

AUTOREFERAT

1. Imię i Nazwisko

Adam Piotrowski

2. Posiadane dyplomy, stopnie naukowe/artystyczne – z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej

2001 - magister w zakresie geografii fizycznej, Wydział Geografii i Studiów Regionalnych, Uniwersytet Warszawski.

2006 - doktor w zakresie Nauk o Ziemi, Instytut Geofizyki Polskiej Akademii Nauk, tytuł rozprawy doktorskiej: Inteligentna analiza danych hydrologicznych.

3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych/ artystycznych

2001-2006 – studia doktoranckie w Instytucie Geofizyki Polskiej Akademii Nauk.

2006-2006 – asystent w Zakładzie Zasobów Wodnych Instytutu Geofizyki Polskiej Akademii Nauk.

2006-nadal – adiunkt w Zakładzie Zasobów Wodnych Instytutu Geofizyki Polskiej Akademii Nauk, a po zmianie nazwy zakładu – w Zakładzie Hydrologii i Hydrodynamiki Instytutu Geofizyki Polskiej Akademii Nauk.

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 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 ze zm.):

a) tytuł osiągnięcia naukowego/ artystycznego

Rozwój Algorytmów Ewolucyjnych i ich wykorzystanie do prognozowania zmiennych hydrologicznych

b) autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa

[1] Piotrowski, A.P. (2013) Adaptive Memetic Differential Evolution with Global and Local neighborhood-based mutation operators. Information Sciences 241, 164-194.

[2] Piotrowski, A.P., Napiorkowski, J.J. (2013) A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modeling. Journal of Hydrology 476, 97-111.

[3] Piotrowski, A.P., Napiorkowski, J.J. (2012) Product-Units neural networks for catchment runoff forecasting. Advances in Water Resources 49, 97-113.

[4] Piotrowski, A.P., Rowinski, P.M., Napiorkowski, J.J. (2012) Comparison of evolutionary computation techniques for noise injected neural network training to estimate

longitudinal dispersion coefficients in rivers. *Expert Systems with Applications* 39, 1354-1361.

[5] Piotrowski, A.P., Napiorkowski, J.J., Kiczko, A. (2012) Differential Evolution algorithm with separated groups for multi-dimensional optimization problems. *European Journal of Operational Research* 216, 33-46.

do pracy tej niezbędna była korekta, która ukazała się w tym samym czasopiśmie:

Piotrowski, A.P., Napiorkowski, J.J., Kiczko, A. (2012) Corrigendum to: "Differential Evolution algorithm with separated groups for multi-dimensional optimization problems" [*Eur. J. Oper. Res.* 216 (2012) 33–46]. *European Journal of Operational Research* 219(2), 488.

[6] Piotrowski, A.P., Napiorkowski, J.J. (2011) Optimizing neural networks for river flow forecasting – Evolutionary Computation methods versus Levenberg–Marquardt approach. *Journal of Hydrology* 407, 12-27.

[7] Piotrowski, A.P., Napiorkowski, J.J. (2010) Grouping Differential Evolution algorithm for multi-dimensional optimization problems. *Control and Cybernetics* 39(2), 527-550.

[8] Rowinski, P.M., Piotrowski, A. (2008) Estimation of parameters of transient storage model by means of multi-layer perceptron neural networks. *Hydrological Sciences Journal* 53(1), 165-178.

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

Dyscypliny naukowe zajmujące się wodą, takie jak hydrologia, hydraulika, zasoby wodne, glaciologia lub oceanologia, są czasami uważane za pogranicze geofizyki, co nie zmienia faktu, że hydrosfera jest niezwykle ważnym elementem planety Ziemi. Wśród dyscyplin zajmujących się wodą na Ziemi hydrologia należy prawdopodobnie do najczęściej wymienianych, choćby dlatego, że zajmuje się problemami, które interesują wielu ludzi na co dzień, jak powodzie, susze czy jakość wody. Głównym celem hydrologii jest nie tylko wyjaśnienie pewnych zjawisk, ale także ich prognozowanie, oszacowanie lub projekcja, co wymaga stosowania różnego typu uproszczonych modeli fizycznych, konceptualnych lub opartych na danych (ang. data-based models). Modele zazwyczaj wymagają kalibracji szeregu parametrów, których wartości z różnych przyczyn nie mogą być pomierzone lub oszacowane przez eksperta. W związku z tym jakość wyników uzyskiwana za pomocą uproszczonych modeli fizycznych lub modeli konceptualnych zależy w znacznej mierze nie tylko od tego, jak dobrze przybliżają one rzeczywiste procesy, ale także od właściwego wyboru procedur optymalizacyjnych. W przypadku modeli opartych na danych, których zastosowania w dyscyplinach zajmujących się wodą bardzo się upowszechniły w ciągu ostatnich 20 lat, właściwy wybór metod optymalizacji jest jeszcze istotniejszy.

Moje główne zainteresowania naukowe, a także główny cel naukowy prac składających się na osiągnięcie naukowe, to znalezienie lub zaproponowanie jak najodpowiedniejszych metod optymalizacji dla modeli opartych na danych, które są wykorzystywane przy rozwiązywaniu wybranych problemów hydrologicznych. Jest to cel łączący trzy elementy: hydrologię, modele i procedury optymalizacyjne. Osiem prac składających się na osiągnięcie naukowe może być podzielonych na dwie grupy: pięć z nich poświęconych jest bezpośrednio zastosowaniu szeregu metod optymalizacyjnych w celu rozwiązania problemów hydrologicznych [2,3,4,6,8], trzy pozostałe [1,5,7] poświęcone są prezentacji

zaproponowanych przeze mnie nowych procedur optymalizacyjnych należących do tak dzisiaj popularnych Algorytmów Ewolucyjnych. Ponieważ do rozmaitych problemów hydrologicznych stosowane są różnego typu modele bazujące na danych, oraz z uwagi na olbrzymią liczbę procedur optymalizacyjnych zaproponowanych dotychczas, zakres moich prac musiał być oczywiście zawężony do wybranych zagadnień i typów metod. W pracach, które składają się na osiągnięcie naukowe, rozważane były dwa problemy hydrologiczne:

1. modelowanie odpływu ze zlewni [2,3,6] na przykładzie górnej części zlewni rzeki Annapolis w Nowej Szkocji (Kanada), położonej w umiarkowanej chłodnej strefie klimatycznej;

2. modelowanie transportu zanieczyszczeń w rzekach [4,8].

Oba problemy zostaną omówione dokładniej w części referatu poświęconej prezentacji poszczególnych publikacji.

Stosowane modele to przede wszystkim niezwykle popularne w różnych dziedzinach nauki wielowarstwowe perceptronowe sieci neuronowe [2,3,4,6,8] oraz mało znane sieci neuronowe typu Product-Units [3]. Inne modele, takie jak regresja liniowa lub konceptualny model HBV, są również stosowane w części prac [8,3] jako metody klasyczne, pozwalające na ocenę wyników uzyskiwanych za pomocą sieci neuronowych.

Wartości parametrów sieci neuronowych są najczęściej optymalizowane (w literaturze używa się też pojęcia „uczenie” lub „trening” sieci neuronowych) za pomocą algorytmów gradientowych. Algorytmy gradientowe, jak na przykład bardzo popularny algorytm Levenberga-Marquardta (Hagan i Menhaj, 1994; Press i in., 2006) są szybkie, łatwe w implementacji i zazwyczaj prowadzą do zadawalających wyników. Niemniej w literaturze bardzo często spotyka się argumentację, że inne metody optymalizacji, niewymagające obliczania pochodnych funkcji celu, są niezbędne do uczenia sieci neuronowych, ponieważ: 1. metody gradientowe nie mają zdolności przeszukiwania globalnego i znajdują jedynie optima lokalne, 2. z różnych przyczyn stosowana funkcja celu może być nieróżniczkowalna (jak np. w przypadku pracy [4]). W związku z tym pokusa, by do treningu sieci neuronowych wykorzystać metody zwane ogólnie metaheurystykami, spośród których dziś szczególną popularnością cieszą się Algorytmy Ewolucyjne, jest zrozumiała i nie nowa. Problematyka uczenia sieci neuronowych za pomocą Algorytmów Ewolucyjnych była poruszana w wielu pracach w ciągu ostatnich dwudziestu kilku lat (Whitley i in., 1990; Yao, 1993; Branke, 1995; Yao i Liu, 1997; Yao, 1999; Cantu-Paz i Kamath, 2005; Ding i in., 2013 i wiele innych). Ponadto Algorytmy Ewolucyjne były czasami wykorzystywane nie tylko do uczenia, ale także do znalezienia odpowiedniej architektury sieci neuronowych (Yao, 1999; Stanley i Miikkulainen, 2002; Islam i in., 2009; Hunter i in., 2012), niemniej w przypadku problemów hydrologicznych to zastosowanie Algorytmów Ewolucyjnych nie wydaje się być szczególnie istotne. Ponieważ w literaturze zaproponowanych zostało bardzo wiele metaheurystyk, które według części autorów mogą mieć bardzo szerokie zastosowania (Fogel, 2000), problem uczenia sieci neuronowych był czasami sugerowany jako jedna z metod oceny działania różnych algorytmów optymalizacji (He i in., 2009). Chociaż poszukiwania właściwych metaheurystyk do uczenia sieci neuronowych trwają co najmniej od lat 80-tych dwudziestego wieku, kwestia ta pozostaje otwarta z dwóch co najmniej powodów: 1. w większości prac liczba porównywanych metod optymalizacyjnych jest bardzo mała, co uniemożliwia wyciąganie zbyt daleko idących wniosków; 2. wnioski z poszczególnych prac są często ze sobą sprzeczne, co może być zarówno efektem błędów, jak i różnych założeń przyjmowanych w poszczególnych pracach.

