Paulina Pacura, 142179

Jacek Gulij, 136714

Informatyka, Wydział Informatyki

Prowadzący: dr Maciej Machowiak

Optymalizacja Kombinatoryczna – Sprawozdanie z projektu

**Algorytm Szeregowania Zadań P||Cmax**

1. Przedstawienie wybranego zagadnienia

1.1 Cel projektu

Celem pracy było stworzenie programu symulującego problem przydziału określonej listy zadań (procesów) do pewnej liczby procesorów. Opracowano algorytm genetyczny rozwiązujący ten problem, a jego skuteczność porównana została z programami napisanymi przez inne grupy biorące udział w projekcie.

1.2 Omówienie teoretyczne

Problem szeregowania zadań jest jednym z NP-trudnych problemów optymalizacyjnych. Polega ono na przypisaniu kolejnych procesów do pewnej liczby procesorów, wykonujących te procesy. Sformułowanie problemu jest następujące:

„Biorąc pod uwagę zestaw *n* procesów, w których proces *ni* ma długość *l* oraz liczbę procesorów *m*, jaki jest minimalny możliwy czas wymagany do zaplanowania wszystkich zadań na procesorach tak, by żadne z nich nie pokrywały się?”

Dlatego optymalnym szeregowaniem jest taka funkcja *s,* która każdemu procesowi przypisuje jeden z procesorów w ten sposób, że czas zakończenia zadania jest możliwie najkrótszy.

Algorytmy szeregowania podzielić można na statyczne i dynamiczne. Algorytm jest statyczny, jeśli decyzje dotyczące przydziału procesorów zapadają przed uruchomieniem programu, natomiast dynamiczny, jeśli w czasie wykonywania.

1. Wybrana metoda rozwiązania problemu
   1. Omówienie różnych typów podejść do problemu szeregowania

Jednym z możliwych podejść do problemu jest szeregowanie listowe. Polega ono na ustaleniu kolejności zadań na liście; następnie za każdym razem, gdy zwalnia się któryś z procesorów, wybierane jest pierwsze wolne zadanie i przypisywane do owego procesora. Przykładową modyfikacją takiego podejścia jest wstępne posortowanie zadań na liście według nierosnących czasów wykonania pj.

Innym przykładowym algorytmem rozwiązującym problem szeregowania procesów jest **algorytm McNaughtona.** Jego głównym założeniem jest wzór pozwalający wyliczenie optymalnej długości:

Na podstawie tego, należy szeregować kolejno zadania aż do osiągnięcia . Po tym należy przerwać zadanie i w przypadku, jeśli się nie zakonczyło, kontynuować je na następnym procesorze od chwili zerowej.

Do realizacji zadania projektowego zdecydowaliśmy się wykorzystać algorytm genetyczny, opisany w dalszych podpunktach.

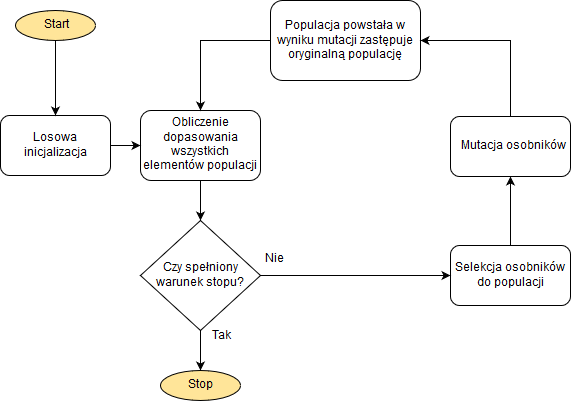
* 1. Początek projektu – wersja zachłanna

Algorytm zachłanny sortuje malejąco procesy i przyporządkowuje je po kolei do aktualnie najmniej zapełnionego czasowo procesora.

* 1. Przedstawienie założeń algorytmu genetycznego

Algorytmy genetyczne to algorytmy wzorujące się na mechanizmie ewolucji: przetrwanie osobników najlepiej dostosowanych. Korzystają one z określeń zapożyczonych z genetyki: populacja to zbiór osobników o określonej liczebności, osobnikami nazywa się rozwiązania, zbiór informacji przypisany do osobnika to genotyp, a fenotyp to zbiór cech podlegających ocenie funkcji przystosowania.

