

Załącznik 2a: Autoreferat

1 Imię i Nazwisko

Wojciech Jaśkowski

2 Posiadane Stopnie i Tytuły Naukowe

1. **Stopień doktora nauk technicznych** - Politechnika Poznańska, Wydział Informatyki; dyscyplina - informatyka, 27 września 2011 r., “Algorithms for Test-Based Problems”, dr hab. inż. Krzysztof Krawiec (promotor), obroniona z wyróżnieniem.
 - wyróżniona w konkursie na najlepszą pracę doktorską Polskiego Stowarzyszenia Sztucznej Inteligencji. (PSSI), 2012.
2. **Tytuł zawodowy magistra** - Politechnika Poznańska, Wydział Informatyki; kierunek - informatyka; specjalizacja - inteligentne systemy wspomaganie decyzji, 26 lipca 2006 r., “Genetic Programming with Cross-task, Knowledge Sharing for Learning of Visual Concepts”, dr hab. inż. Krzysztof Krawiec (promotor)
 - medal “summa cum laude” (przyznano 3 medale na 4000 absolwentów).
3. **Tytuł zawodowy inżyniera** - Politechnika Poznańska, Wydział Informatyki i Zarządzania; kierunek - informatyka, 2004 r., “Lifetch - Life saving system”, dr inż. Jan Kniat (promotor)

3 Dotychczasowe zatrudnienie w jednostkach naukowych

1. Adiunkt
 - Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska,
 - od 10.2011.
2. Asystent Naukowy
 - Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska,
 - 02.2009–09.2012.
3. Asystent

- Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska,
- 10.2010–08.2011.

4. Referent Techniczny

- Instytut Informatyki, Wydział Elektryczny, Politechnika Poznańska;
- 04.2004–01.2006.

5. Wizyty naukowe

(a) NNAISENSE, Lugano, Szwajcaria,

- staż podoktorski w prywatnej jednostce badawczej,
- 16 miesięcy (05.2017–08.2018).

(b) IDSIA (Dalle Molle Institute for Artificial Intelligence), Manno, Szwajcaria,

- staż podoktorski w ramach programu MNiSW “Mobilność Plus”,
- 8 miesięcy (09.2016 - 04.2017).

(c) Gas Powered Games, Seattle, USA

- staż doktorancki, 2 tygodnie, 2007.

4 Wskazanie osiągnięcia naukowego wynikającego z ustawy o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki.

4.1 Tytuł osiągnięcia naukowego

Algorytmy i Metody Ucznienia się Strategii Gier.

4.2 Lista prac wchodzących w skład osiągnięcia naukowego

- [A1] Jaśkowski, W. Mastering 2048 with delayed temporal coherence learning, multi-stage weight promotion, redundant encoding and carousel shaping. *IEEE Transactions on Computational Intelligence and AI in Games* (in print, 2017). **[IF 2016: 1.113]**
- [A2] Liskowski, P., Jaśkowski, W. & Krawiec, K. Learning to play othello with deep neural networks. *IEEE Transactions on Computational Intelligence and AI in Games* (in print, 2017). **[IF 2016: 1.113]**
- [A3] Jaśkowski, W. & Szubert, M. Coevolutionary CMA-ES for knowledge-free learning of game position evaluation. *IEEE Transactions on Computational Intelligence and AI in Games* **8**, 389–401 (2016). **[IF 2016: 1.113]**

- [A4] Jaśkowski, W., Liskowski, P., Szubert, M. & Krawiec, K. Performance profile: a multi-criteria performance evaluation method for test-based problems. *International Journal of Applied Mathematics and Computer Science* **26**, 215–229 (2016). [IF 2016: 1.420]
- [A5] Jaśkowski, W., Szubert, M., Liskowski, P. & Krawiec, K. High-dimensional function approximation for knowledge-free reinforcement learning: a case study in SZ-Tetris. In *GECCO'15: Proceedings of the 17th annual conference on Genetic and Evolutionary Computation*, 567–574. ACM (ACM Press, Madrid, Spain, 2015).
- [A6] Jaśkowski, W. Systematic n-tuple networks for othello position evaluation. *ICGA Journal* **37**, 85–96 (2014). [IF 2014: 0.293]
- [A7] Szubert, M., Jaśkowski, W. & Krawiec, K. On scalability, generalization, and hybridization of coevolutionary learning: a case study for Othello. *IEEE Transactions on Computational Intelligence and AI in Games* **5**, 214–226 (2013). [IF 2013: 1.167]
- [A8] Jaśkowski, W., Liskowski, P., Szubert, M. & Krawiec, K. Improving coevolution by random sampling. In Blum, C. (ed.) *GECCO'13: Proceedings of the 15th annual conference on Genetic and Evolutionary Computation*, 1141–1148 (ACM, Amsterdam, The Netherlands, 2013).
- [A9] Jaśkowski, W., Szubert, M. & Liskowski, P. Multi-criteria comparison of coevolution and temporal difference learning on othello. In Esparcia-Alcazar, A. I. & Mora, A. M. (eds.) *EvoApplications 2014*, vol. 8602 of *Lecture Notes in Computer Science*, 301–312 (Springer, 2014).