W ciągu ostatnich dwudziestu lat bardzo popularne stały się tak zwane Obliczenia Ewolucyjne (ang. Evolutionary Computation), czyli metody inspirowane ewolucją biologiczną. Obejmują one wiele rodzin metod optymalizacji, między innymi Strategie Ewolucyjne (ang. Evolution Strategies) (Bäck i Schwefel, 1993; Hansen i Ostermaier, 1996),

Algorytmy Genetyczne (ang. Genetic Algorithms) (Holland, 1975), Programowanie Genetyczne (ang. Genetic Programming) (Koza, 1992), Differential Evolution (DE) (Storn i Price, 1995) (w polskiej literaturze od niedawna funkcjonuje termin Ewolucja Różnicowa, niemniej na potrzeby niniejszego referatu pozostanę przy bardziej popularnym terminie angielskim) i szereg innych metod (Onwobulu i Babu, 2004). Do metod pokrewnych należą równie popularne w ostatnich latach metody Inteligencji Roju (ang. Swarm Intelligence) inspirowane zachowaniami stadnymi organizmów żywych, takie jak Optymalizacja Rojem Cząstek (Particle Swarm Optimization) (Eberhart i Kennedy, 1995), Algorytmy Mrówkowe (Ant Colony Optimization) (Dorigo i in., 1996), metody Cat Swarm (Chu i in., 2006) i Group Search Optimizers (He i in., 2009). Niektóre metody optymalizacji są inspirowane innymi zachowaniami organizmów żywych, jak na przykład Biogeography-based Algorithm (Simon i in., 2008) i Cuckoo Search Approach (Yang i Deb, 2009). Oczywiście nie wszystkie metaheurystyki odnoszą się do inspiracji biologicznych. Bardzo dużą popularnością cieszą się od dawna algorytm Symulowanego Wyżarzania (Kirkpatrick i in., 1983), ostatnio powstaje coraz więcej algorytmów, które – zdaniem ich autorów – są inspirowane przez niebiologiczne procesy, zjawiska, prawa lub nawet poglądy filozoficzne, np. grawitację (Rashedi i in., 2009), reakcje chemiczne (Lam i Li, 2010), kooperację (Masegosa i in., 2013; Civicioglu, 2013), koncept Brzytwy Ockhama (Iacca i in., 2012; Caraffini i in., 2013) czy nawet harmonię muzyki (Geem i in., 2001). Niemniej nie wszystkie nowo proponowane metaheurystyki są warte zainteresowania i praktycznego zastosowania (Crepisek i in., 2012). Przy dokonywaniu wyboru metody należy uważać na jej szczególne cechy, które mogą okazać się skuteczne jedynie przy rozwiązywaniu części problemów, w innych zaś stanowić raczej pułapkę (Weise i in., 2012). Jest to szczególnie ważne, gdy metoda ma być zastosowana do rozwiązywania problemów, na których dotychczas nie była testowana. Ponadto, wiele metod zaproponowanych w ciągu ostatnich dwudziestu lat wykazuje częściowe podobieństwo do starszych algorytmów, zwanych metodami bezpośrednimi (ang. Direct Search) (Kolda i in., 2003), takich jak algorytm Nelder-Meada (Nelder i Mead, 1965), algorytm Rosenbrocka (Rosenbrock, 1960) lub Controlled Random Search (Price, 1977). Reasumując, mimo że dzisiaj mamy do wyboru olbrzymią liczbę metaheurystyk, większość algorytmów jest słabo umotywowana, nie ma dowodów zbieżności, ich zachowanie w trakcie rozwiązywania zadania optymalizacyjnego jest niewystarczająco jasne, a do tego wiele metod wykazuje znaczne wzajemne podobieństwo bez wyraźnego uzasadnienia. W efekcie wybór właściwych metaheurystyk do konkretnego zadania nie jest łatwy.

Można postawić pytanie, dlaczego powstało tak wiele metaheurystyk? Generalnie metaheurystyki są użyteczne przede wszystkim do rozwiązywania problemów wielomodalnych, mających wiele optimum lokalnych. W innych wypadkach stosuje się raczej metody gradientowe (jeśli funkcja jest różniczkowalna), lub znane od wielu lat szybkie metody bezpośrednie. Gdy liczba optimum lokalnych jest duża, olbrzymiego znaczenia nabiera problem właściwego zrównoważenia zdolności eksploatacyjnych (znajdywania optimum lokalnego w najbliższym rejonie przestrzeni poszukiwań) i eksploracyjnych (znajdowania obiecujących obszarów w odleglejszych częściach przestrzeni poszukiwań) algorytmu. Ponieważ jest to zagadnienie trudne i zwykle silnie zależne od rozwiązywanego problemu, motywowało ono powstanie wielu różnego typu metaheurystyk – w ostatnich latach także takich, które mają zdolności samoadaptacyjne, dzięki czemu powinny być bardziej elastyczne i nadawać się do zastosowania do większej liczby problemów praktycznych.

Oczywiście przetestowanie wszystkich metaheurystyk, czy choćby Algorytmów Ewolucyjnych, nie jest możliwe. W swoich badaniach musiałem dokonać wstępnej selekcji i skoncentrować się na pewnych typach metod. Szczególną uwagę poświęciłem metodom z

rodziny Differential Evolution, z uwagi na ich rosnącą popularność i specyficzne cechy zachowania tych algorytmów w trakcie przeszukiwania przestrzeni poszukiwań, które pozwalają na – do pewnego stopnia automatyczne – równoważenie procesów eksploracyjnych i eksploatacyjnych. W swoich pracach wykorzystywałem również wybrane metody Inteligencji Roju, Strategii Ewolucyjnych, multi-algorytmów oraz klasyczne metody przeszukiwania bezpośredniego i algorytmy gradientowe. Zaproponowałem trzy nowe Algorytmy Ewolucyjne z rodziny Differential Evolution, dwa z nich zostały opublikowane w wysokopunktowanych czasopismach notowanych w bazie Journal Citation Reports w kategoriach Computer Science [1,5]. Część z tych algorytmów została z dużym sukcesem wykorzystana w kolejnych pracach do rozwiązywania problemów hydrologicznych.

Dyskusja zawartości poszczególnych artykułów podzielona jest na trzy podgrupy tematyczne: artykuły związane z transportem zanieczyszczeń w rzekach [8,4], artykuły dotyczące modelowania odpływu ze zlewni [6,3,2] oraz prace prezentujące nowe Algorytmy Ewolucyjne [7,5,1]. Większość tych prac porusza także dodatkowe zagadnienia, które będą pokrótce opisane poniżej.

Opis prac [8,4]

W tych dwóch pracach poruszany był problem oszacowania wartości parametrów jednowymiarowych modeli transportu zanieczyszczeń w rzekach. Praca [8] była jednocześnie pierwszą, w której wykorzystałem różne metaheurystyki w celu rozwiązania postawionego problemu hydrologicznego.

Większość modeli transportu zanieczyszczeń wymaga znajomości wartości jednego lub kilku parametrów reprezentujących morfologiczne lub hydrauliczne cechy danego odcinka rzeki. Zazwyczaj uzyskanie ich wartości za pomocą pomiarów bezpośrednich jest kosztowne i czasochłonne, często wymaga przeprowadzenia eksperymentu znacznikowego. W związku z tym, dla większości takich parametrów zaproponowano szereg formuł empirycznych (Cheong i in., 2007; Wallis i Manson, 2004; Deng i in., 2002) pozwalających na ich oszacowanie na podstawie stosunkowo łatwo mierzalnych cech danego odcinka rzeki. W ciągu ostatniego dziesięciolecia w wielu pracach zaproponowano wykorzystanie do tego celu szeregu modeli opartych na danych, w tym sieci neuronowych. Niemniej wybór zmiennych wejściowych do formuły empirycznej czy modelu bazującego na danych, czyli takich cech danego odcinka rzeki, które będą miały istotny wpływ na wartość szacowanego parametru, a jednocześnie będą rzeczywiście stosunkowo łatwe do uzyskania, nie jest sprawą prostą. Jest to częściowo spowodowane małą liczbą dostępnych danych eksperymentalnych.

