

# Równoległe i Rozproszone Obliczenia Ewolucyjne w Poszukiwaniach Morskich

Tadeusz Puźniakowski

## Streszczenie

Rozprawa jest podzielona na osiem rozdziałów i trzy dodatki.

W rozdziale 1 zarysowano praktyczny problem wiodący: dysponując wiedzą na temat możliwego obszaru przebywania  $k$  rozbitków na morzu, gdzie  $k \geq 1$ , wyznacz optymalną strategię ich poszukiwania za pomocą jednej jednostki poszukującej. Problem ten jest naturalnym uogólnieniem zagadnienia poszukiwania jednego rozbitka przez jedną jednostkę poszukującą, badanego w ramach projektu badawczo-rozwojowego, w którym uczestniczył autor — „Platforma badawcza oraz prototyp mobilnego stanowiska planowania nawodnych poszukiwań morskich w oparciu o teorię algorytmów ewolucyjnych.”. W projekcie tym zastosowano algorytm ewolucyjny korzystający z symulatora ruchu rozbitka na morzu do oceny rozwiązań.

Przedstawiono hipotezę badawczą pracy mówiącą o tym, że korzystając z dostępnych metod zrównoleglania obliczeń można znacząco przyspieszyć proces generowania efektywnych strategii poszukiwań wielu rozbitków za pomocą algorytmów genetycznych prowadząc obliczenia równoległe i/lub rozproszone na powszechnie dostępnym sprzęcie.

Kolejne zagadnienia wymienione we wstępie, to:

- Sprawdzenie jakości strategii drzewiastych w porównaniu do strategii sekwencyjnych w poszukiwaniu rozbitków.
- Porównanie wydajności oferowanych przez rozwiązania wykorzystujące heterogeniczną architekturę sprzętową z rozwiązaniami tradycyjnymi, opartymi wyłącznie o CPU.
- Oszacowanie kosztu prac programistycznych, niezbędnych do adaptacji oprogramowania do efektywnej pracy z heterogenicznymi architekturami sprzętowymi.

Dalsza część wstępu jest poświęcona klasycznemu podejściu do planowania akcji ratowniczych na morzu a także pracom prowadzonym w Polsce, w ramach projektów badawczo-rozwojowych zajmujących się zastosowaniem teorii zbiorów rozmytych.

W rozdziale 2 omówiono w skrócie podstawowe technologie pozwalające na zrównoleglenie obliczeń: OpenMP – standard do prowadzenia obliczeń równoległych na maszynach z pamięcią współdzieloną, MPI – standard do komunikacji międzyprocesowej w klastrach komputerów

oraz OpenCL – standard do prowadzenia obliczeń w heterogenicznych środowiskach obliczeniowych. Wszystkie wybrane standardy są otwarte, darmowe i nieprzypisane do jednego konkretnego podmiotu.

Rozdział 3 opisuje pierwszy z przeprowadzonych eksperymentów porównujących wydajność zrównoleglania obliczeń przy pomocy OpenMP i OpenCL, na różnego rodzaju sprzętach (CPU i GPU). W eksperymencie tym wybrano prosty problem algorytmiczny jakim jest mnożenie macierzy. Testowany algorytm, w przypadku uruchomienia za pośrednictwem OpenCL na wielordzeniowym CPU, osiąga znacznie lepsze rezultaty niż ten sam algorytm zaimplementowany z wykorzystaniem tradycyjnie w takim przypadku stosowanego OpenMP.

Rozdział 4 poświęcony jest analizie struktur danych przydatnych do reprezentowania drzewiastych rozwiązań problemu poszukiwania wielu obiektów z wykorzystaniem strategii generowanej offline. Zaprezentowano w nim propozycję opisu strategii za pomocą drzew. Przeprowadzona w tym rozdziale analiza złożoności pamięciowej, drzewiastej reprezentacji strategii wskazuje, że mamy do czynienia z *lekkimi drzewami*, to znaczy z drzewami o wielomianowej wielkości.

Z implementacyjnego punktu widzenia tradycyjna drzewiasta reprezentacja danych nie sprawdza się w przypadku zastosowania jej na GPU. Z tego powodu zdefiniowano i udowodniono bijekcję między współrzędnymi węzłów drzewa strategii, a indeksami wektora reprezentującego to drzewo.