4.3 Omówienie celu naukowego ww. prac oraz osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania

4.3.1 Cel Badań

Głównym celem badań było zaprojektowanie nowych, efektywnych algorytmów uczących się strategii dla gier oraz wielokryterialnych metod pozwalających analizę algorytmów uczących oraz strategii, które są efektem ich działania.

4.3.2 Kontekst Badań

“Gry są Formułą 1 sztucznej inteligencji” [1]. Abstrakcyjny charakter i precyzyjne reguły czynią gry atrakcyjnym polem testowym dla badań nad sztuczną inteligencją [2, p. 161]. Centralnym problemem dla każdej gry jest spośród wszystkich strategii danej gry, znalezienie takiej która maksymalizuje pewną miarę jakości (np. częstość wygranych, oczekiwaną liczbę punktów). Ręczne projektowanie strategii jest żmudne i nie gwarantuje sukcesu, dlatego dąży się do rozwiązywania tego problemu za pomocą metod uczenia maszynowego. W sytuacji idealnej, metody te nie powinny wymagać wkładu wiedzy człowieka (ang. *knowledge-free methods* [3]), poza opisem reguł gry. Wynik gry oraz jej przebieg są jedyną informacją uczącą. Uczenie strategii odbywa się na podstawie wyników wielu, często setek tysięcy rozgrywek.

Na problem uczenia się strategii gry można spojrzeć z wielu różnych perspektyw. Można go zakwalifikować jako problem uczenia ze wzmocnieniem, gdyż gra jest sekwencyjnym problemem decyzyjnym.

Niektóre algorytmy (np. metoda różnic czasowych [4]) bazują na tym fakcie, korzystając z informacji o obserwowanej zależności pomiędzy kolejnymi stanami.

Jednak szukać strategii gier można także metodami optymalizacji globalnej (np. algorytmy ewolucyjnymi [5] lub koewolucyjnymi [6] w przypadku gier wieloosobowych). Algorytmy te ignorują informację o strukturze problemu i traktują problem szukania strategii gry jako problem optymalizacyjny (ang. black-box optimization; w przypadku gier jednoosobowych) lub koopptymalizacyjny (w przypadku gier wieloosobowych, [7]). Funkcja celu jest dobrze określona (np. częstotliwość wygranych), ale w praktyce niemożliwa do precyzyjnego obliczenia ze względu na liczbę możliwych testów, na których trzeba ocenić daną strategię (np. stanów początkowych, przeciwników). Dlatego, z tej perspektywy, można problem szukania strategii gry można nazwać problemem opartym na testach (ang. test-based problem [8]).

4.3.3 Osiągnięte Wyniki

Wyniki, które uzyskałem w prezentowanym tutaj cyklu prac dotyczą algorytmów uczenia się strategii gier takich jak algorytmy (ko)ewolucyjne oraz metody różnic czasowych, efektywnych aproksymatorów funkcji dla gier oraz (wielokryterialnych) metod porównywania algorytmów uczenia się oraz strategii przez nie wygenerowanych.

Profile Jakości Różne miary oceny strategii spotykane w literaturze agregują wyniki wielu gier do jednej wielkości skalarnej. Otrzymana w ten sposób liczba nie jest w stanie w pełni odzwierciedlić różnic i podobieństw pomiędzy porównywanymi strategiami. W szczególności, agregacja prowadzi do potencjalnej kompensacji: nagrody uzyskiwane przez ocenianą strategię za wygrane z pewną grupą przeciwników mogą znosić się z karami za porażki. W konsekwencji dwie strategie mogą uzyskiwać identyczną wartość miary oceny nawet, gdy wyniki ich konfrontacji z tymi samymi przeciwnikami są odmienne.

Brak spójnej, informatywnej metodyki porównywania i oceny strategii, a więc także algorytmów uczących, które je produkują, hamuje więc rozwój w tej dziedzinie.

W [A8] zaproponowano ogólną wielokryterialną metodę porównywania algorytmów dla problemów opartych na testach, która pozwala na uniknięcie straty informacji w wyniku agregacji ocen. Metoda profili jakości (ang. performance profiles) charakteryzuje rozwiązania za pomocą wektora wyników interakcji z testami o różnej trudności. Aby zademonstrować użyteczność metody, zastosowano ją aby przeanalizować zachowanie graczy w Othello wygenerowanych przez pięć algorytmów koewolucyjnych.