W pracach [8,4] perceptronowe sieci neuronowe są wykorzystywane do szacowania parametrów dwóch różnych modeli transportu zanieczyszczeń. Uzyskane wyniki są porównywane z oszacowaniami otrzymywanymi za pomocą klasycznych formuł empirycznych. Artykuł [4] dotyczy określania jednego tylko parametru, współczynnika dyspersji podłużnej, niezbędnego do zastosowania najprostszego równania adwekcji-dyspersji (Taylor, 1953) na danym odcinku rzeki. Praca [8] dotyczy oszacowań wartości trzech parametrów modelu martwych stref (ang. transient storage model) (Bencala i Walters, 1983), który również jest często używany do określania transportu zanieczyszczeń w rzekach.

W pracy [8] porównane zostały wyniki uzyskane z sieci neuronowych uczonych za pomocą trzech metod optymalizacji. Wykorzystano algorytm Levenberga-Marquardta, który jest jedną z najpopularniejszych metod uczenia sieci neuronowych, oraz dwie podstawowe wersje metaheurystyk, które stają się niezwykle popularne w różnych dziedzinach nauki: Optymalizacji Rojem Częstek i Differential Evolution. Główne wyniki pracy można podsumować następująco: 1. Perceptronowe sieci neuronowe pozwalają na wyraźną poprawę

jakości uzyskiwanych wyników w stosunku do formuł empirycznych lub klasycznej regresji liniowej. Wniosek ten dotyczy oszacowań każdego z trzech parametrów modelu martwych stref. Niemniej dokładność oszacowań, szczególnie w odniesieniu do parametru opisującego czas przebywania cząstki w martwej strefie, jest wciąż niewystarczająca do zastosowania metody w praktyce. Oszacowania współczynnika dyspersji podłużnej i powierzchni martwych stref są dokładniejsze, niemniej wartości wszystkich trzech parametrów są niezbędne do uruchomienia modelu. Wydaje się, że bez znacznego zwiększenia liczby odcinków rzek, dla których dostępne są dane eksperymentalne, znacząca poprawa jakości wyników nie będzie mogła być osiągnięta. 2. Na podstawie uzyskanych wyników podstawowa wersja algorytmu Differential Evolution wydawała się konkurencyjną metodą optymalizacji w stosunku do algorytmu Levenberga-Marquardta. Niemniej, mała liczba dostępnych danych mogła znacząco wpłynąć na taką konkluzję, niepotwierdzoną zresztą ani w moich innych pracach, ani w pracach większości innych autorów (które jednak dotyczyły zastosowań podobnej metodyki do innych zagadnień).

W pracy [4] wykorzystano szereg Algorytmów Ewolucyjnych do uczenia sieci neuronowych w celu oszacowania współczynnika dyspersji podłużnej w rzekach. Dodatkowym celem pracy było przetestowanie techniki dodawania szumu do sieci neuronowej, mającej na celu zapobieganie tak zwanemu przeuczeniu. Ponieważ sieci neuronowe są uważane za przykład uniwersalnego aproksymatora (ang. universal approximator) (Hornik, 1989), są podatne na przeuczenie, innymi słowy w trakcie optymalizacji wartości parametrów mogą dopasowywać się nie tylko do sygnału, ale i do szumu zawartego w próbce uczącej. Jest kilka sposobów na zmniejszenie tego zagrożenia, jednym z nich jest metoda sztucznego dodawania szumu do próbki uczącej (ang. noise injection) (Holmstrom i Koistinnen, 1992), użyteczna szczególnie w przypadku, gdy dysponuje się małą liczbą danych. Ponieważ w pracy wykorzystano nieróżniczkowalną funkcją celu (w postaci średniego błędu absolutnego), Algorytmy Ewolucyjne zostały przetestowane na problemie, do którego rzeczywiście nie można było zastosować algorytmów gradientowych. Problem szacowania współczynników dyspersji podłużnej był przeze mnie już poruszany we wcześniejszych pracach, między innymi w moim pierwszym artykule w czasopiśmie z listy JCR (Rowiński i in., 2005) oraz w doktoracie, niemniej ani kwestia Algorytmów Ewolucyjnych, ani testowania metod zapobiegających przeuczeniu sieci nie były wówczas rozważane. W pracy [4] do treningu sieci neuronowych z uwzględnieniem dodawania szumu wykorzystanych zostało dziewięć różnych metaheurystyk, sześć wariantów algorytmów Differential Evolution, dwa warianty Optymalizacji Rojem Cząstek oraz Strategia Ewolucyjna CMA-ES. Aby uzyskać zadawalającą próbkę wyników, każdy algorytm został uruchomiony pięćdziesięciokrotnie. Różnice jakości wyników uzyskiwanych za pomocą sieci neuronowych uczonych pięcioma stosunkowo nowymi wariantami algorytmu Differential Evolution były niewielkie. Niemniej metoda GDE, zaproponowana przeze mnie w pracy [7], okazała się nieznacznie lepsza od pozostałych. Wyniki pokazały także, że nawet gdy różnice między wynikami uzyskiwanymi za pomocą najlepszych metod optymalizacji są małe, nie oznacza to, by wybór metody był mało istotny, ponieważ blisko połowa testowanych metod wypadła znacznie słabiej. Co więcej, wyniki uzyskiwane za pomocą dwóch najslabiej spisujących się metod były obciążone większym błędem niż oszacowania uzyskiwanych za pomocą dawnych formuł empirycznych. Metoda dodawania szumu pozwoliła na redukcję średniego błędu absolutnego o 2-20% dla niezależnych danych testowych (w zależności od wariantu) w stosunku do wyników uzyskiwanych za pomocą sieci neuronowych niewykorzystujących żadnej metody zapobiegania przeuczeniu.

Opis prac [6,3,2]

Modelowanie odpływu ze zlewni jest jednym spośród najpopularniejszych i najważniejszych zadań hydrologii. Artykuł Hsu i in. (1995) zapoczątkował bardzo długą listę prac poświęconych wykorzystaniu sieci neuronowych do tego celu. Niemniej, według części autorów (Abrahart i in., 2012) aplikacje sieci neuronowych do prognozowania przepływu rzeczno-odbywały się do tej pory w sposób raczej „chaotyczny”. W wielu pracach hydrologicznych można zauważyć dosyć ogólne potraktowanie szeregu szczegółów technicznych, które mogą mieć znaczący wpływ na jakość uzyskiwanych wyników. W trzech prezentowanych w niniejszej części autoreferatu [6,3,2] artykułach nacisk kładziony był zarówno na problem hydrologiczny, jak i na poszczególne aspekty metodologiczne.

Problem hydrologiczny poruszany we wszystkich trzech pracach jest podobny – dotyczy dobrego prognozowania odpływu ze zlewni górnego odcinka rzeki Annapolis położonej w Nowej Szkocji, w Kanadzie, na podstawie danych hydrometeorologicznych. Zlewnia rzeki Annapolis jest w znacznej części pokryta wzgórzami o wysokości nieprzekraczającej 300 m.n.p.m., znaczna jej część jest zalesiona. Położona jest ona w strefie klimatu wilgotnego kontynentalnego według klasyfikacji Koppena. Opady śniegu występują od listopada do kwietnia. W czasie zimowych miesięcy występują znaczne wahania temperatury, w efekcie przemienność okresów z opadami śniegu, silnymi mrozami i roztopami znacznie utrudnia modelowanie. Mimo że w okresie letnim opady występują, z uwagi na stosunkowo wysokie temperatury powietrza i znaczną ewapotranspirację, stany wód są najniższe. Mimo widocznych różnic klimatycznych między Polską a Nową Szkocją, można także zauważyć między nimi wiele podobieństw – na obu obszarach notuje się duże sezonowe wahania warunków pogodowych, a opad śniegu i jego topnienie mają znaczny wpływ na przepływy rzeczne w okresie zimowo-wiosennym. Według podziału Koppena, oba obszary znajdują się w tej samej strefie klimatycznej. Wspomniane podobieństwa oraz łatwa dostępność danych kanadyjskich były motywacją do zajęcia się tą zlewnią.

