług.

Drugi rozdział zawiera przegląd literatury. Na wstępie Autor opisuje procedurę wyboru źródeł do analizy, wskazuje zakres tematyczny prac, które zostały uwzględnione w przeglądzie a także tych, które zostały wyłączone z przeglądu, określa bazy publikacji, w których poszukiwano materiałów naukowych, dokonuje kategoryzacji wybranych materiałów i ostatecznie przechodzi do ich szczegółowej analizy. Obejmowała ona trzy wcześniej zdefiniowane kategorie, tj. algorytmy komponowania usług, algorytmy harmonogramowania usług oraz alokacje zasobów do usług. Należy stwierdzić, iż przegląd literatury został przeprowadzony bardzo starannie.

Rozdział trzeci mieści się tylko na jednej stronie i definiuje cel, tezę rozprawy i wymienia elementy pracy, które zostały opracowane i zaimplementowane w celu udowodnienia tezy i osiągnięcia celu.

W rozdziale czwartym Autor przedstawia opracowany przez siebie model matematyczny systemów klasy SOA. Model ten definiuje składowe systemów SOA, tj. procesy, usługi, algorytmy komponowania i harmonogramowania usług, komponenty programowe implementujące usługi, serwery i środowiska uruchomieniowe, relacje występujące między składowymi oraz ich parametry wydajnościowe, czasowe i niezawodnościowe. Ponadto, rozdział ten definiuje również zadanie optymalizacji wskazując zmienne decyzyjne, ograniczenia oraz wektorową funkcję celu.

Rozdział piąty rozprawy to opis algorytmu optymalizacji będącego kombinacją metody brutalnej siły oraz algorytmu genetycznego. Idea algorytmu zaproponowanego przez Autora polega na zastosowaniu algorytmu genetycznego do określenia przydziału komponentów realizujących usługi do serwerów. Algorytm genetyczny jest uruchomiany dla każdej możliwej pary: (algorytm komponowania usług, algorytm harmonogramowania usług). W rozdziale tym Autor szczegółowo opisuje: sposób kodowania przydziału komponentów do serwerów w postaci chromosomu, procedurę generowania populacji początkowej z algorytmem naprawy zakodowanych rozwiązań znajdujących się poza zbiorem rozwiązań dopuszczalnych, operatory genetyczne, oraz procedurę oceny każdego rozwiązania wytworzonego przez algorytm genetyczny.

W rozdziale szóstym Autor zaprezentował architekturę symulatora implementującego model matematyczny systemów SOA zdefiniowany w rozdziale czwartym. W tym celu Autor posługuje się notacją UML.

Rozdział siódmy to raport z przeprowadzonych badań. Na wstępie Autor przedstawia warunki badań, które odnoszą się głównie do parametrów systemu SOA poddawanego optymalizacji a także do parametrów algorytmu optymalizacji. Następnie Autor prezentuje po kolei: wyniki badań dla wstępnie ustalonych parametrów, wyniki badań zbieżności algorytmu zarówno w dziedzinie rozwiązań jak i w dziedzinie ocen dla różnych ustawień algorytmu optymalizacji, wyniki badań wrażliwości algorytmu rozumianej jako stabilność w odpowiedzi na zmianę parametrów zadania, oraz wyniki badań wydajnościowych implementacji systemu mierzących czas symulacji dla różnych parametrów.

Ostatnia część rozprawy to wnioski końcowe, w których Autor podsumowuje wykonane przez siebie badania, wskazuje co stanowi o oryginalności metody przedstawionej w rozprawie, ponadto wyszczególnia on możliwe modyfikacje opracowanej przez siebie metody oraz oprogramowania badawczego.