Profile jakości pozwalają na odkrycie interesujących różnic pomiędzy graczami, których nie dostrzega się, gdy porównuje się je za pomocą skalarów (np. średnich). W szczególności, pozwalają one zaobserwować, że ewolucja z losowym próbkowaniem przeciwników znajduje strategie sprawujące się dobrze przeciwko średniej sily przeciwnikom, podczas gdy strategie będący efektem koewolucji radzą sobie lepiej przeciwko silnym przeciwnikom.

Metodykę tę rozwinięto następnie w [A4], w której przeprowadzono także eksperymenty dla problemu Iterowanego Dylematu Więźnia. Praca na formalizuje tę koncepcję oraz proponuje dwa algorytmy generowania danych dla profili.

Profile jakości zostały użyte także w [A9], w której to pracy wielokryterialnie porównano metody szukania strategii gier: metodę różnic czasowych oraz algorytm koewolucyjny. Zastosowano tam

następujące kryteria porównawcze: i) oczekiwana użyteczność, ii) średni wynik przeciwko ręcznie zaprojektowanej heurystyce oraz iii) wynik gry bezpośredniej. Dodatkowo, algorytmy porównano za pomocą profili jakości. Mimo iż, oczekiwana użyteczność agentów znalezionych przez obie metody jest podobna, strategie znalezione za pomocą metody różnic czasowych są znacznie lepsze w grze przeciwko silnym przeciwnikom niż strategie znalezione przez algorytm koewolucyjny, który koncentruje się na skuteczności gry z słabszymi przeciwnikami. Okazało się także, że metoda różnic czasowych produkuje strategie mniej zróżnicowane niż koewolucja. Badania te potwierdziły użyteczność profili jakości jako narzędzia do porównywania i analizy strategii i algorytmów.

Profile jakości wydają się więc skuteczną metodą pozwalającą na lepsze zrozumienie charakterystyki algorytmów optymalizacyjnych stosowanych do dowolnych problemów opartych na testach.

Systematyczne sieci n-krotkowe Dla zadań uczenia się ze wzmocnieniem, w których stany mają charakter dyskretny atrakcyjnymi aproksymatorami funkcji, które nie wymagają ręcznego konstruowania cech są (nieliniowe) sieci n-krotkowe. Aproksymatory te były przedmiotem badań w [A7], gdzie m.in. rozważano problem skalowalności sieci n-krotkowych i ich skuteczność w połączeniu z algorytmami ewolucyjnymi i koewolucyjnymi.

Okazuje się, że skuteczność sieci n-krotkowych istotnie zależy od ich kształtu i rozplanowania na dwuwymiarowej planszy do gry. Najbardziej popularna dotychczas metoda zakładała losowe generowanie małej liczby złożonych n-krotek. W [A6] wykazano, że uczenie się parametrów sieci n-krotkowych jest łatwiejsze, jeśli użyje się do tego dużej liczby systematycznie umieszczonych prostych krotek. Zademonstrowano, że dla Othello, klasyczne strategie ewolucyjne umożliwiają wyewoluowanie 2-krotkowych systematycznych sieci, które są efektywniejsze niż najlepsze dotychczas znane rozwiązania dla problemu 1-pły Othello (w benchmarku Othello League). Wyewoluowane sieci mają zaledwie 288 parametrów, czyli 10 razy mniej niż dotychczasowe najlepsze rozwiązania.

Systematyczne sieci n-krotkowe okazują się skuteczne, niezależnie od zastosowanej metody uczenia się, dla środowisk o umiarkowanej wymiarowości, co potwierdzono dla gry 2048 uczoney wariantem metody różnic czasowych [A1], Othello, gdzie zastosowano kompetytywny algorytm koewolucyjny [A3] oraz wariantu gry Tetris, gdzie stosowano algorytmy ewolucyjne oraz metody różnic czasowych [A5].

Algorytmy uczenia się ze wzmocnieniem W [A3] zaproponowano Co-CMAES, kompetytywny wariant Covariance Matrix Adaptation Evolution Strategy (CMAES). Algorytm ten zastosowano do nauki funkcji oceny planszy dla gry Othello. Pokazano, że w porównaniu z klasycznie używanymi w koewolucji strategiami ewolucyjnymi, Co-CMAES uczy się znacznie szybciej, znajduje lepsze rozwiązania i skaluje się istotnie lepiej wraz ze wzrostem liczby parametrów. Jego zalety są wyraźne w szczególności dla dużych przestrzeni rozwiązań (tysiące optymalizowanych parametrów). Dla (1-pły) Othello, połączenie Co-CMAES z systematycznymi sieciami n-krotkowymi zaowocowało znalezienie lepszych strategii od najlepszych dotychczas znanych.

