

Sterowanie procesami dyskretnymi

Sprawozdanie 4

Zakres sprawozdania:

- Problem przepływowy z kryterium C średnie i opóźnieniami transportowymi
 - Stworzenie rozwiązania przybliżonego za pomocą algotymy NEH
 - Próba ulepszenia otrzymanych wyników za pomocą algorytmu symulowanego wyżarzania
-

Opis problemu

Rozwiązywany problem jest bardzo podobny do tego, który rozwiązywaliśmy na 2 laboratorium. Jest to klasyczny problem przepływowy, lecz tym razem rozwiązywaliśmy go względem kryterium średniego czasu zakończenia zadań oraz dołożyliśmy do niego opóźnienia transportowe. Opóźnienie transportowe oznaczają, że pomiędzy wykonywaniem pracy nad tym samym zadaniem na dwóch różnych maszynach musi upłynąć pewien z góry ustalony okres czasu.

Aby stworzyć rozwiązanie problemu posłużyliśmy się algorytmem NEH, a następnie zastosowaliśmy algorytm symulowanego wyżarzania, aby poprawić otrzymane rezultaty.

Środowisko

System operacyjny: Windows 8 Professional

Platforma programistyczna: .NET 4.5

Język programowania: C#

IDE: Visual Studio 2012 Ultimate (wersja z MSDNAA)

Komputer wyposażony w 2 rdzeniowy procesor korzystający z technologii Hyper-threading (więc mogący wykonywać równolegle 4 operacje)

Implementacja

Implementacja algorytmów, oprogramowania pomocniczego i oprogramowania testującego poprawność algorytmów oraz oprogramowania testującego szybkość działania algorytmów jest dostępna w plikach:

- AutoOrderingOptimization.cs – generyczna klasa z metodami umożliwiającymi przeprowadzenie symulowanego wyżarzania po podaniu handlera do funkcji celu, kontenera z obiektami oraz handler do funkcji mutującej
- FileOperations.cs – klasa pomocnicza do ułatwienia przeprowadzania operacji na plikach
- Program.cs – główna klasa programu
- Task.cs – klasa przechowująca zadanie
- NEHOrdering.cs – klasa pozwalająca wykonać szeregowanie algorytmem NEH
- UnitTest1.cs – klasa przechowująca testy jednostkowe

Implementacja algorytmów została przeprowadzona zgodnie z obecnymi trendami w programowaniu metodą **TDD** (Test Driven Development) – przed implementacją każdej funkcjonalności tworzyliśmy test, który będzie testował poprawność naszej implementacji.