Rozdział 5 przedstawia szczegóły związane z implementacją oraz organizacją kodu programu, który wykorzystano do testów, opisanych w rozdziale 6. Przedstawiono w nim własną, opartą o pracę Blum-Blum-Schrub wersję generatora liczb pseudolosowych wraz z oceną wiarygodności uzyskanej implementacji. Okazało się to niezbędne, bowiem framework OpenCL nie zawiera generatora liczb pseudolosowych. Określono także stabilność wyników oceny strategii podawanych przez symulator. Zaobserwowano, że już od 512 symulacji zbieżność wyników podawanych przez symulator jest satysfakcjonująca, wariancja wyników z próbki wielkości 3188 wynosi poniżej 0,00054.

Rozdział 6, poprzedzający eksperymenty, jest poświęcony implementacji Algorytmu Genetycznego dla proponowanej reprezentacji strategii poszukiwań. Wyjaśniono w nim sposób realizacji strategii drzewiastej oraz opisano działanie funkcji oceny, opartej o symulator stworzony na bazie modelu, wykorzystanego w projekcie badawczo rozwojowym realizowanym przez autora w latach 2009/2010. Rozdział ten zawiera także informacje dotyczące poszczególnych operatorów genetycznych wraz z odwołaniem do literatury w przypadku, jeśli jakiś operator został już gdzieś zastosowany.

Rozdział 7 opisuje przeprowadzone eksperymenty oraz uzyskane wyniki.

Czas obliczeń związanych z generowaniem strategii na GPU przy wykorzystaniu technologii OpenCL może być nawet ponad 10-krotnie mniejszy niż w przypadku obliczeń prowadzonych jedynie na CPU.

Wyniki uzyskane przy dokładnie takich samych warunkach początkowych (uwzględniając również ziarno generatora liczb pseudolosowych) mają niewielki rozrzut, w zależności od sprzętu, na którym są wyliczane. Najczęściej wyniki są takie same na tej samej klasie sprzętu, w przypadku obliczeń na jednym hoście. Na przykład na wszystkich kartach NVIDIA wyniki są dokładnie takie same.

Wyniki pierwszych eksperymentów okazały się niezgodne z oczekiwaniami – uzyskane drzewiaste strategie poszukiwań nie były znacząco lepsze, a często wręcz gorsze od strategii sekwencyjnych, dlatego w sekcji 7.6 przeprowadzono pogłębioną analizę natury problemu. Okazało się, po pierwsze, że trudno jest poprawić strategie uzyskane algorytmem genetycznym przy wykorzystaniu algorytmu wspinaczkowego. Uwaga ta dotyczy zarówno strategii drzewiastych jak i sekwencyjnych. Po drugie, twierdzenie o wyższości strategii drzewiastych nad sekwencyjnymi pozostaje w mocy – jeśli uruchomimy algorytm genetyczny na populacji początkowej złożonej z najlepszych strategii sekwencyjnych, to, przeważnie, uzyskamy strategię drzewiastą o lepszej ocenie niż najlepsza sekwencyjna. Na uwagę zasługują wyniki zawarte w tabeli 7.5 prezentujące nawet kilkunastoprocentowy wzrost wartości funkcji oceny.

Rozdział 8 zawiera podsumowanie uzyskanych wyników oraz przedstawia możliwości rozwoju omawianych zagadnień.

Dodatek A traktuje o pierwszym podejściu do problemu implementacji Algorytmu Genetycznego w całości za pomocą OpenCL.

Okazało się, że dla problemów z nieskomplikowaną funkcją oceny — tu wzięty został pod uwagę problem **one-max** — można znacząco przyspieszyć proces znajdowania optymalnych rozwiązań.

Niestety, podczas próby wykorzystania tego podejścia do problemu poszukiwania rozbitków, wyszły na jaw pewne problemy związane z niedojrzałością technologii. W szczególności pojawiały się problemy z kompilacją kodu w OpenCL. Po tych eksperymentach autor postanowił zastosować tradycyjne podejście do algorytmów genetycznych, to znaczy zrównoleglenie tylko funkcji oceny oraz generowania tras rozbitków. Program został jednak napisany z myślą o przeniesieniu całości obliczeń na technologię OpenCL.

Dodatek B zawiera informacje o obsłudze programu załączonego do pracy.

W dodatku C znajdują się teoretyczne rozważania dotyczące analizy porównawczej efektywności strategii drzewiastych i strategii sekwencyjnych poszukiwań w ujęciu probabilistycznym.