Algorytm Co-CMAES daje się zastosować do nauki modeli, które mają tysiące parametrów, ale jest zbyt wymagający obliczeniowo gdy problem wymaga aproksymatorów funkcji z milionami parametrów. Do takich problemów należy gra SZ-Tetris. Jest to ograniczona wersja gry Tetris i trudny problem uczenia się ze wzmocnieniem. Wcześniejsze prace pokazały, że, podobnie do oryginalnej gry Tetris, metody próbujące estymować funkcję wartości stanów takie jak metoda różnic czasowych, nie są dla

tego problemu skuteczne. Najlepsze do tej pory rozwiązania dla tej gry uzyskiwano za pomocą metod przeszukiwania bezpośredniego, np. CME (Cross-Entropy Method), chociaż były one wciąż gorsze niż prosta ręcznie napisana heurystyka. W pracy [A5] pokazano, że lepsze wyniki osiągnąć można stosując systematyczne sieci n-krotkowych (vide [A6]) i VD-CMA-ES, liniowego wariantu CMA-ES, jako metody optymalizacji dla problemów wysokowymiarowych. Pokazano także, że użycie dużej systematycznej sieci n-krotkowej (4 miliony parametrów) pozwala osiągnąć wynik zbliżony do VD-CMA-ES przy użyciu metody różnic czasowych przy 20-krotnie mniejszym koszcie obliczeniowym. Najlepsza osiągnięta strategia jest lepsza niż jakakolwiek dotąd znana strategia dla tej gry.

Kolejny wątek badań rozwijał metody uczenia się ze wzmocnieniem dla znanych procesów decyzyjnych Markova. Jako benchmark wybrano jednoosobową niedeterministyczną grę 2048, która w 2015 roku zdobyła dużą popularność na urządzeniach mobilnych. W pracy [A1] podjęto próbę automatycznego (bez użycia wiedzy eksperckiej) znalezienia strategii dla gry 2048. Jako aproksymatora funkcji użyto systematycznej sieci n-krotkowej dostosowanej dla niewielkiej ale wielowartościowej planszy 2048. Pokazano, że klasyczna metoda różnic czasowych może być istotnie ulepszona dzięki zaproponowanym technikom: i) wieloetapowemu aproksymatorowi funkcji z promocją wagi, oraz ii) kształtowaniem karuzelowym. Dodatkowo, opracowano dwie techniki zwiększające obliczeniową efektywność procesu uczenia: iii) opóźnioną aktualizację wag oraz iv) optymistyczne nieblokujące zrównoleglenie. Synergia zaproponowanych technik pozwoliła na opracowanie najlepszego znanego do tej pory programu grającego w 2048 uwzględniając wszystkie dostępne publicznie algorytmy i metody zaprojektowane ręcznie. Warto zaznaczyć, że nauczona strategia uzyskuje w 2048 istotnie lepsze rezultaty niż człowiek. Mimo że od udostępnienia (online) publikacji na ten temat (wraz z kodem źródłowym) minęły już dwa lata, do tej pory nikomu nie udało się opracować lepszej strategii dla tej gry.

Wyniki ilościowe Niektóre z wyżej wymienionych badań doprowadziło do osiągnięcia najlepszych znanych wyników dla poszczególnych problemów testowych. Algorytm ETDL [A7] uzyskał najlepszy znany wynik w (już nieistniejącej) Online Othello League (89% wygranych przeciwko standardowemu przeciwnikowi SWH). Wynik algorytmu poprawiono rok później w pracy [O1] (dostępna online) stosującej metodę z [A6] uzyskując 96%.

Wyniki uzyskane w Lidze Othello często nie przekładają się na wygrane z innymi przeciwnikami niż SWH [A7]. Dlatego tak istotne są porównania turniejowe (każdy z każdym). Taki turniej uwzględniający wszystkie 14 dotychczas najlepszych opublikowanych strategii 1-ply przeprowadzony został na potrzeby pracy [A3]. Obejmował on 182 000 rozgrywek i pokazał wyższość strategii nauczonej za pomocą Co-CMAES [A3]. Wygrywając 90% gier turniejowych Co-CMAES pokonało dwa najsilniejsze dotychczas algorytmy ([9], 82%, i [10], 79%).

Wynik ten istotnie poprawiony został niedawno [A2], gdzie jedna z zaproponowanych w pracy sieci konwolucyjnych uzyskała w turnieju z tymi samymi przeciwnikami 97% zwycięstw, pokonując Co-CMAES w aż 93% gier. Sieć konwolucyjna okazała się także zdecydowanie lepsza niż Edax (w wersji 2-ply), jeden z najsilniejszych programów dla Othello. Warto jednak podkreślić, że podejście zaprezentowane w [A2] używa wiedzy dziedzinowej (bazy rozgrywek), czego algorytm Co-CMAES nie wykorzystuje.

Strategia dla gry 2048 uzyskana za pomocą metody opisanej w [A1] jest dostępna publicznie od grudnia 2015 roku i dotychczas nie została pokonana.

Strategia dla gry SZ-Tetris uzyskana w 2015 roku [A5] została pokonana dopiero niedawno [11].