Dobowe dane hydrologiczne i meteorologiczne dla okresu 30 lat zostały uzyskane z „Water Survey of Canada” oraz „Canada’s National Climate Data and Information Archive”. Dane hydrologiczne pochodzą ze stacji umiejscowionej w Wilmot (stacja zamyka zlewnię o powierzchni 546 km²). Dane meteorologiczne, uwzględniające minimalne i maksymalne temperatury dobowe, opady deszczu i śniegu (osobno) oraz grubość pokrywy śnieżnej pochodzą ze stacji położonej 9 kilometrów na wschód, na lotnisku Greenwood Airfield. W pierwszym artykule poświęconym tej tematyce [6] wykorzystano jedynie 10-letni ciąg danych, późniejsze dwie prace powstały już w oparciu o dane 30-letnie. Czas koncentracji dla zlewni wynosi nieco poniżej 1 dnia, niemniej waha się w zależności od przyjętej metody obliczeń. W miarę odpowiedni czas koncentracji, dostępność długich ciągów danych pomiarowych prawie pozbawionych dziur oraz znaczna dobowa dynamika przepływu obserwowana od jesieni do wiosny czynią ze zlewni rzeki Annapolis dogodny poligon badawczy. Pozwolę sobie nadmienić, chociaż wykracza to poza zakres prac składających się na osiągnięcie naukowe prezentowanych w tym referacie, iż obecnie pracuję nad porównaniem prognoz odpływu uzyskiwanych za pomocą sieci neuronowych i modeli konceptualnych kalibrowanych różnymi metodami dla rzeki Annapolis i zlewni z obszaru Polski południowej.

Z metodologicznego punktu widzenia, poza porównaniem Algorytmów Ewolucyjnych, w pracach [3] i [2] poruszono także inne zagadnienia. W artykule [3] zaproponowano wykorzystanie, prawdopodobnie po raz pierwszy w hydrologii, sieci neuronowych typu Product-Units. W pracy [2] zweryfikowano zasadność stosowania kilku metod zapobiegających przeuczeniu sieci neuronowych do modelowania odpływu ze zlewni w sytuacji, gdy liczba dostępnych danych jest znaczna.

Artykuł [6] to moja pierwsza praca łącząca modelowanie odpływu ze zlewni, sieci neuronowe i Algorytmy Ewolucyjne. Wykorzystano jedynie dane 10-letnie, z czego 4 lata

stanowiły tak zwane zbiór testowy, nieużywany w trakcie kalibracji. Zastosowano szereg architektur perceptronowych sieci neuronowych o różnej liczbie parametrów, różniły się one między innymi zmiennymi wejściowymi. Do uczenia każdej architektury sieci neuronowych zastosowano algorytm Levenberga-Marquardta oraz osiem Algorytmów Ewolucyjnych. Każdy eksperyment został powtórzony pięćdziesięciokrotnie. Większość stosowanych Algorytmów Ewolucyjnych to metody nowe, znaczna ich część nie była wcześniej porównywana między sobą. W pracy pokazano wartości średnie i mediany wskaźników jakości wyników uzyskanych w ciągu 50 uruchomień każdego testowanego wariantu, gdyż ocena wyników na podstawie tych dwóch statystyk nie zawsze jest podobna. Różnice jakości wyników uzyskiwanych za pomocą sieci neuronowych trenowanych różnymi metodami optymalizacji były bardzo duże. Najlepsze wyniki uzyskiwano za pomocą dwóch metod: wariantu algorytmu Differential Evolution zwanego DEGL (Das i in., 2009) oraz algorytmu Levenberga-Marquardta. Spośród pozostałych przetestowanych metod można wyróżnić jedynie algorytm Inteligencji Roju EPUS-PSO (Hsieh i in., 2009). Większe różnice obserwowano, porównując mediany wskaźników jakości wyników niż ich wartości średnie. Spośród dwóch najskuteczniejszych metod algorytm Levenberga-Marquardta okazał się, zgodnie z przewidywaniami, znacząco szybszy. Różne architektury perceptronowych sieci neuronowych wymagały optymalizowania różnej liczby parametrów. Okazało się, że generalnie jakość wyników uzyskiwanych przy zastosowaniu algorytmów z rodziny Differential Evolution pogarszała się wraz ze wzrostem liczby parametrów wymagających optymalizacji szybciej, niż w przypadku innych metod. Dwie najlepsze metody optymalizacji zastosowane do optymalnej architektury sieci pozwalały na uzyskiwanie wyników dających współczynnik Nasha-Sutcliffea powyżej 0.91 dla niezależnych danych testowych. Co interesujące, najłabsze spośród przetestowanych metod optymalizacji prowadziły do wyników, dla których współczynnik ten wynosił między 0 a 0.5. Wynik ten wyraźnie wskazuje, iż tylko część metaheurystyk może być konkurencją dla algorytmów gradientowych przy uczeniu sieci neuronowych oraz, że nietrafny wybór metody może prowadzić do bardzo niepomysłnych rezultatów.

Artykuł [3] przybliżył sieci neuronowe typu Product-Units środowisku hydrologów. Wydaje się dziwne, iż ten rodzaj sieci neuronowych, chociaż zaproponowany został jeszcze w 1989 roku przez Durbina i Rumelharta (1989), nie doczekał się zainteresowania środowiska wcześniej. Zaletą sieci Product-Units jest niewątpliwie stosunkowo mała liczba parametrów wymagających optymalizacji. Pewnym problemem natomiast jest fakt, iż z uwagi na to, że wagi sieci występują w potęgach, funkcja celu ma zazwyczaj bardzo nieregularny kształt i jest trudna do optymalizacji. Aby przedstawić bardziej kompletny obraz sieci Product-Units, w pracy [3] przedyskutowano wpływ na jakość uzyskiwanych wyników zastosowania różnych algorytmów optymalizacji i metod zapobiegających przeuczeniu sieci, wyboru odpowiedniej architektury oraz ograniczeń kosztowych. Wyniki uzyskiwane za pomocą sieci Product-Units porównano z wynikami uzyskiwanymi za pomocą sieci perceptronowych i modelu konceptualnego HBV, zarówno z korektą wyników, jak i bez niej. Spośród 30-letniej serii pomiarowej, dane z 10 lat stanowiły niezależny zbiór testowy. Wykorzystano te same zmienne wejściowe do modelu, które dawały najlepsze wyniki w pracy [6], różnice w architekturze sieci dotyczyły więc tylko liczby neuronów ukrytych. Jak w artykule [6], każdy eksperyment powtórzono pięćdziesięciokrotnie, porównywano zarówno wartości średnie, jak i mediany uzyskanych wskaźników jakości.

Istotną nowością zaproponowaną w pracy [3] było wprowadzenie ograniczeń na zakres wag sieci neuronowej typu Product-Units. Pokazano że, przynajmniej dla badanego problemu, gdy zakres wag zostanie ograniczony do przedziału $[-1,1]$, optymalizacja takich sieci neuronowych może być stosunkowo łatwa. Właściwie dobierając poszczególne elementy sieci Product-Units, uzyskano prognozy przepływu o około 10% lepsze

(porównując błąd średniokwadratowy) niż w przypadku sieci neuronowych perceptronowych i modelu HBV z korektą wyników. Zastosowanie szerszych lub węższych przedziałów wag sieci Product-Units prowadzi do pogorszenia jakości prognoz.

Na 11 przetestowanych algorytmów optymalizacji składało się tym razem aż 9 wariantów algorytmu Differential Evolution, wśród nich zarówno algorytm DEGL, jak i zaproponowany przeze mnie w pracy [7] algorytm GDE, które pozwalały na uzyskiwanie najkorzystniejszych wyników odpowiednio w pracach [6] (DEGL) i [4] (GDE). Pozostałe dwa algorytmy to metoda Levenberga-Marquardta oraz EPUS-PSO. Zastosowano także wariant algorytmu DEGL z mutacją uwzględniającą odległości między osobnikami (ang. Proximity-based mutation operator), zaproponowaną przez Efitropakisa i in. (2011). Porównując poszczególne algorytmy, można wysnuć analogiczne wnioski jak w przypadku pracy [6] – najlepsze wyniki uzyskiwane są, gdy sieci neuronowe optymalizuje się za pomocą algorytmów DEGL i Levenberga-Marquardta. Wskaźniki jakości wyników uzyskiwanych za pomocą algorytmu DEGL i jego wariantu zaproponowanego przez Efitropakisa i in. (2011) są prawie identyczne. Interesujące wyniki uzyskano przeprowadzając dodatkowe testy dla wybranej architektury sieci Product-Units, gdzie dopuszczono bardzo dużą (ale wciąż realistyczną) liczbę wywołań funkcji celu w trakcie optymalizacji sieci. Okazało się wówczas, że jakość wyników uzyskiwanych z sieci neuronowych uczonych różnymi algorytmami z rodziny Differential Evolution staje się zbliżona, gdy czas obliczeń jest wystarczająco długi. Oznacza to, że większość tych algorytmów nie „utyka” przedwcześnie w optimum lokalnym. W takim razie główną zaletą algorytmu DEGL dla uczenia sieci neuronowych typu Product-Units jest względna szybkość (w stosunku do innych wariantów Differential Evolution).

Zaskakującym wynikiem pracy był fakt, że w przypadku sieci Product-Units z wagami ograniczonymi do przedziału $[-1,1]$, metody zapobiegające przeuczeniu sieci okazały się niepotrzebne, najlepsze wyniki uzyskiwano, gdy żadna z nich nie była stosowana. W przypadku sieci perceptronowych uzyskiwano odmienne rezultaty (patrz opis pracy [2]), czego innego także można się było spodziewać w oparciu o literaturę tematu. W związku z tym wynik taki albo został osiągnięty przypadkowo (tzn. jest prawdziwy jedynie dla danych ze zlewni Annapolis), albo świadczy o kolejnej wyższości sieci Product-Units z ograniczeniami na wielkość wag nad sieciami perceptronowymi. Wymaga to weryfikacji w przyszłości na danych pochodzących z innych zlewni.