Podsumowując należy stwierdzić, iż struktura pracy jest generalnie bez zarzutu. Jedynym mankamentem wydaje się być brak listy skrótów i symboli matematycznych używanych w pracy. Taka lista spowodowałaby łatwiejszą lekturę pracy, szczególnie tej jej części poświęconej modelowi matematycznemu. Co więcej, sporządzając taką listę Autor mógłby uniknąć błędów w definicji kryteriów k3, k4 i k6 (szegóły podane są w pkt. 6 recenzji „Uwagi szczegółowe”)

Język pracy bardzo staranny w początkowych rozdziałach staje się coraz bardziej niedbały w jej końcowych częściach. Pojawiają się tam coraz częściej błędy językowe, których przykłady zamieszczone są w pkt. 6 recenzji. Wspomniane błędy, a pojawia się ich w pracy bardzo dużo, odnoszą się jednak głównie do formy, a nie do treści i zdaniem recenzenta nie powinny mieć znacznego wpływu na

całościową ocenę dokonań Autora zaprezentowanych w rozprawie i jego wkładu w dyscyplinę nauki informatyka techniczna i telekomunikacja.

4) Mocne strony rozprawy

Rozprawa ma dwie mocne strony, tj. model matematyczny i symulacyjny systemów klasy SOA oraz środowisko badawcze zaimplementowane przez Autora.

Model matematyczny i jego implementacja czyli model symulacyjny są kluczowymi elementami zaproponowanej w pracy oryginalnej metody efektywnego rozmieszczenia komponentów oraz sterowania ich obciążeniem w systemach klasy SOA. Największą zaletą modeli jest ich uniwersalność, która pozwala na definicję a następnie symulację działania dowolnego systemu SOA. Co więcej produkują one różne parametry wydajnościowe, czasowe czy odnoszące się do wielkości używanych zasobów. Oznacza to, iż dzięki obu modelom jesteśmy w stanie gruntownie zbadać istniejące systemy a także dokonać ich rekonfiguracji w celu poprawy osiągnięć. Możemy również z wyprzedzeniem optymalnie zaprojektować nowe rozwiązania unikając w ten sposób błędów mogących wpłynąć na sukces przedsięwzięcia biznesowego zaimplementowanego w postaci SOA.

Oba modele pokazały również gruntowną wiedzę Autora dotyczącą zarówno budowy jak i funkcjonowania systemów SOA. Dzięki tej wiedzy Autor był w stanie zawrzeć w modelach wszystkie elementy mające wpływ na parametry tych systemów istotne z punktu widzenia ogólnie pojętej efektywności działania.

System badawczy zaimplementowany przez Autora to kolejny mocny punkt rozprawy. Z jednej strony, wartością samą w sobie jest fakt samodzielnego zaprojektowania i implementacji tego systemu. Autor decydując się na takie rozwiązanie wykazał, że oprócz zdolności ściśle matematycznych niezbędnych każdemu naukowcowi-inżynierowi ma również umiejętności projektowe oraz implementacyjne charakterystyczne dla zawodu informatyka. Jest to bardzo ważne, gdyż świadczy o tym, że Autor jest w stanie samodzielnie poszukiwać rozwiązań postawionych problemów badawczych, jest w stanie zaimplementować opracowane rozwiązania, a następnie dokonać ich weryfikacji, co charakteryzuje każdego kompleksowego badacza.

Dodatkowym mocnym punktem opracowanego systemu badawczego jest to, iż stanowi on skalowalne narzędzie, które po pewnych zmianach może być zastosowane w połączeniu z dowolnym algorytmem optymalizacji. W pracy Autor proponuje użycie algorytmu genetycznego razem z prostą metodą pełnego przeglądu przestrzeni rozwiązań, jednak optymalizacja SOA może być z powodzeniem zrealizowana każdą inną metodą optymalizacji dyskretną. System badawczy zaprojektowany i zaimplementowany przez Autora pozwala na łatwą reorganizację i zastąpienie aktualnie stosowanych rozwiązań innymi.

5) Słabe strony rozprawy

Zdaniem recenzenta, rozprawa ma dwa słabe punkty, z których pierwszym jest brak punktu odniesienia dla proponowanej metody optymalizacji natomiast drugim jest część rozwiązań zastosowanych w tej metodzie.