5 Omówienie pozostałych osiągnięć naukowo-badawczych

Osiągnięcia naukowe opisane w sekcji 4.3 stanowią tylko część badań, które dotyczą gier, algorytmów koewolucyjnych i metod uczenia ze wzmocnieniem, nad którymi pracowałem. Jednym z najciekawszych innych wątków stanowi cykl prac o algorytmie koewolucyjnym, który nie używa miary przystosowania (ang. fitnessless coevolution) [O2, O3]. Metoda ta dla gier kompetytywnych zamiast oceniać osobniki, a następnie wykonywać selekcję na podstawie przyznanych ocen, proponuje dokonywać selekcji na podstawie wyników bezpośrednich interakcji w ramach turnieju pucharowego. Metoda ta ma szereg interesujących właściwości teoretycznych. Została skutecznie zastosowana do gry Antwars [O4].

Gry były także tematem prac proponujących hybrydyzację algorytmów ewolucyjnych i metody różnic czasowych [O5, O6, O7]. Taka hybrydyzacja pozwala łączyć zalety obu rodzin algorytmów (lepsze właściwości eksploracyjne ewolucji i szybsze lokalne przeszukiwanie metody różnic czasowych). Proponowane algorytmy były testowane dla gry Othello oraz Go.

[O8] praca była pierwszą pracą dotyczącą gry 2048 (por. [A1]), w której wstępnie pokazano, że metoda różnic czasowych jest szybko w stanie osiągać dobre wyniki dla tej gry bez użycia wiedzy dziedzinowej.

[O9] opisuje jedno z pierwszych badań poświęconych uczeniu się strategii gier czasu rzeczywistego i pierwszą pracą, w której podjęto próbę rozwiązania tego problemu za pomocą metod programowania genetycznego.

Z kolei [O10] opisuje badania nad DQN (Deep Q-Networks) w kontekście niskowymiarowego problemu uczenia się ze wzmocnieniem (Keepaway Soccer). Problem ten oparty jest na realistycznym, fizycznym symulatorze robotycznej piłki nożnej. Algorytm DQN został wcześniej zaproponowany dla wysokowymiarowych problemów wizualnego uczenia się ze wzmocnieniem. W pracy [O10] zadano pytanie, czy jest on równie efektywny dla problemów niskowymiarowych (mała liczba rzeczywistych cech) oraz które z jego komponentów są w tym kontekście najważniejsze. Eksperymenty dowiodły, że algorytm jest skuteczny, ale kluczową rolę odgrywa tylko jeden z jego elementów: experience replay. Dzięki optymalizacji hiperparametrów, pokazano, że DQN dla Keepaway Soccer osiąga lepsze wyniki niż (najlepsza znany do tej pory dla tego problemu) metoda programowania genetycznego używając ułamka mocy obliczeniowej wykorzystywanej przez konkurencyjne metody.

Idąc tropem ostatniego przełomu w uczeniu maszynowym dla zadań klasyfikacji obrazów, w [O11] zaproponowano platformę do eksperymentowania z algorytmami uczenia maszynowego (w tym głębokiego), a w szczególności uczenia ze wzmocnieniem. Platforma oparta jest na grze Doom, a agenty mają do dyspozycji jedynie informację wizualną - 'widzą' tylko to co widzi człowiek w taką grę grający. Środowisko więc jest częściowoobserwowalne, jest środowiskiem czasu rzeczywistego, a stany są wysokowymiarowymi obrazami 2D lub 3D (głębia). W środowisku przeprowadzono kilka eksperymentów obliczeniowych. Pokazano, że wariant algorytmu DQN potrafi znaleźć skuteczne rozwiązania dla prostych lecz nietrywialnych zadań w tym środowisku.

Mój dorobek zawiera także cykl artykułów poświęconych problemom opartym na testach (ang. test-based problems), które były tematem mojej rozprawy doktorskiej [O12, O13, O14, O15]. Nowsza praca dotyczącej tej problematyki [O16] pokazuje, że liczbę interakcji w takich problemach można

znacznie zredukować za pomocą adaptacyjnej metody faktoryzacji macierzy. [O17] proponuje metodę kształtowania (ang. *shaping*) do wyboru testów o odpowiedniej trudności. Oprócz trudności testów ważna jest także ich różnorodność, co pokazano w [O18].

Przez szereg lat zajmowałem się problematyką syntezy programów za pomocą programowania genetycznego. Efektem tych badań jest cykl prac [O19, O20, O21, O22, O23, O24, O25, O26, O27], proponujący algorytmy i operatory genetyczne dla problemów uczenia się z informacji obrazowych. Rozpoznawanie obrazów ma tutaj charakter generatywny (“rozpoznaję kształt, jeśli potrafię go odtworzyć”). W ramach badań opracowałem nowe metody przenoszenia i współdzielenia wiedzy pomiędzy systemami uczącymi się.