W pracy zauważono także, że jakość wyników uzyskanych za pomocą sieci neuronowych perceptronowych oraz otrzymanych z modelu HBV z adaptacją wyników jest niemal identyczna. Jak wspomniano powyżej, zastosowanie sieci Product-Units z ograniczeniami wartości wag pozwala na poprawę jakości wyników nawet o 10% (porównując błąd średniokwadratowy).

Artykuł [2] poświęcony jest testowaniu przydatności metod zapobiegających przeuczeniu sieci neuronowych stosowanych do modelowania odpływu ze zlewni. Według mojej wiedzy, zagadnienie to zostało wcześniej poruszone w jednej tylko pracy hydrologicznej (Giustolisi i Laucelli, 2005), której autorzy jednak interesowali się jedynie bardzo małymi zlewniami (do 5 km^2) w zupełnie innych warunkach klimatycznych (Włochy) i nie brali pod uwagę Algorytmów Ewolucyjnych jako metod optymalizacji sieci neuronowych. W pracy [2] pokazano w jaki sposób połączenie różnych metod zapobiegających przeuczeniu sieci z którąś z dwóch najlepiej spisujących się metod optymalizacji (algorytmem Levenberga-Marquardta i DEGL) wpływa na jakość uzyskiwanych wyników.

W pracy [2] porównano trzy typy metod zapobiegających przeuczeniu sieci neuronowych: wczesnego zatrzymywania (ang. early stopping) w wersji zaproponowanej przez Prechleta (1998), metody zwanej Optimized Approximation Algorithm wprowadzonej przez Liu i in. (2008), oraz kilku wariantów dodawania szumu, bazujących na propozycjach Holmstroma i

Koistinnena (1992). Testowano różne architektury perceptronowych sieci neuronowych, by określić wpływ liczby parametrów na wyniki uzyskiwane za pomocą poszczególnych optymalizatorów i metod zapobiegających przeuczeniu. Zbiory danych, liczba powtórzeń każdego eksperymentu oraz metody oceny wyników były takie same jak w pracy [3].

W pracy [2] pokazano, że najlepsze wyniki można uzyskać za pomocą metody dodawania szumu, ale tylko pod warunkiem właściwego doboru szeregu jej parametrów, co jest zadaniem niełatwym i czasochłonnym. Poza tym odchylenie standardowe wartości wskaźników jakości wyników uzyskiwanych za pomocą tej metody bywają duże. Wyniki uzyskiwane z użyciem metody wczesnego zatrzymywania są jedynie nieznacznie słabsze, metoda jest jednak dużo szybsza i prostsza w użyciu. W związku z tym wybór metody zapobiegającej przeuczeniu powinien zależeć od tego, jak dużo czasu można na obliczenia poświęcić. Optimized Approximation Algorithm, mimo że jest historycznie najnowszą z przetestowanych metod, nie sprawdzał się, przynajmniej w zastosowaniu do wybranego zagadnienia [2]. Dwie prawdopodobne tego przyczyny mają charakter bardzo techniczny i nie będą tu dyskutowane, są one opisane o pracy [2].

Porównanie wyników uzyskiwanych za pomocą algorytmów DEGL i Levenberga-Marquardta prowadziło do podobnych wniosków jak wysnuwane w pracy [6]. Prostsze perceptronowe sieci neuronowe (mające mniej parametrów) pozwalały na uzyskiwanie lepszych wyników niż bardziej skomplikowane, jeśli do treningu używano algorytmu DEGL. Sieci mające więcej zmiennych wejściowych (a więc korzystające z dodatkowej informacji) i większą liczbę neuronów ukrytych mogą być użyteczne, jeśli stosuje się algorytm Levenberga-Marquardta. Ogólnie, bardziej skomplikowane sieci perceptronowe trenowane algorytmem Levenberga-Marquardta prowadzą do lepszych wyników niż prostsze, trenowane za pomocą algorytmu DEGL. Porównując natomiast wyniki uzyskane dla przez sieci neuronowe o prostszych architekturach, można wskazać na minimalną przewagę algorytmu DEGL.

Opis prac [1,5,7]

Poza wykorzystaniem metod optymalizacji do rozwiązywania problemów hydrologicznych w swojej pracy podejmowałem także próby zaproponowania własnych Algorytmów Ewolucyjnych. W trzech kolejnych publikacjach zaproponowałem trzy algorytmy z rodziny Differential Evolution. Rodzina ta jest obecnie bardzo popularna w różnych dziedzinach nauki i dynamicznie się rozwija (Das i Suganthan, 2011). Algorytmy zaproponowane w pierwszych dwóch pracach należą do grupy tak zwanych rozłożonych algorytmów Differential Evolution (ang. Distributed Differential Evolution), w ramach których populacja osobników przeszukujących przestrzeń poszukiwań jest podzielona na subpopulacje (grupy), które zazwyczaj działają niezależnie od siebie, ale w pewnych sytuacjach wymieniają się informacjami lub osobnikami. Określenie zasad wymiany informacji lub osobników jest zwykle najtrudniejsze przy tworzeniu takich algorytmów. Trzeci algorytm, zaproponowany w mojej ostatniej pracy [1] należy do adaptacyjnych i memetycznych metod Differential Evolution.

Mój pierwszy algorytm optymalizacji, zwany Grouped Multi-Strategy Differential Evolution (GDE) został zaproponowany w pracy [7]. Główną myślą przewodnią algorytmu jest wykorzystanie informacji dotyczącej relacji między lokalizacją w przestrzeni poszukiwań znanych na danym etapie obliczeniowym optimum lokalnych w celu znalezienia optimum globalnego, wykorzystując ideę algorytmów rozłożonych. Populacja osobników została w algorytmie GDE podzielona na cztery grupy (subpopulacje), trzy spośród nich mogą wymieniać informacje jedynie w rzadkich, ściśle określonych sytuacjach, natomiast czwarta ma stale pełen dostęp do informacji zgromadzonej przez wszystkie osobniki. Osobniki wykorzystują trzy różne strategie mutacji i krzyżowania, co czyni algorytm

bardziej elastycznym w stosunku do klasycznych wariantów Differential Evolution. Algorytm GDE został porównany z dwoma innymi, stosunkowo prostymi algorytmami Differential Evolution na zbiorze 13 funkcji testowych (10- do 100-wymiarowych) o różnym zakresie trudności. Wyniki były zachęcające, wyższość zaproponowanego algorytmu była szczególnie wyraźna przy rozwiązywaniu trudniejszych problemów 50- i 100-wymiarowych. Ponieważ algorytmy rozłożone są silnie wrażliwe na wielkość populacji, gdyż zarówno korzystanie ze zbyt małych, jak i zbyt licznych grup, może negatywnie wpłynąć na ich zbieżność do optimum, przetestowany został wpływ wielkości populacji na wyniki uzyskiwane przez GDE. Zaproponowany algorytm został w późniejszych pracach zastosowany do uczenia sieci neuronowych do celów oszacowywania wartości współczynników dyspersji podłużnej w rzekach (w pracach [4] oraz Piotrowski i in., 2010) oraz modelowania odpływu ze zlewni [6,3]. W ciągu dalszych prac doszedłem jednak do wniosku, że algorytm GDE można znacząco ulepszyć. Zdałem sobie również sprawę z pewnych braków, które wystąpiły w artykule [7]. Po pierwsze, porównanie nowej metody z zaledwie dwiema starszymi to zdecydowanie za mało. Po drugie, dobór 13 funkcji testowych nie był najszcześniejszy, gdyż brakowało wśród nich tak zwanych problemów obracanych (ang. rotated problems). Tymczasem od czasu opublikowania pracy Salomona (1996) wiadomo, że Algorytmy Ewolucyjne powinny być testowane na problemach obracanych. Znaczna część metaheurystyk osiąga bardzo dobre wyniki w przypadku, gdy optima lokalne są generalnie zlokalizowane równolegle do układu osi współrzędnych w przestrzeni poszukiwań, natomiast w innych wypadkach spisuje się dużo słabiej. Jest to najczęściej efekt specyficznych cech stosowanych operacji krzyżowania i dotyczy jak najbardziej także algorytmów z rodziny Differential Evolution. Wszystko to motywowało mnie do prac nad modyfikacją algorytmu GDE i bardziej dogłębnym porównaniem udoskonalonej jego wersji z innymi metodami. Nowo powstały algorytm nazwany został Differential Evolution algorithm with Separated Groups (DE-SG) [5].