Wyniki

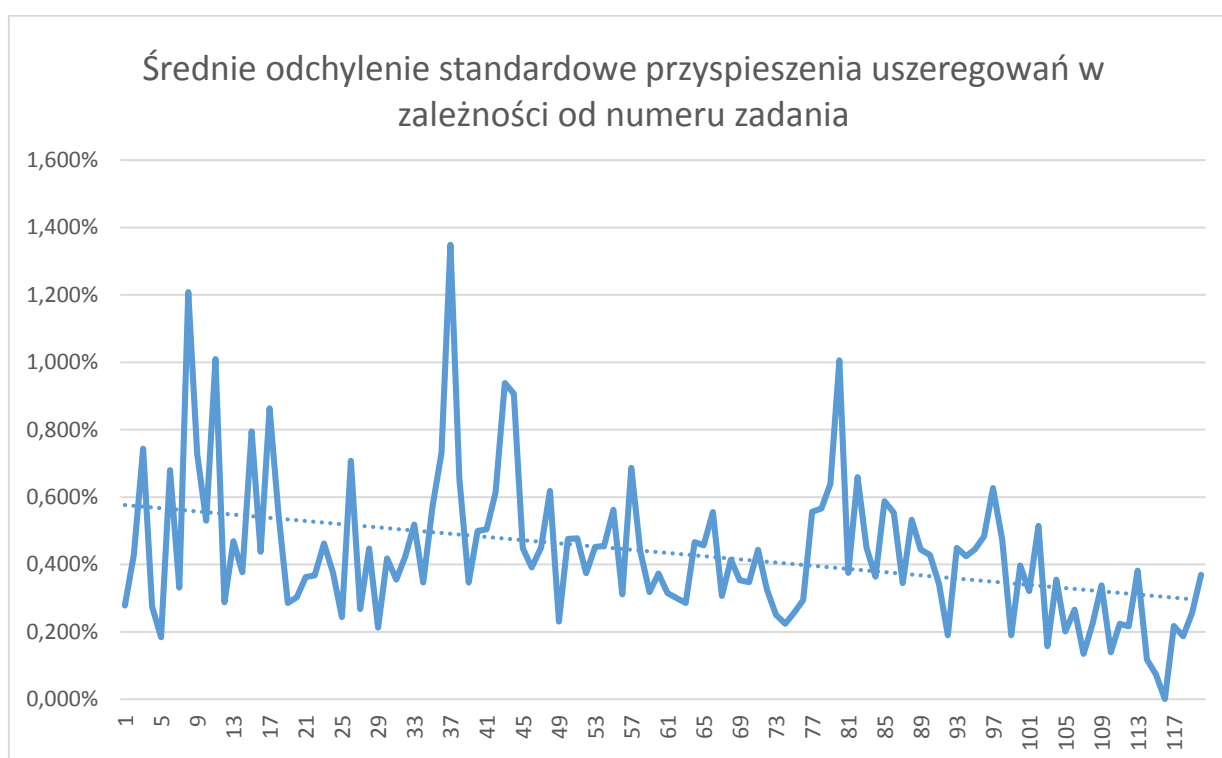
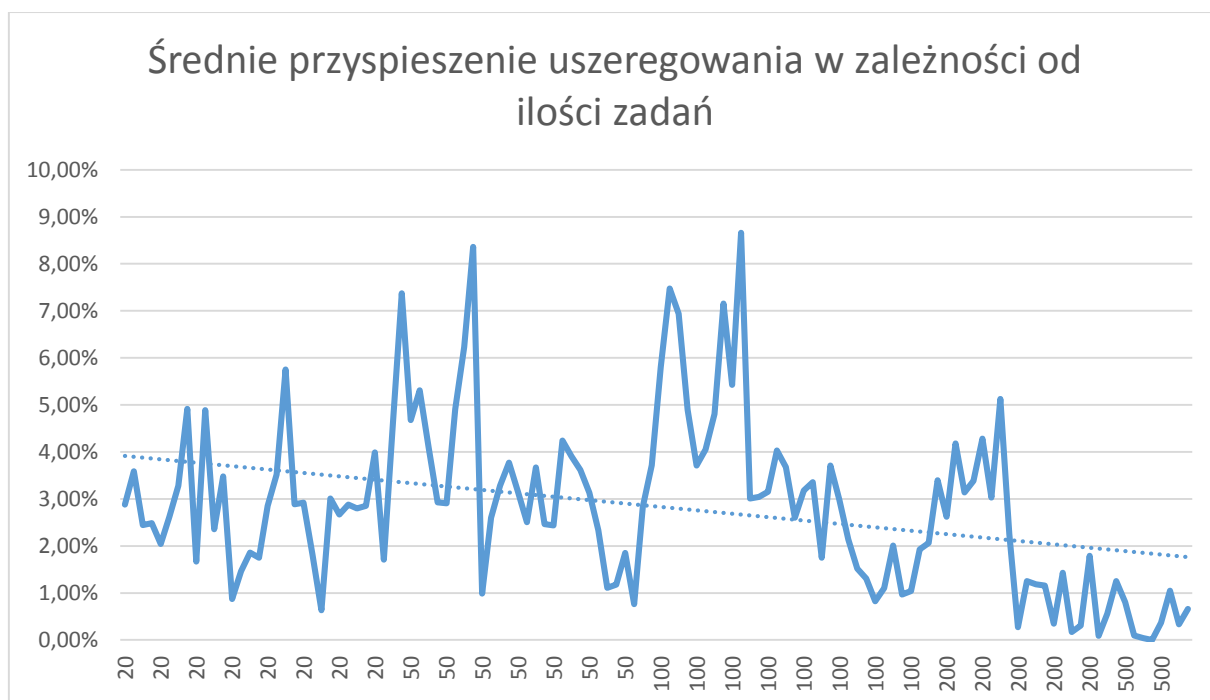
Zaimplementowany przez nas algorytm NEH dla zmodyfikowanego problemu daje takie same wyniki jak ten z programu z Pańskiej strony internetowej. Pozwala to domniemywać, że implementacja daje prawidłowe rezultaty.