Programowanie genetyczne przez wiele lat cierpiało na brak standaryzowanych problemów testowych, które pozwoliłyby bezpośrednio porównywać proponowane algorytmy, co hamowało postęp w tej dziedzinie. Wziąłem udział w pracach nad konsensusową propozycją zbioru problemów testowych dla programowania genetycznego [O28, O29].

W ramach projektu Lider (Narodowe Centrum Badań i Rozwoju), za pomocą metod optymalizacji wielokryterialnej szukałem najlepszego profilu koła tramwajowego, co jest istotnym zagadnieniem praktycznym wpływającym na komfort jazdy pojazdami szynowymi oraz koszty ich eksploatacji [O30, O31].

Efektem mojej współpracy z Wydziałem Fizyki Uniwersytetu Adama Mickiewicza w Poznaniu jest cykl prac z neuronauki dotyczący wpływu wyobraźni motorycznej na efektywność uczenia się zadań motorycznych [O32, O33, O34].

Jestem autorem także kilku pojedynczych prac o innej tematyce. W [O35] zaproponowałem efektywny algorytm dla postawionego przez Google problemu “Machine Reassignment”. Problem ten ma istotne znaczenia dla optymalizacji wykorzystania zasobów w centrach obliczeniowych (optymalizacja kombinatoryczna). W [O36] opracowałem obliczeniową miarę symetrii dla sztucznych organizmów i pokazałem jej zastosowania (sztuczne życie). W pracy [O37] opisałem efektywne algorytmy wyboru najlepszego spośród zbioru osobników dla środowisk stochastycznych przy zadanym budżecie obliczeniowym (obliczenia ewolucyjne). [O38] jest efektem mojej pracy nad podejściem metaserwerowym dla problemu predykcji struktury białek (biologia obliczeniowa).

Lista prac – Pozostałe Osiągnięcia

- [O1] Jaśkowski, W. Systematic n-tuple networks for position evaluation: Exceeding 90% in the othello league. Tech. Rep. RA-06/2014, arXiv:1406.1509, Institute of Computing Science, Poznan University of Technology, Poznań, Poland (2014).
- [O2] Jaśkowski, W., Krawiec, K. & Wieloch, B. Evolving strategy for a probabilistic game of imperfect information using genetic programming. *Genetic Programming and Evolvable Machines* **9**, 281–294 (2008). [**IF 2009: 1.091**]
- [O3] Jaśkowski, W., Wieloch, B. & Krawiec, K. Fitnessless coevolution. In Keijzer, M. (ed.) *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 355–362. Association for Computing Machinery (Association for Computing Machinery, Atlanta, GA, USA, 2008).

- [O4] Jaśkowski, W., Krawiec, K. & Wieloch, B. Winning Ant Wars: Evolving a human-competitive game strategy using fitnessless selection. In O’Neill, M. (ed.) *Genetic Programming 11th European Conference, EuroGP 2008, Proceedings*, vol. 4971 of *Lecture Notes in Computer Science*, 13–24 (Springer-Verlag, 2008).
- [O5] Krawiec, K., Jaśkowski, W. & Szubert, M. Evolving small-board go players using coevolutionary temporal difference learning with archive. *International Journal of Applied Mathematics and Computer Science* **21**, 717–731 (2011). [**IF 2011: 0.487**]
- [O6] Szubert, M., Jaśkowski, W. & Krawiec, K. Learning board evaluation function for othello by hybridizing coevolution with temporal difference learning. *Control and Cybernetics* **40**, 805–831 (2011).
- [O7] Szubert, M., Jaśkowski, W. & Krawiec, K. Coevolutionary temporal difference learning for othello. In *IEEE Symposium on Computational Intelligence and Games*, 104–111 (Milano, Italy, 2009).
- [O8] Szubert, M. & Jaśkowski, W. Temporal difference learning of n-tuple networks for the game 2048. In *IEEE Conference on Computational Intelligence and Games*, 1–8 (IEEE, Dortmund, 2014).
- [O9] Lichocki, P., Krawiec, K. & Jaśkowski, W. Evolving teams of cooperating agents for real-time strategy game. In Giacobini, M. *et al.* (eds.) *Applications of Evolutionary Computing, EvoWorkshops*, vol. 5484 of *Lecture Notes in Computer Science*, 333–342 (Springer, 2009).
- [O10] Kurek, M. & Jaśkowski, W. Heterogeneous team deep q-learning in low-dimensional multi-agent environments. In *IEEE Conference on Computational Intelligence and Games*, 201–208 (IEEE, Santorini, Greece, 2016).
- [O11] Kempka, M., Wydmuch, M., Runc, G., Toczek, J. & Jaśkowski, W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*, 341–348 (IEEE, Santorini, Greece, 2016).
- [O12] Jaśkowski, W. & Krawiec, K. Formal analysis, hardness and algorithms for extracting internal structure of test-based problems. *Evolutionary Computation* **19**, 639–671 (2011). [**IF 2011: 1.061**]
- [O13] Jaśkowski, W. & Krawiec, K. How many dimensions in cooptimization? In Krasnogor, N. (ed.) *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, 829–830 (Association for Computing Machinery, 2011).
- [O14] Jaśkowski, W. & Krawiec, K. Coordinate system archive for coevolution. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 1–10 (IEEE, Barcelona, 2010).
- [O15] Jaśkowski, W. & Krawiec, K. Formal analysis and algorithms for extracting coordinate systems of games. In *IEEE Symposium on Computational Intelligence and Games*, 201–208 (Milano, Italy, 2009).