Algorytm DE-SG, zaproponowany w pracy [5], był rozwinięciem algorytmu GDE, ale czerpał także idee z innych rozłożonych lub samoadaptacyjnych algorytmów Differential Evolution oraz tak zwanych modeli wyspowych (ang. Island Models) (Tanese, 1989). Niemniej zarówno struktura algorytmu DE-SG, jak i metody wymiany informacji i osobników między subpopulacjami (grupami), są w znacznej mierze unikatowe. W przeciwieństwie do większości rozłożonych algorytmów Differential Evolution, populacja DE-SG jest podzielona na połowy i zasady migracji osobników są odmienne dla każdej z nich. Każda połowa składa się z szeregu 10-osobnikowych grup, które generalnie pracują osobno. Mała grupa pozwala na szybką wymianę informacji, w związku z tym zazwyczaj jest zdolna do względnie szybkiego zlokalizowania optimum lokalnego. Natomiast eksplorację ułatwiają reguły wymiany informacji i osobników między grupami. Szczególnie istotna jest tu nowa idea, by reguły migracji osobników w zasadniczej części różniły się dla dwóch połów populacji. Grupy w ramach połówek populacji są uszeregowane, grupa z jednego końca przyciąga do siebie najlepsze osobniki, grupa z drugiego – najgorsze. W jednej połowie populacji najlepsze osobniki migrują szybko do najlepszej grupy, natomiast najgorsze – powoli. W drugiej połowie tempo migracji jest odwrotne. Daje to algorytmowi dodatkową elastyczność przy rozwiązywaniu różnego typu problemów. Korzystając z idei zaproponowanej przez Qin i in. (2009) dla algorytmu SADE, w DE-SG stosowane są dwie różne strategie mutacji i krzyżowania. Jedna ma przede wszystkim zalety przy eksploracji, druga – eksploatacji. DE-SG został porównany z 8 różnymi Algorytmami Ewolucyjnymi, w tym także kilkoma najnowszymi wariantami Differential Evolution opublikowanymi w najbardziej prestiżowych czasopismach z dziedziny. Porównanie, przeprowadzone na 19 obracanych problemach testowych 10- do 50-wymiarowych, pozwoliło na wskazanie DE-SG jako najlepszej spośród testowanych metod. Pewną wadą pracy natomiast była założona

przeze mnie duża (wyższa niż zwykle przyjmowane w literaturze), niemniej oczywiście taka sama dla wszystkich metod, liczba dopuszczalnych wywołań funkcji celu.

Koncepcja trzeciego z zaproponowanych przeze mnie algorytmów [1] była odmienna. Mimo że istnieje już wiele algorytmów Differential Evolution, prosta idea łączenia najistotniejszych cech z najbardziej efektywnych wariantów w nową całość, tak, by powstał algorytm jeszcze „lepszy”, jest ciągle rzadko wykorzystywana w literaturze. Nowy algorytm zaproponowany w pracy [1] łączy najistotniejsze cechy trzech bardzo popularnych i skutecznych wariantów Differential Evolution. Po pierwsze, wprowadzono adaptację wartości parametrów algorytmu oraz prawdopodobieństw użycia każdej spośród 5 strategii mutacji w wersji podobnej do zaproponowanej w SADE (Qin i in., 2009). Po drugie, w pewnych sytuacjach zaproponowany algorytm korzysta z metod przeszukiwania o bardziej lokalnym charakterze, wykorzystując algorytm Neldera-Meada (1965) – idea takiego podejścia memetycznego pochodzi z pracy Caponio i in. (2009). Po trzecie, mutacja jest podzielona na dwie fazy – lokalną i globalną, przy czym mutacja lokalna bazuje na koncepcji sąsiedztwa osobników (topologia pierścienia) – ideę tę wprowadzono do algorytmów Differential Evolution przy konstrukcji algorytmu DEGL w pracy Das i in. (2009). Nowy algorytm nazwano Adaptive Memetic Differential Evolution with Global and Local neighborhood-based mutation operators (AM-DEGL). AM-DEGL został porównany z 13 wariantami Differential Evolution, w tym zarówno najnowszymi, jak i moim poprzednim algorytmem (DE-SG) na zbiorze 25 funkcji testowych (10-, 30- i 50-wymiarowych), zawierających problemy obracane, przesunięte i hybrydowe. Praca pokazała duże możliwości wzmiankowanej wyżej idei łączenia najskuteczniejszych cech istniejących już algorytmów. Chociaż bowiem żaden z testowanych algorytmów nie uzyskał najlepszego wyniku dla większości spośród 25 problemów, uśrednione wyniki uzyskiwane za pomocą AM-DEGL są statystycznie istotnie lepsze (na poziomie istotności 5%) od wyników uzyskiwanych za pomocą algorytmów konkurencyjnych dla problemów wielowymiarowych. W pracy przetestowano także znaczenie szeregu komponentów algorytmu AM-DEGL – okazało się, że często usunięcie lub modyfikacja tylko jednego komponentu ma mały wpływ na jakość otrzymywanych wyników.

W większości prac poświęconych wprowadzeniu nowego Algorytmu Ewolucyjnego autorzy ograniczają się do pokazania, iż w jakimś sensie ich nowa metoda jest „lepsza” od innych, wybranych w mniej lub bardziej obiektywny sposób. W pracy [1] przedyskutowano szczegółowo inny punkt widzenia. Dawno już postawiono pytanie, czy proponowanie nowych metaheurystyk ma sens, jeżeli, mówiąc w dużym uproszczeniu, udowodniono (No Free Lunch Theorems; Wolpert i Macready, 1997), że wartość oczekiwana jakości wyników wszystkich możliwych metaheurystyk jest identyczna, gdy bierze się pod uwagę wszystkie możliwe problemy. Oznacza to między innymi, że żadna z metaheurystyk nie może być lepsza od przeszukiwania losowego. W związku z tym w ostatnim rozdziale pracy [1] przedyskutowano ograniczenia i implikacje No Free Lunch Theorems w oparciu o bogatą, acz niestety rzadko cytowaną literaturę. Znacznie upraszczając, bardzo ważnym elementem w dyskusji jest określenie „wszystkie problemy” – gdyż wśród nich jedynie nieliczne mogą mieć jakiegokolwiek znaczenie dla kogokolwiek, a ewentualna ilustracja graficzna zdecydowanej większości z nich ukazałaby oglądającemu jedynie obraz rozmazany i chaotyczny. Z tego i innych, nie dyskutowanych w tym referacie powodów, część autorów pomija ten temat w swoich rozważaniach. Niemniej zastosowanie Algorytmów Ewolucyjnych może prowadzić do zupełnej porażki nawet dla problemów, które mogą jak najbardziej być dla użytkownika istotne i nie wyglądają skomplikowanie. Jako przykład w pracy [1] zaproponowano ciągłą i różniczkowalną funkcję, której minimum jest poszukiwane w 2-wymiarowej przestrzeni i uwzględnione są ograniczenia kostkowe. W pracy [1] pokazano, że uśredniona jakość wyników uzyskiwanych za pomocą każdego spośród 14

rozważanych wariantów Differential Evolution jest gorsza od uśrednionej jakości wyników otrzymywanych przez proste przeszukiwanie losowe. Pokazano także, że w przypadku, gdy każdy spośród 14 algorytmów jest uruchomiony w celu znalezienia maksimum wzmiankowanej funkcji, w trakcie poszukiwań znajduje on niższą wartość funkcji celu niż w trakcie poszukiwania minimum – wynik ten, mimo że na pierwszy rzut oka zaskakujący, jest również zgodny z wnioskami wyciąganymi z No Free Lunch Theorems. Dyskusję powyższą przeprowadzono w pracy [1] w celu 1. krótkiego podsumowania dotychczasowych prac dotyczących No Free Lunch Theorems; 2. stymulowania debaty nad sposobem wprowadzania i porównywania, tak licznych przecież, nowych Algorytmów Ewolucyjnych.

Podsumowanie:

Jednym z głównych problemów hydrologii jest dostępność i jakość danych. Ogólnie biorąc, gdy ilość dostępnych danych jest silnie ograniczona, jak w przypadku oszacowywania wartości współczynników dyspersji podłużnej w rzekach, jakość wyników uzyskiwanych za pomocą modeli optymalizowanych różnymi metodami, w tym Algorytmami Ewolucyjnymi, jest umiarkowanie zróżnicowana (co absolutnie nie znaczy, że zbliżona). Gdy liczba danych jest duża, jak w przypadku modelowania odpływu ze zlewni rzeki Annapolis, wpływ optymalizatora na jakość uzyskiwanych z modelu wyników jest bardzo duży. Co może dziwić, niezależnie od tego, że w bardzo wielu pracach stosowane są różnego typu metaheurystyki do uczenia sieci neuronowych, oraz, mimo iż algorytmy gradientowe nie mają zdolności eksploracyjnych, większość popularnych dziś wariantów Algorytmów Ewolucyjnych, łącznie z większością najnowszych ich wersji, w świetle przeprowadzonych analiz nie powinno być stosowanych do uczenia sieci neuronowych. Jedynie kilka metod pozwala na osiągnięcie podobnych lub marginalnie lepszych wyników niż uzyskiwane za pomocą najskuteczniejszych algorytmów gradientowych. Co interesujące, te warianty Algorytmów Ewolucyjnych, które okazują się najskuteczniejsze w przypadku uczenia sieci neuronowych (DEGL, EPUS-PSO, w niektórych wypadkach zaproponowany przeze mnie GDE) nie wypadają najlepiej w porównaniach przeprowadzanych na podstawie klasycznych funkcji testowych.