Mechanizm działania algotytmów genetycznych polega na przeszukiwaniu przestrzeni alternatywnych rozwiązań w celu odnalezienia najlepszych rezultatów. Istotnym elementem jest losowość – dzięki losowym zmianom podczas mutacji do populacji wprowadzana jest różnorodność. Początkowo populacja inicjowana jest losowo; następuje dalej ocena przystosowania chromosomów. Miarę przystosowania danego osobnika stanowi funkcja przystosowania; pozwala ona na wybór najlepiej przystosowanych osobników (czyli o największej wartości tej funkcji). Po dokonaniu oceny przystosowania chromosomów w populacji, następuje sprawdzenie warunku zatrzymania (np. czy otrzymano już żądaną wartość? Czy dalsze działanie nie poprawia już uzyskanej wartości?) oraz selekcja chromosomów polegająca na wybraniu tych osobników, które będą brały udział w tworzeniu następnej generacji, które następnie poddawane są działaniu operatorów mutacji oraz krzyżowaniu, a w otrzymane w wyniku chromosomy wchodzą w skład nowej generacji. Celem jest odnalezienie chromosomu o największej wartości funkcji przystosowania.



Rysunek 1 Schemat blokowy przedstawiajacy zasadę działania algorytmu

* 1. Szczegóły implementacyjne algorytmu genetycznego

W wybranym podejściu założono, że każdy procesor wykonać może każde zadanie (brak ograniczeń pod względem przydzielania zadań do procesora). Chromosom stanowi uporządkowanie procesów w pewnej kolejności. Funkcja przystosowania, dla której szeregujemy chromosomy malejąco, jest algorytmem skrajnie zachłannym, który dla zadanej maksymalnej długości pracy procesora bierze po jednym procesie (według kolejności ustalonej w chromosomie) i przyporządkowuje go do aktualnie wybranego procesora. W przypadku zapełnienia czasu algorytm zamyka aktualny proces i zaczyna zapełniać kolejny. Funkcja ta zwraca łączną sumę czasów nieprzyporządkowanych procesów, czyli w przypadku zwrócenia wartości 0 wybrany chromosom stanowi przykład rozwiązania z podanym maksymalnym czasem długości pracy. Za każdym razem, gdy zostanie uzyskana wartość 0, algorytm zmniejsza zadaną maksymalną długość procesora i cykl trwa dalej. Do następnej populacji przechodzi aktualny zwycięzca oraz z rozkładem liniowym malejącym są wybierane mutacje chromosomów z aktualnej populacji. Mutacja chromosomu to przestawienie w kolejności dwóch procesów losową ilość razy.

1. Wyniki działania alorytmu dla poszczególnych instancji
   1. Tabela porównująca instancje wykorzystywane w ramach zajęć

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instancja** | **Najlepszy wynik uzyskany w grupie** | **Wynik z zadanym maksymalnym czasem** | **Wynik bez zadanego maksymalnego czasu** | **Generacje na sekundę** |
| m25 | **3459** | **3459** w 48500 generacji | 3466 | 150 |
| m50 | 150 | 151 w 15500 generacji | 153 | 250 |
| m50n1000 | 9764 | 9793 w 3400 generacji | 9797 | 30 |
| m50n200 | **979** | **979** w 41000 generacji | 986 | 150 |
| m10n200 | 10999 | 11000 w 5000 generacji | 11001 | 150 |

* 1. Parametry algorytmu dla danych instancji

mutationRate = 0.001

populationSize = 1000

Zadany maksymalny czas pracy procesora był wybierany na podstawie najlepszego wyniku w grupie.

Wyniki bez zadanego czasu ograniczone zostały do minuty.

* 1. Wnioski

1. Zadanie projektowe polegało na stopniowym opracowaniu jak najskuteczniejszego algorytmu rozwiązującego dany problem, dzięki czemu w trakcie pracy możliwym było obserwowanie wzrostu (lub spadków) skuteczności wybranej metody.
2. Algorytm zachłanny wykonuje się znacznie szybciej, ale wyniki są znacznie oddalone od optymalnych.
3. Pomimo optymalizacji algorytmu wokoło ustalonego maksymalnego czasu w praktyce okazuje się, że dynamiczne dostosowywanie tej granicy nie ma znaczącego wpływu na prędkość znajdywania rozwiązań.
4. Znaczącą poprawę w wydajności algorytmu stanowiło przyspieszenie funkcji przystosowania oraz mutacji osobników – jeden cykl tworzenia generacji wykonuje się w czasie O(np+kp) gdzie p jest rozmiarem populacji, a k współczynnikiem mutacji.