- [O16] Liskowski, P. & Jaśkowski, W. Accelerating coevolution with adaptive matrix factorization. In *GECCO'17: Proceedings of the 19th annual conference on Genetic and Evolutionary Computation*. ACM (ACM Press, Berlin, Germany, 2017).
- [O17] Szubert, M., Liskowski, P., Jaśkowski, W. & Krawiec, K. Shaping fitness function for evolutionary learning of game strategies. In Blum, C. (ed.) *GECCO'13: Proceedings of the 15th annual conference on Genetic and Evolutionary Computation*, 1149–1156 (ACM, Amsterdam, The Netherlands, 2013).
- [O18] Szubert, M., Jaśkowski, W., Liskowski, P. & Krawiec, K. The role of behavioral diversity and difficulty of opponents in coevolving game-playing agents. In Mora, A. M. & Squillero, G. (eds.) *EvoApplications 2015*, vol. 9028 of *Lecture Notes in Computer Science*, 394–405 (Springer, Copenhagen, Denmark, 2015).
- [O19] Jaśkowski, W., Krawiec, K. & Wieloch, B. Cross-task code reuse in genetic programming applied to visual learning. *International Journal of Applied Mathematics and Computer Science* **24**, 183–197 (2014). [**IF 2014: 1.227**]
- [O20] Jaśkowski, W., Krawiec, K. & Wieloch, B. Multitask visual learning using genetic programming. *Evolutionary Computation* **16**, 439–459 (2008). [**IF 2008: 3.000**]
- [O21] Jaśkowski, W., Krawiec, K. & Wieloch, B. Genetic programming for generative learning and recognition of hand-drawn shapes. In Cagnoni, S. (ed.) *Evolutionary Image Analysis and Signal Processing*, vol. 213 of *Studies in Computational Intelligence*, 281–290 (Springer Berlin / Heidelberg, 2009).
- [O22] Jaśkowski, W., Krawiec, K. & Wieloch, B. Multi-task code reuse in genetic programming. In Ebner, M. *et al.* (eds.) *GECCO-2008 Late-Breaking Papers*, 2159–2164. Association for Computing Machinery (Association for Computing Machinery, Atlanta, GA, USA, 2008).
- [O23] Jaśkowski, W., Krawiec, K. & Wieloch, B. Genetic programming for cross-task knowledge sharing. In Thierens, D. (ed.) *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, vol. 2, 1620–1627. Association for Computing Machinery (Association for Computing Machinery, London, 2007).
- [O24] Jaśkowski, W., Krawiec, K. & Wieloch, B. Evolutionary learning with cross-class knowledge reuse for handwritten character recognition. In Brazdil, P. & Bernstein, A. (eds.) *Proceedings of Planning to Learn Workshop, PlanLearn'07*, 21–30 (2007).
- [O25] Jaśkowski, W., Krawiec, K. & Wieloch, B. Knowledge reuse for an ensemble of GP-based learners. In Arabas, J. (ed.) *Evolutionary Computation and Global Optimization 2007*, vol. 160 of *Prace Naukowe Politechniki Warszawskiej*, 135–142 (Oficyna Wydawnicza Politechniki Warszawskiej, Bedlewo, Poland, 2007).
- [O26] Jaśkowski, W., Krawiec, K. & Wieloch, B. Learning and recognition of hand-drawn shapes using generative genetic programming. In Giacobini, M. (ed.) *Applications of Evolutionary Computing, EvoWorkshops 2007: EvoCOMNET, EvoFIN, EvoIASP, EvoInteraction, EvoMUSART, EvoSTOC, EvoTransLog*, vol. 4448 of *LNCS*, 281–290 (Springer Verlag, Valencia, Spain, 2007).