Bibliografia:

- Abrahart, R.J., Anctil, F., Coulibaly, P., Dawson, C.W., Mount, N.J., See, L.M., Shamseldin, A.Y., Solomatine, D.P., Toth, E., Wilby, R.L., 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. *Progress in Physical Geography* 36(4), 480-513.
- Bäck, T., Schwefel, H. P., 1993. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation* 1(1), 1-23.
- Bencala, K.E., Walters, R.A., 1983. Simulation of solute transport in a mountain pool-and-riffle stream: a transient storage model. *Water Resources Research* 19(3), 718-724.
- Branke, J., 1995. Evolutionary algorithms for neural network design and training. Technical Report No. 322, Institute AIFB, University of Karlsruhe.
- Cantu-Paz, E., Kamath, C., 2005. An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics* 35(5), 915-927.
- Caponio, A., Neri, F., Tirronen, V., 2009. Super-fit control adaptation in memetic differential evolution frameworks. *Soft Computing* 13(8-9), 811-831.
- Caraffini, F., Neri, F., Iacca, G., Mol, A., 2013. Parallel Memetic Structures. *Information Sciences* 227(1), 60-82.
- Cheong, T.S., Younis, B.A., Seo, I.W., 2007. Estimation of key parameters in model for solute transport in rivers and streams. *Water Resources Management* 21(7), 1165-1186.

- Chu, S., Tsai, P., Pan, J., 2006. Cat swarm optimization. *Lecture Notes in Computer Science* 4099, 854–858.
- Civicioglu, P., 2013. Artificial cooperative search algorithm for numerical optimization problems. *Information Sciences* 229, 58-76.
- Crepinsek, M., Liu, S.H., Mernik, L., 2012. A note on teaching-learning-based optimization algorithm. *Information Sciences* 212, 79-93.
- Das, S., Abraham, A., Chakraborty, U.K., Konar, A., 2009. Differential Evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation* 13(3), 526-553.
- Das, S., Suganthan, P.N., 2011. Differential Evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* 15(1), 27-54.
- Deng, Z.Q., Bengtsson, L., Singh, V.P., 2002. Longitudinal dispersion coefficient in single-channel streams. *Journal of Hydraulic Engineering ASCE* 128(10), 901–916.
- Ding, S., Li, H., Su, C., Yu, J., Jin, F., 2013. Evolutionary artificial neural networks: a review. *Artificial Intelligence Review* 39, 251-260.
- Dorigo, M., Maniezzo, V., Coloni, A., 1996. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man and Cybernetics – Part B: Cybernetics* 26(1), 29-41.
- Durbin, R., Rumelhart, D.E., 1989. Product Units: a computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation* 1, 133–42.
- Eberhart, R.C., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39-43.
- Epitropakis, M.G., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N., 2011. Enhancing differential evolution utilizing proximity-based mutation operations. *IEEE Transactions on Evolutionary Computation* 15(1), 99-119.
- Fogel, D.B., 2000. What is Evolutionary Computation? *IEEE Spectrum* 37(2), 26-32.
- Geem, Z.W., Kim, J.H., Loganathan, G.V., 2001. A new heuristic optimization algorithm: Harmony search. *Simulation* 76(2), 60-68.
- Giustolisi, O., Laucelli, D., 2005. Improving generalization of artificial neural networks in rainfall-runoff modelling. *Hydrological Sciences Journal* 50(3), 439-457.
- Hagan, M.T., Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks* 5(6), 989-993.
- Hansen, N., Ostermeier, A., 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proc. IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, pp. 312–317.
- He, S., Wu, Q.H., Saunders, J.R., 2009. Group search optimizer: an optimization algorithm inspired by animal search behaviour. *IEEE Transactions on Evolutionary Computation* 13(5), 973-990.
- Holland, I.H., 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor.
- Holmstrom, L., Koistinen, P., 1992. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks* 3, 24-38.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feed forward networks are universal approximators. *Neural Networks* 2, 359–366.
- Hsieh, S.T., Sun, T.Y., Liu, C.C., Tsai, S.J., 2009. Efficient population utilization strategy for Particle Swarm Optimizer. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 39(2), 444-456.
- Hsu, K.L., Gupta, H.V., Sorooshian, S., 1995. Artificial neural network modeling on the rainfall–runoff process. *Water Research and Resources* 31(10), 2517–2530.

- Hunter, D., Yu, H., Pukish, M.S., Kolbusz, J., Wilamowski, B.M., 2012. IEEE Transactions on Industrial Informatics 8(2), 228-240.
- Iacca, G., Neri, F., Mininno, E., Ong, Y.S., Lim, M.H., 2012. Ockham's razor in memetic computing: three stage optimal memetic exploration. Information Sciences 188, 17-43.
- Islam, M.M., Sattar, M.A., Amin, M.F., Yao, X., Murase, K., 2009. A new constructive algorithm for architectural and functional adaptation of artificial neural networks. IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 39(6), 1590-1605.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671-680.
- Kolda, T.G., Lewis, R.M., Torczon, V., 2003. Optimization by direct search: new perspectives on some classical and modern methods. SIAM Review 45(3), 385-482.
- Koza, J.R., 1992. Genetic Programming: On the programming of computers by means of natural selection. MIT Press.
- Lam, A.Y.S., Li, V.O.K., 2010. Chemical reaction inspired metaheuristic for optimization. IEEE Transactions on Evolutionary Computation 14(3), 381-399.
- Liu, Y., Starzyk, J.A., Zhu, Z., 2008. Optimized approximation algorithm in neural networks without overfitting. IEEE Transactions on Neural Networks 19(6), 983-995.
- Masegosa, A.D., Pelta, D.A., Verdegay, J.L., 2013. A centralised cooperative strategy for continuous optimisation: The influence of cooperation in performance and behaviour. Information Sciences 219, 73-92.
- Nelder, A., Mead, R., 1965. A simplex-method for function minimization. Computer Journal 7(4), 308-313.
- Onwobulu, G.C., Babu, B.V., 2004. New optimization techniques in engineering. Studies in Fuzziness and Soft Computing, Vol. 141, Springer-Verlag.
- Piotrowski, A.P., Rowinski, P.M., Napiorkowski, J.J. (2010) Uncertainty study of data-based models of pollutant transport in rivers. Proceedings of River Flow 2010 Conference, Braunschweig, Germany, 8-10 September.
- Prechlet, L., 1998. Automatic early stopping using cross-validation: quantifying the criteria. Neural Networks 11(4), 761-777.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 2006. Numerical recipes in Fortran 77. The Art of Scientific Computing. Cambridge University Press, Cambridge, UK.
- Price, W.L., 1977. A Controlled Random Search procedure for global optimisation. Computer Journal 20(4), 367-370.
- Qin, A.K., Huang, V.L., Suganthan, P.N., 2009. Differential Evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation 13(2), 398-417.
- Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: A gravitational search algorithm. Information Sciences 179, 2232-2248.
- Rosenbrock, H.H., 1960. An automated method for finding the greatest or least value of a function. Computer Journal 3(3), 175-184.
- Rowinski, P.M., Piotrowski, A., Napiorkowski, J.J., 2005. Are artificial neural networks techniques relevant for the estimates of longitudinal dispersion coefficient in rivers? Hydrological Sciences Journal 50(1), 175-187.
- Salomon, R., 1996. Re-evaluating genetic algorithm performance under coordinate rotation on benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. BioSystems 39, 263-278.
- Simon, D., 2008. Biogeography-based optimization. IEEE Transactions on Evolutionary Computation 12(6), 702-713.

- Stanley, K.O., Miikkulainen, R., 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2), 99-127.
- Storn, R., Price, K.V., 1995. Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. Report TR-95-012, International Computer Sciences Institute, Berkeley, California, USA.
- Tanese, R., 1989. Distributed genetic algorithms. *Proceedings of 3rd International Conference on Genetic Algorithms*, San Francisco, CA, USA, pp. 434–439.
- Taylor, G.I., 1953. Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society of London, Series A*, 219, 186–203.
- Wallis, S.G., Manson, J. R., 2004. Methods for predicting dispersion coefficients in rivers. *Water Management*, 157(WM3), 131–141.
- Weise, T., Chiong, R., Tang, K., 2012. Evolutionary optimization: Pitfalls and booby traps. *Journal of Computer Science and Technology* 27(5), 907-936.
- Whitley, D., Starkweather, T., Bogart, C., 1990. Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel Computing* 14(3), 347-361.
- Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67–82.
- Yang, X.S., Deb, S., 2009. Cuckoo Search via Levy Flights. *2009 World Congress on Nature & Biologically Inspired Computing*, pp. 210-214.
- Yao, X., 1993. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems* 8(4), 539-567.
- Yao X., Liu, Y., 1997. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on neural networks* 8(3), 694-713.
- Yao, X., 1999. Evolving artificial neural networks. *Proceedings of the IEEE* 87(9), 1423-1447.