Aby poprawić wyniki algorytmu NEH wykorzystaliśmy algorytm symulowanego wyżarzania, a dokładną zmianę wyników można zobaczyć w załączonym pliku „wyniki.xlsx”. Wnioski z jego działania opisane są w rozdziale „wnioski”.

Średnie przyspieszenie uszeregowania przez algorytm symulowanego wyżarzania dla różnych wielkości instancji

Ilość zadań	Ilość maszyn	Średnie przyspieszenie
20	5	3,08%
20	10	2,68%
20	20	2,53%
50	5	5,13%
50	10	2,91%
50	20	2,44%
100	5	5,89%
100	10	3,15%
100	20	1,58%
200	10	3,35%
200	20	0,80%
500	20	0,51%

Wykresy



Wnioski

Wykorzystanie algorytmów heurystycznych takich jak algorytm symulowanego wyżarzania pozwala poprawić jakość rozwiązań problemów, dla których jesteśmy w stanie wyznaczyć tylko rozwiązania przybliżone.

Co ciekawe algorytm symulowanego wyżarzania tak samo jak algorytm NEH potrzebuje coraz więcej czasu, aby móc znaleźć lepsze rozwiązanie dla większej instancji problemu.

Algorytm symulowanego Wyżarzania jest algorytmem polegającym na losowych zmianach, co oznacza, że wielokrotne wykorzystanie tego samego algorytmu nawet na tej samej instancji problemu może przynieść różne rezultaty.

W zależności od parametrów z jakim zostało wywołane symulowane wyżarzanie (temperatura, prędkość stygnięcia, ilość iteracji bez zmian potrzebna do zakończenia algorytmu), algorytm wykonuje się przez różną ilość czasu. Otrzymane wyniki są zdecydowanie lepsze, gdy wywołamy algorytm kilkakrotnie z parametrami zmniejszającymi długość pracy algorytmu i zwróceniu najlepszej znalezionej wartości, niż przy jednokrotnym „długim” wywołaniu.

Z regresji liniowej widocznej na wykresach widać, że średnie przyspieszenie uszeregowania zadań spada wraz ze wzrostem wielkości instancji problemu. Z tabelki widać, że tak naprawdę duży spadek widoczny jest dopiero przy największych instancjach problemu, wcześniej wartości są bardzo mocno zmienne.

Średnia wartość przyspieszenia dla wszystkich instancji problemu wyniosła 2,883% przy odchyleniu standardowym 1.847%. Widać tutaj, że algorytm ten daje bardzo zróżnicowane wyniki (średnia jest równa 1,56 wartości sigmy).

Algorytm symulowanego wyżarzania poprawił wyniki w bardzo dużym stopniu i uważamy, że zdecydowanie nadaje się do rozwiązywania tego typu problemów tym bardziej, że nie jest konieczna skomplikowana analiza problemu, a tylko znajomość funkcji celu i stworzenie funkcji mutującej uszeregowanie.