W pracy postawiono następującą tezę: *„Zastosowanie metody optymalizującej alokację komponentów do serwerów oraz wybór najlepszych algorytmów harmonogramowania i komponowania usług pozwoli na uzyskanie rozwiązań wymagających mniejszych zasobów sprzętowych oraz dających większą wartość użytkownikom systemu poprzez krótszy czas realizacji procesów, większą niezawodność i przewidywalność wykonania procesów biznesowych”*. Niestety wydaje się, iż postawiona teza jest niemożliwa do udowodnienia. Wynika to z tego, iż nie wiadomo co stanowi punkt odniesienia, do czego proponowane rozwiązanie ma być porównane żeby być w stanie stwierdzić, że zużywane zasoby są mniejsze, niezawodność większa a czasy realizacji procesów krótsze.

Jeśli założymy, iż domyślnym punktem odniesienia jest albo ręczna metoda konfiguracji systemu SOA albo dowolnie wybrana metoda „state-of-the-art” to do wykazania prawdziwości powyższej tezy konieczne byłoby porównanie osiągnięć tych metod z osiągnięciami metody proponowanej przez Autora. Niestety w pracy brak takiego porównania. W konsekwencji nie jest możliwa rzetelna ocena metody i jej usytuowanie na tle innymi rozwiązań.

Inna opcją w zakresie udowodnienia tezy rozprawy jest wykazanie, iż metoda zaproponowana w pracy jest w stanie generować ściśle optymalne rozwiązania. W takim przypadku nie jest potrzebny żaden punkt odniesienia, nie musimy przeprowadzać żadnych porównań bo nic nie jest w stanie zaproponować niczego lepszego niż ścisłe optimum. Wydaje się, iż właśnie tą drogą podążył Autor. Niestety również w tym przypadku trudno mówić o bezdyskusyjnym wykazaniu prawdziwości tezy. W rozdziale 7 rozprawy Autor zawarł badania zbieżności proponowanej przez siebie metody w dziedzinie rozwiązań jak i ich ocen. Jak się okazało optymalne rozwiązania generowane w różnych przebiegach symulacji różnią się od siebie co oznacza, iż metoda nie prowadzi do jednego ścisłego optimum a kończy działanie w wielu lokalnych optimach co zresztą sam Autor zauważył w pracy.

Inaczej sytuacja wygląda w dziedzinie ocen cząstkowych. W tym przypadku, badania pokazały, iż finalne rozwiązania wielu uruchomień procesu optymalizacji są bardzo do siebie podobne. Z jednej strony, taki rezultat może sugerować przynależność wyników końcowych do frontu Pareto czyli generowanie rozwiązań Pareto-optymalnych, co z kolei oznaczałoby wykazanie prawdziwości tezy rozprawy, jednak z drugiej strony nie ma żadnej pewności, że tak właśnie jest, i że w rozważanym problemie nie jesteśmy w stanie znaleźć innych lepszych rozwiązań niż te wyznaczone w trakcie symulacji. Aby zwiększyć taką pewność wystarczyłoby porównać opracowaną przez Autora metodę z innymi metodami optymalizacji o potwierdzonej reputacji.

Tak jak już wspomniano wcześniej, drugim słabym punktem rozprawy jest sama metoda optymalizacji a właściwie część rozwiązań w niej zastosowanych. Pierwszym z nich, które budzi poważne wątpliwości jest metoda selekcji i budowy populacji rozwiązań w kolejnych iteracjach algorytmu genetycznego. Zastosowane w pracy rozwiązanie jest ekstremalnie elitystyczne czyli silnie promujące najlepsze osobniki. Nie dość, że osobniki te z dużym prawdopodobieństwem pozostają w niezmiennym stanie w kolejnych iteracjach to dodatkowo są one rodzicami osobników potomnych zastępujących najgorsze osobniki usunięte z populacji. Takie podejście sprawia, iż proponowany algorytm można uznać za populacyjną formę metody wspinaczki górskiej. Konsekwencją jest szybka zbieżność algorytmu do optimum lokalnego, co potwierdzone zostało zresztą w badaniach. Nie jest możliwa w tym przypadku skuteczna eksploracja obszarów przestrzeni rozwiązań nie reprezentowanych w początkowej puli (populacji) przydziałów komponentów do serwerów. Każdy nowy przydział, występujący z dala od dotychczas zbadanych rejonów przestrzeni przydziałów i będący na przykład wynikiem krzyżowania nie przetrwa w kolejnej iteracji, gdyż zazwyczaj występuje on w losowym miejscu i nie miał do tej pory czasu zostać lokalnie ulepszony, na przykład poprzez użycie mutacji. W konsekwencji, nowe, być może obiecujące, przydziały będą powstawały, jednak natychmiast będą „umierały”. Takie podejście jest mało efektywne i raczej nie powinno się sprawdzać w bardziej złożonych problemach optymalizacji systemów SOA.