- [O27] Jaśkowski, W., Krawiec, K. & Wieloch, B. Knowledge reuse in genetic programming applied to visual learning. In Thierens, D. (ed.) *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, vol. 2, 1790–1797. Association for Computing Machinery (Association for Computing Machinery, London, 2007).
- [O28] White, D. R. *et al.* Better GP benchmarks: community survey results and proposals. *Genetic Programming and Evolvable Machines* **14**, 3–29 (2013). [IF 2013: 1.065]
- [O29] McDermott, J. *et al.* Genetic programming needs better benchmarks. In Soule, T. (ed.) *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, 791–798. ACM (ACM, 2012).
- [O30] Staśkiewicz, T., Firlik, B., Jaśkowski, W. & Wittenbeck, L. On developing tram wheel new profile using multibody simulation tools. In Ambrosio, J. A., Schiehlen, W. & Pombo, J. (eds.) *Proceedings of EUROMECH Colloquium 578: Rolling Contact Mechanics for Multibody System Dynamics*, 1–11 (Instituto Superior TÁŠcnico Universidade de Lisboa, 2017).
- [O31] Staśkiewicz, T., Firlik, B., Jaśkowski, W. & Wittenbeck, L. On designing a durable and safe tram wheel profile. In Spiriyagin, M., Gordon, T., Cole, C. & McSweeney, T. (eds.) *Dynamics of Vehicles on Roads and Tracks: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017)*, 1–6 (CRC Press, 2017).
- [O32] Sobierajewicz, J., Przekoracka-Krawczyk, A., Jaśkowski, W. & van der Lubbe, R. H. J. How effector-specific is the effect of sequence learning by motor execution and motor imagery? *Experimental Brain Research* **12**, 3757–3769 (2017). [IF 2016: 1.917]
- [O33] Sobierajewicz, J., Przekoracka-Krawczyk, A., Jaśkowski, W., Verwey, W. & Lubbe, R. The influence of motor imagery on the learning of a fine hand motor skill. *Experimental Brain Research* **235**, 305–320 (2017). [IF 2016: 1.917]
- [O34] Sobierajewicz, J., Szarkiewicz, S., Przekoracka-Krawczyk, A., Jaśkowski, W. & van der Lubbe, R. To what extent can motor imagery replace motor execution while learning a fine motor skill? *Advances in Cognitive Psychology* **12**, 179–192 (2016). [IF 2016: 1.357]
- [O35] Jaśkowski, W., Szubert, M. & Gawron, P. A hybrid MIP-based large neighborhood search heuristic for solving the machine reassignment problem. *Annals of Operations Research* **242**, 33–62 (2016). [IF 2016: 1.709]
- [O36] Jaśkowski, W. & Komosiński, M. The numerical measure of symmetry for 3D stick creatures. *Artificial Life* **14**, 425–443 (2008). [IF 2008: 1.164]
- [O37] Jaśkowski, W. & Kotłowski, W. On selecting the best individual in noisy environments. In Keijzer, M. *et al.* (eds.) *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 961–968. Association for Computing Machinery (Association for Computing Machinery, Atlanta, GA, USA, 2008).
- [O38] Jaśkowski, W., Blazewicz, J., Lukasiak, P., Milostan, M. & Krasnogor, N. 3D-Judge — a metaserver approach to protein structure prediction. *Foundations of Computing and Decision Sciences* **32**, 3–14 (2007).

Literatura

- [1] Donkers, J., Uiterwijk, J. & de Voogt, A. Mancala games-topics in mathematics and artificial intelligence. *Board Games in Academia* 17–21 (2001).
- [2] Russell, S. J. & Norvig, P. *Artificial intelligence: A Modern approach* (Prentic Hall, 2010).
- [3] Mańdziuk, J. *Knowledge-Free and Learning-Based Methods in Intelligent Game Playing*, vol. 276 (Springer-Verlag New York Incorporated, 2010).
- [4] Sutton, R. & Barto, A. *Reinforcement learning*, vol. 9 (MIT Press, 1998).
- [5] Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning* (Addison-wesley, 1989).
- [6] Axelrod, R. The evolution of strategies in the iterated prisoner's dilemma. In Davis, L. (ed.) *Genetic Algorithms in Simulated Annealing*, 32–41 (Pitman, London, 1987).
- [7] Popovici, E., Bucci, A., Wiegand, R. P. & de Jong, E. D. Coevolutionary principles. In Rozenberg, G., Back, T. & Kok, J. N. (eds.) *Handbook of Natural Computing* (Springer-Verlag, 2011).
- [8] Bucci, A. & Pollack, J. B. A mathematical framework for the study of coevolution. In De Jong, K. A., Poli, R. & Rowe, J. E. (eds.) *Foundations of Genetic Algorithms 7*, 221–236 (Morgan Kaufmann, San Francisco, 2003).
- [9] Manning, E. P. Using resource-limited nash memory to improve an othello evaluation function. *Computational Intelligence and AI in Games, IEEE Transactions on* **2**, 40–53 (2010).
- [10] Runarsson, T. & Lucas, S. Preference Learning for Move Prediction and Evaluation Function Approximation in Othello. *Computational Intelligence and AI in Games, IEEE Transactions on* **6**, 300–313 (2014).
- [11] Elfving, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks* (2018).


Wojciech Jaśkowski
(podpis)