5. Omówienie pozostałych osiągnięć naukowo-badawczych (artystycznych)

Transport zanieczyszczeń w rzekach

Brałem udział w eksperymencie znacznikowym na rzece Narwi przeprowadzonym w ramach grantu „Procesy transportowe w korytach rzecznych” (kierownik: Paweł Rowiński). Wziąłem udział w pracach polowych oraz przy określaniu współczynników dyspersji podłużnej dla poszczególnych odcinków rzeki, zarówno za pomocą szeregu modeli empirycznych, jak i na podstawie danych pomiarowych.

Opublikowałem kilka prac poświęconych zagadnieniu modelowania transportu zanieczyszczeń w rzekach na podstawie różnego typu danych, zarówno pochodzących od współautorów, jak i zaczerpniętych z literatury. Pomijając dwie prace, które dotyczyły zastosowania Algorytmów Ewolucyjnych do tego celu i były omówione w części opisującej główne osiągnięcie naukowe, wyniki zostały opisane w szeregu publikacji konferencyjnych oraz w trzech artykułach, które ukazały się w czasopiśmie z bazy JCR (w tym jedna praca ukazała się przed obroną doktoratu).

Opublikowane artykuły:

Piotrowski, A.P., Napiorkowski, J.J., Rowinski, P.M., Wallis, S.G., 2011. Evaluation of temporal concentration profiles for ungauged rivers following pollution incidents. *Hydrological Sciences Journal* 56(5), 883-894.

Piotrowski, A., Wallis, S.G., Napiorkowski, J.J., Rowinski, P.M., 2007. Evaluation of 1-D tracer concentration profile in a small river by means of multi-layer perceptron neural networks. *Hydrology and Earth System Sciences* 11, 1883-1896.

Rowinski, P.M., Piotrowski, A., Napiorkowski, J.J., 2005. Are artificial neural networks techniques relevant for the estimates of longitudinal dispersion coefficient in rivers? *Hydrological Sciences Journal* 50(1), 175-187.

Napiorkowski J.J., Piotrowski A., Rowinski P.M., Wallis S.G., 2012. Product Unit neural networks for estimations of longitudinal dispersion coefficients in rivers. 2nd IAHR Europe Congress, 27-29 June, Germany, Munich.

Piotrowski, A.P., Rowinski, P.M., Napiorkowski, J.J., 2010. Uncertainty study of data-based models of pollutant transport in rivers. *Proceedings of River Flow 2010 Conference*, Braunschweig, Germany, 8-10 September.

Piotrowski, A.P., Rowinski, P.M., Napiorkowski, J.J., 2009. Estimation of parameters of models of pollutant transport in rivers depending on data availability. 33rd IAHR Congress: *Water Engineering for a Sustainable Environment*, Vancouver, pp. 1179-1186.

Napiorkowski, J.J., Piotrowski, A., Rowinski, P.M., Wallis, S.G., 2008. Prediction of the fate of pollutants in rivers by means of nonlinear Volterra series. *River Flow 2008: Proceedings of the International Conference on Fluvial Hydraulics*, Çeşme-İzmir, Turkey, 3-5 September, 2469-2476.

Wallis, S.G., Piotrowski, A., Rowinski, P.M., Napiorkowski, J.J., 2007. Prediction of dispersion coefficients in a small stream using artificial neural networks. *Proceedings of 32nd IAHR Congress*, Venice.

Rowinski, P.M., Guymer, I., Bielonko, A., Napiorkowski, J.J., Pearson, J., Piotrowski, A., 2007. Large scale tracer study of mixing in a natural lowland river. *Proceedings of 32nd IAHR Congress*, Venice.

Piotrowski, A., Rowinski, P.M., Napiorkowski, J.J., 2006. Assessment of longitudinal dispersion coefficient by means of different neural networks. *Proceedings of the 7th International Conference on Hydroinformatics 2006*, Nice, France.

Piotrowski, A., Napiorkowski, J.J., 2005. Dispersion coefficient assessment by means of different neural networks. *Materiały VIII Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa.

Prognozowanie przepływów rzecznych

Szereg moich pierwszych prac naukowych poświęcony był prognozowaniu przepływu rzeczno. Wykorzystywane do tego celu były różnego typu modele bazujące na danych: regresja liniowa, różne typy sieci neuronowych, modele rekonstrukcji przestrzeni fazowej, podejście najbliższych sąsiadów oraz szeregi Voltery. Prace dotyczyły zarówno prognoz autoregresyjnych, jak i odpływu ze zlewni. Dane pozyskane były z kilku obszarów świata: z Polski (Nysa Kłodzka), USA (Illinois) oraz Kanady (kilka lokalizacji w zachodniej części kraju). W późniejszym okresie zainteresowałem się wykorzystaniem Algorytmów Ewolucyjnych do kalibracji niektórych modeli służących do modelowania odpływu ze zlewni – czego pokłosiem są 3 prace zaliczone do głównego osiągnięcia naukowego.

Opublikowane artykuły:

Piotrowski, A., Napiorkowski, J.J., Rowinski, P.M., 2006. Flash-flood forecasting by means of neural networks and nearest neighbour approach – a comparative study. *Nonlinear Processes in Geophysics* 13, 443-448.

Napiorkowski, J.J., Piotrowski, A., 2005. Artificial neural networks as an alternative to the Volterra series in rainfall-runoff modeling. *Acta Geophysica Polonica*, 53(4), 459-472.

Piotrowski, A., Rowinski, P.M., Napiorkowski, J.J., 2004. River flow forecast by selected black box models. *River Flow 2004*, Ed: M. Greco, A. Carravetta, R. D. Morte, Leiden, Netherlands.

Piotrowski, A., 2003. Porównanie prognoz przepływów rzecznych otrzymanych z modeli przestrzeni fazowej i sieci neuronowych. *Współczesne Problemy Hydrauliki Wód Śródlądowych, Materiały XXIII Ogólnopolskiej Szkoły Hydrauliki*, Gdańsk.

Poszerzony abstrakt:

Piotrowski, A., Napiorkowski, J.J., Rowinski, P.M., 2004. Extended phase-space reconstruction technique for the prediction of river flows. *Geophysical Research Abstracts*, Vol. 6, 07446, 2004;

Sterowanie zbiornikiem wodnym

Byłem zaangażowany w prace dotyczące sterowania zbiornikiem Siemianówka (położonego w Polsce północno-wschodniej) w celu poprawy warunków wodnych Narwiańskiego Parku Narodowego. Głównym moim zadaniem była pomoc w zakresie wyboru właściwej metody optymalizacji do postawionego zadania.

Opublikowany artykuł:

Kiczko, A., Piotrowski, A., Napiorkowski, J.J., Romanowicz, R.J., 2008. Combined reservoir management and flow routing modelling: Upper Narew case study. *River Flow 2008: Proceedings of the International Conference on Fluvial Hydraulics*, Çeşme-İzmir, Turkey, 3-5 September, 1921-1928.

Geofizyczne zagrożenia obiektów jądrowych

W latach 2011-2012 brałem udział w grantie dla młodych naukowców finansowanym przez Instytut Geofizyki PAN, mającym na celu podniesienie wiedzy i kwalifikacji młodych pracowników Instytutu w zakresie oceny zagrożeń geofizycznych obiektów jądrowych.

Granty i nagrody

Obecnie kieruję dwoma grantami. Grant dla młodych naukowców finansowany jest przez Instytut Geofizyki PAN i dotyczy kwestii wyboru najodpowiedniejszych metod optymalizacji modeli odpływu ze zlewni położonych na obszarach o klimacie umiarkowanym. Drugi grant, finansowany przez Ministerstwo Nauki i Szkolnictwa Wyższego, dotyczy prognozowania temperatury wody w rzekach za pomocą modeli empirycznych.

Poza omówionymi wcześniej grantami byłem także głównym wykonawcą grantu promotorskiego finansowanego przez Ministerstwo Nauki i Informatyzacji.

W 2008 roku uzyskałem Stypendium Fundacji na Rzecz Nauki Polskiej „Start” dla młodych naukowców, a także, w roku 2009, jego przedłużenie. W 2010 roku zostałem nagrodzony Stypendium im. Profesora Kacpra Rybickiego przyznawanym młodym naukowcom w Instytucie Geofizyki PAN.