Kolejnym mankamentem zaproponowanej metody jest procedura naprawy rozwiązań nie spełniających ograniczeń. Zgodnie z treścią rozprawy procedura ta zastosowana jest wyłącznie podczas generowania populacji początkowej co oznacza, że rozwiązania nie spełniające ograniczeń, powstałe w kolejnych iteracjach algorytmu w wyniku zastosowania operatorów genetycznych, nie są naprawiane i algorytm traktuje je tak samo jak rozwiązania poprawne.

Sama procedura naprawy, nawet gdyby była stosowana w stosunku do każdego wadliwego rozwiązania również budzi wątpliwości. Każda taka procedura powoduje dryf algorytmu w kierunku obszarów, w których generowane są naprawiane rozwiązania. Co więcej, pewne obszary są w takim przypadku dla algorytmu niedostępne nawet jeśli w ich pobliżu występują rozwiązania poprawne o wysokiej jakości. Efektem takiego podejścia jest generalnie utrudniona eksploracja przestrzeni rozwiązań i utykanie algorytmu w minimach lokalnych. Lepsze wyniki w takim przypadku uzyskuje się zazwyczaj poprzez karanie rozwiązań wadliwych co zmusza algorytm do samodzielnego poszukiwania rozwiązań poprawnych.

Kolejnym elementem proponowanej metody, który budzi pewne zastrzeżenia jest skalaryzacja problemu wielokryterialnego zastosowana w procedurze oceny rozwiązań. Zastrzeżenia nie wynikają z samego sposobu redukcji zadania wielokryterialnego do zadania z jednym kryterium a z celowości takiego zabiegu. Po pierwsze, pomimo tego, iż ocena rozwiązań przeprowadzana jest na podstawie skumulowanego kryterium to i tak algorytm zachowuje w pamięci rozwiązania niezdominowane

problemu wielokryterialnego. Po drugie, po co używać jednokryterialnych algorytmów genetycznych skoro istnieją algorytmy wielokryterialne bazujące na relacji dominacji. I w końcu po trzecie, wydaje się, iż zastosowana procedura skalaryzacji może być jednym z powodów, dla którego finalne rozwiązania wielu symulacji posiadały podobne do siebie oceny cząstkowe.

6) Uwagi szczegółowe

Niniejsza część recenzji zawiera uwagi szczegółowe, które zostały przedstawione w kolejności powstawania tych uwag w trakcie lektury rozprawy czyli od uwag z początkowych fragmentów rozprawy do uwag dotyczących stron końcowych:

1. Str. 25: „W jej ramach zostanie zastosowany algorytm genetyczny pozwalający na przeszukiwanie przestrzeni rozwiązań alokacji oraz symulator działania systemów SOA” – w rozprawie zaproponowano algorytm optymalnej alokacji komponentów oraz wyboru optymalnych algorytmów komponowania i harmonogramowania usług a nie tylko algorytm alokacji.
2. Duża liczba błędów językowych, oprócz tzw. literówek, pojawiają się również błędy interpunkcyjne i logiczne, które wpływają negatywnie na spójność tekstu i utrudniają jego rozumienie, np. str. 26 – „W konstruowania takiej metody zostały opracowane”, str. 74 – „Rozwiązania, w których kolejki były krótkie symulowane, były znacznie szybciej niż te, w których kolejki były...”, str. 75 – „Jest to najprawdopodobniej spowodowane dużą liczbą mierzonych charakterystyk, a przez to uzyskiwaniem podobnych wymagań od systemu w ujęciu globalnym” (czy można uzyskać wymagania od systemu?).
3. Procesy biznesowe są w modelu matematycznym reprezentowane jako grafy. Dlaczego w pracy pojawiły się przykładowe procesy symulowane w trakcie badań w postaci właśnie grafów lub BPMN? Umożliwiłoby to czytelnikowi wizualną ocenę złożoności problemu, z którym miał do czynienia proponowany algorytm.
4. Str. 36: redefinicja P^S . Symbol ten oznacza zarówno wektor wielkości pamięci operacyjnej serwerów wyrażonej w liczbie megabajtów jak i macierz pokazującą przepustowość transferu danych pomiędzy serwerami.
5. Str. 36, wzór (17). Jaki sens ma definiowanie S^E jeśli brak szyny powoduje brak możliwości obsługi usług na danym serwerze? To tak jakby tego serwera nie było, nie jest on dostępny jako element SOA.
6. Str. 38, wzór (22). Zgodnie z tym wzorem, zmiennymi decyzyjnymi w procesie optymalizacji są: przydział komponentów do serwerów, wybór algorytmów harmonogramowania i komponowania ale również to czy do danego serwera jest podpięta szyna ESB!
7. Str. 38, wzory (23), (24). W podanych ograniczeniach uwzględnione są wyizolowane komponenty oraz środowiska uruchomieniowe. Wydaje się, że brak w tych ograniczeniach mocy serwerów i ich pamięci zużywanych przez usługi, szynę ESB, czy też zapytania od silnika BPM.
8. Str. 39, wzór (26). Zgodnie z tekstem rozprawy jest to średni oczekiwany czas. Czy nie powinien to być po prostu oczekiwany czas?
9. Str. 39, wzory (26), (27). Czym jest w tych wzorach X?
10. Str. 39, 40, kryteria k3 i k4. Oba kryteria są tak samo nazwane a mają zupełnie inną definicję.
11. Kryterium k3 i k5 są niepotrzebne w pracy, zastąpiono je kryteriami k4 i k6.
12. Str. 41, redefinicja k6. Symbol ten oznacza zarówno oczekiwany stopień wykorzystania przydzielonych zasobów pamięci na rzecz realizacji usług w czasie t (wzór 32), jak i liczbę niezrealizowanych procesów w systemie (wzór 33).
13. Str. 41, wzór (34). Które k6 jest brane pod uwagę, czy to zdefiniowane w (32) czy też to zdefiniowane w (33)?
14. Zdaniem recenzenta zastosowanie języka Java z jego maszyną wirtualną w procesie złożonych symulacji podczas, których wykorzystuje się techniki wymagające obliczeniowo takie jak algorytmy genetyczne nie jest najlepszym rozwiązaniem. Jak autor sam zauważył

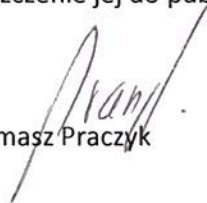
w tak złożonych problemach jak optymalizacja SOA powinno się dążyć do użycia GPU i obliczeń równoległych.

