

Równoległy algorytm mrówkowy

dla jednomaszynowego problemu szeregowania zadań

Piotr Gródek Tomasz Pawlak

Instytut Informatyki
Uniwersytet Wrocławski

23 maja 2012

Opis problemu

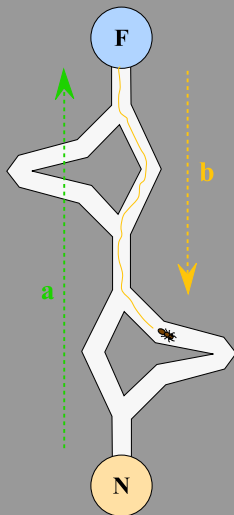
- dany jest zbiór n ponumerowanych zadań $N = \{1, 2, \dots, n\}$, które należy wykonać bez zatrzymywania na jednej maszynie,
- maszyna wykonuje co najwyżej jedno zadanie,
- każde zadanie t_i jest opisane przez wartości p_i, w_i, d_i :
 - p_i - czas wykonania zadania,
 - w_i - waga funkcji kosztów,
 - d_i - termin wykonania zadania (linia krytyczna).
- oznaczając przez C_i termin zakończenia zadania i wprowadzamy opóźnienie zdefiniowane jako:

$$T_i = \max\{0, C_i - d_i\}$$

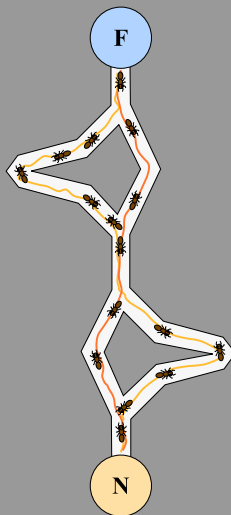
Cel:

Szukamy permutacji zadań π , która minimalizuje wartość

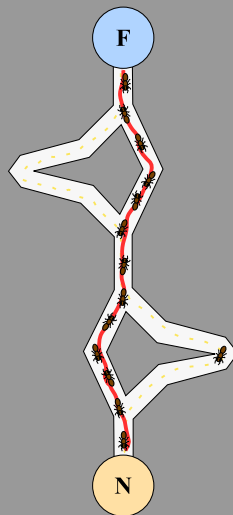
$$Y(\pi) = \sum w_{\pi(i)} T_{\pi(i)}.$$



1



2



3

Najpierw kilka obserwacji:

- możliwe drogi to wszystkie permutacje zadań
- liczba feromonu rośnie wraz ze spadkiem wartości $Y(\pi)$
- mrówka wydaje się idealnym kandydatem na osobny wątek

Z tych obserwacji wynika, że:

- mrówka musi pamiętać które zadania musi jeszcze wykonać - będziemy pamiętać listę zadań do wykonania,
- będziemy mieć dwuwymiarową tablicę feromonów, gdzie pod indeksem $[i, j]$ będzie liczba feromonu związane z wyborem j -tego zadania na miejscu i , przy czym jego ilość zostawiona przez mrówkę jest postaci $\frac{1}{Y(\pi)}$,
- jeśli każda mrówka byłaby osobnym wątkiem to rośnie koszt synchronizacji oraz mamy mniejszy rozrzut badanych ścieżek czyli rośnie prawdopodobieństwo wpadnięcia w lokalne minimum.

Nasz algorytm polega na tym, że:

- tworzymy równoległe **mrowiska**, które w każdej iteracji wysyłają zadaną liczbę *niezależnych* mrówek (które też są osobnymi wątkami),
- każde mrowisko pamięta najlepsze osiągnięte wyniki,
- raz na jakiś czas tworzony jest na tej podstawie ranking mrowisk oraz zostają wymieszane tablice feromonów (najlepsze z najgorszymi).

Dodatkowo chcemy mieć graficzny interfejs użytkownika i graficzny podgląd rozkładu feromonów co wymaga uruchomienia kontrolera mrowisk jako osobny wątek i dodatkową synchronizację.

Program został napisany w języku C# przy użyciu mechanizmów programowania wielowątkowego z platformy .NET 4.0.

Algorytm

- ① Stwórz n mrowisk jako osobne wątki
- ② Dopóki nie przerwano obliczeń:
 - ① W każdym mrowisku powtórz i razy:
 - ① Wygeneruj k mrówek
 - ② Dla każdej mrówki znajdź ścieżkę na podstawie intensywności feromonu oraz ew. heurystyki
 - ③ Uaktualnij feromony
 - ② Posortuj mrowiska względem znalezionych ścieżek
 - ③ Zapamiętaj najlepszą ścieżkę
 - ④ Wymieszaj feromony z zadanymi współczynnikami tak, że najlepszy miesza się z najgorszym itd.
- ③ Zwróć najlepszą ścieżkę.

Czas na prezentację programu :)

Dane testowe pochodzą ze [strony J E Beasley](#).

Zaprezentujemy teraz wyniki testów dla przykładowej grupy instancji naszego problemu.

Poza wynikami obliczeń porównaliśmy również wydajność dla różnych konfiguracji programu.

Dziękujemy za uwagę.