15. Str. 44: Poszukiwanie optymalnego zestawienia algorytmów harmonogramowania i komponowania usług oraz alokacji komponentów do serwerów jest realizowane w dwóch zagnieżdżonych pętlach, w których każda pętla odpowiada jednemu zestawieniu powyższych algorytmów i jednemu uruchomieniu algorytmu genetycznego. Oznacza to, iż jeden przebieg całego algorytmu optymalizacji wymaga wielu uruchomień algorytmu genetycznego. Jest to rozwiązanie bardzo czasochłonne, tym bardziej jeśli implementacja algorytmu jest sekwencyjna. Wydaje się, iż zdecydowanie lepszym rozwiązaniem jest zakodowanie typów algorytmów harmonogramowania i komponowania bezpośrednio w chromosomie. Mała liczba rozważanych algorytmów spowoduje wydłużenie chromosomów o bardzo małą liczbę genów nie powodując znacznego rozszerzenia przeszukiwanej przestrzeni rozwiązań i tym samym nie utrudniając znacznie zadania algorytmowi genetycznemu.
16. Str. 46: „Symulacja genu” – symulować możemy co najwyżej działanie systemu SOA zakodowanego w genotypie, co więcej gen to pojedyncza wartość genotypu.
17. Str. 58. Wyniki symulacji były uśredniane dla 10 uruchomień – to bardzo mało aby uzyskać wiarygodne dane statystyczne.
18. Str. 58. W większości symulacji maksymalna liczba iteracji algorytmu genetycznego to 1000. W przypadku trudnych problemów optymalizacyjnych to bardzo mało. Co więcej, jeśli po 1000 iteracji rozwiązania generowane przez algorytm się zmieniały lub zmiana była bardzo mała to oznacza szybką zbieżność algorytmu do jednego rozwiązania (małą różnorodność materiału genetycznego w populacji uniemożliwiająca w praktyce jakkolwiek postęp w poszukiwaniach nowych lepszych rozwiązań) o czym wspomniano już w recenzji. Mogą to potwierdzać również wyniki dla zwiększonej liczby iteracji oraz powiększonej populacji rozwiązań, dla których poprawa w stosunku do oryginalnych ustawień algorytmu była znikoma. Nie ma po prostu sensu przedłużać działania algorytmu w sytuacji stagnacji ewolucji wokół jednego lub kilku punktów.
19. Str. 67. Badanie wrażliwości algorytmu przeprowadzone zostało dla zmian w zakresie od 1 do 5%. Te zmiany mogły nie powodować żadnych modyfikacji oryginalnego problemu, np. zwiększenie liczby serwerów o największej mocy o 5% czyli z wartości 5 do 5.25 oznacza praktycznie brak jakiegokolwiek zmiany w tym zakresie – po zaokrągleniu otrzymujemy ponownie wartość 5.
20. Str. 73: „Większa liczba przeanalizowanych genotypów sprawia, że zwiększa się prawdopodobieństwo znalezienia rozwiązań bliskich optymalnym” – samo zwiększenie wielkości populacji nie spowoduje zwiększenia prawdopodobieństwa odszukania rozwiązań bliskich optymalnym. Rozwiązania mogą być źle zainicjowane, a także mogą być nieefektywnie przetwarzane w kolejnych iteracjach co nawet przy bardzo dużej liczbie osobników w populacji nie poprawi w żaden sposób efektywności algorytmu.
21. Str. 76: „Wydajność metody jest zatem na zadowalającym poziomie w odniesieniu do prezentowanego zadania o dużym rozmiarze” lub w podsumowaniu „Wydajność implementacji metody była wystarczająca do optymalizacji problemu dużej wielkości” – co to znaczy „na zadowalającym poziomie” czy też co to znaczy, że wydajność metody jest wystarczająca? Żeby takie zdania były uprawnione konieczne byłoby zdefiniowanie zadowalającej wydajności. Po raz kolejny brak w tym przypadku jakiegokolwiek poziomu odniesienia.

7) Konkluzja

Pomimo wszystkich wymienionych wyżej uwag, przedstawioną do recenzji rozprawę oceniam pozytywnie. Stwierdzam, że przedmiotowa rozprawa doktorska spełnia wymogi przewidziane w art. 13 ustawy przywołanej w pkt.1 niniejszej recenzji - jest oryginalnym rozwiązaniem problemu naukowego oraz wykazuje ogólną wiedzę teoretyczną Doktoranta w uprawianej dyscyplinie naukowej

(Informatyka techniczna i telekomunikacja), jak i umiejętność samodzielnego prowadzenia pracy naukowej. Poziom merytoryczny i formalny rozprawy jest w znacznej jej części wysoki, co pozwala na jej ocenę pozytywną. Zasadne jest więc przyjęcie rozprawy i dopuszczenie jej do publicznej obrony.



Tomasz Praczyk

