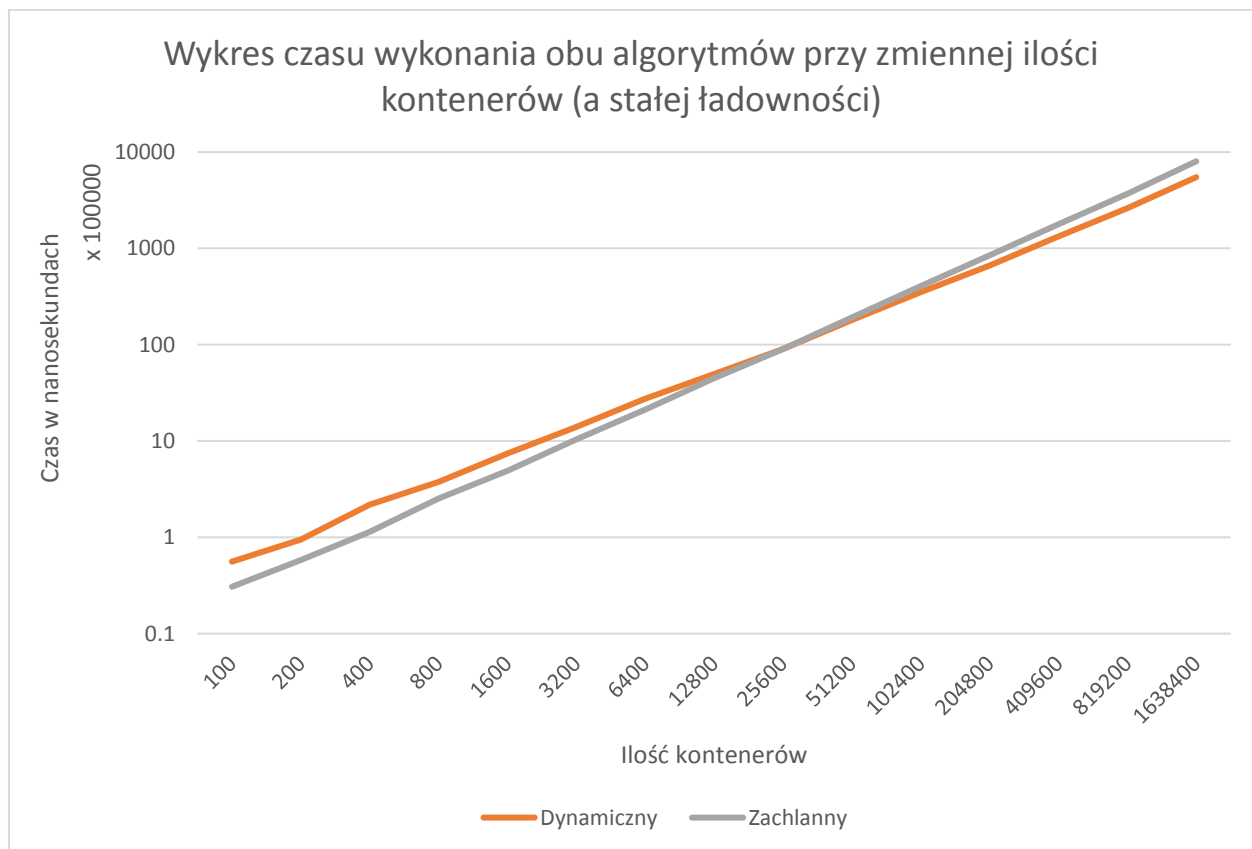


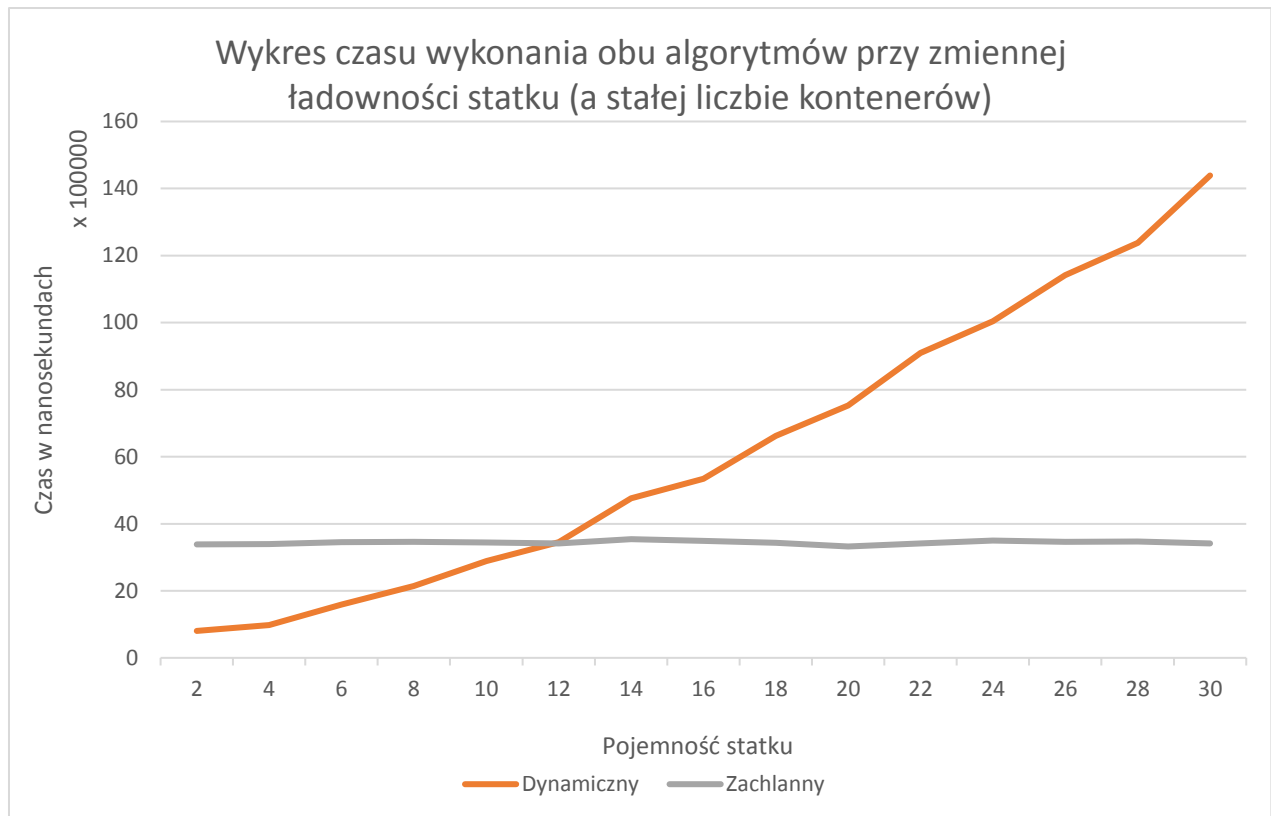
Algorytmy i Struktury Danych - Sprawozdanie 4: Problem plecakowy

Problem plecakowy jest problemem optymalizacyjnym, polegającym na wyborze podzbioru ze zbioru elementów o podanej wadze i wartości, tak by ich sumaryczna wartość była jak największa, a suma wag była nie większa od danej pojemności plecaka. Problem plecakowy ma szerokie zastosowanie w przeróżnych dziedzinach i możemy go tak naprawdę zaaplikować do niezliczonej ilości codziennych sytuacji. Może być przedstawiany np. jako problem złodzieja rabującego sklep, a w tym zadaniu dotyczył statku ładowanego kontenerami o różnych wagach i wartościach.

Do rozwiązania problemu zaprogramowaliśmy dwa algorytmy, które poddamy analizie i porównaniu:

- Algorytm programowania dynamicznego – polegający na stworzeniu tabeli rozwiązań danego problemu metodą dynamiczną, to znaczy w oparciu o rozwiązanie problemu mniejszego instancją lecz zawierającego się w głównym.
- Algorytm zachłanny – polegający na sortowaniu zbioru elementów na podstawie stosunków ich wartości do wagi w kolejności malejącej, a następnie wstawianiu ich w tej kolejności z omijaniem tych elementów, których waga jest większa niż aktualnie dostępna pojemność.





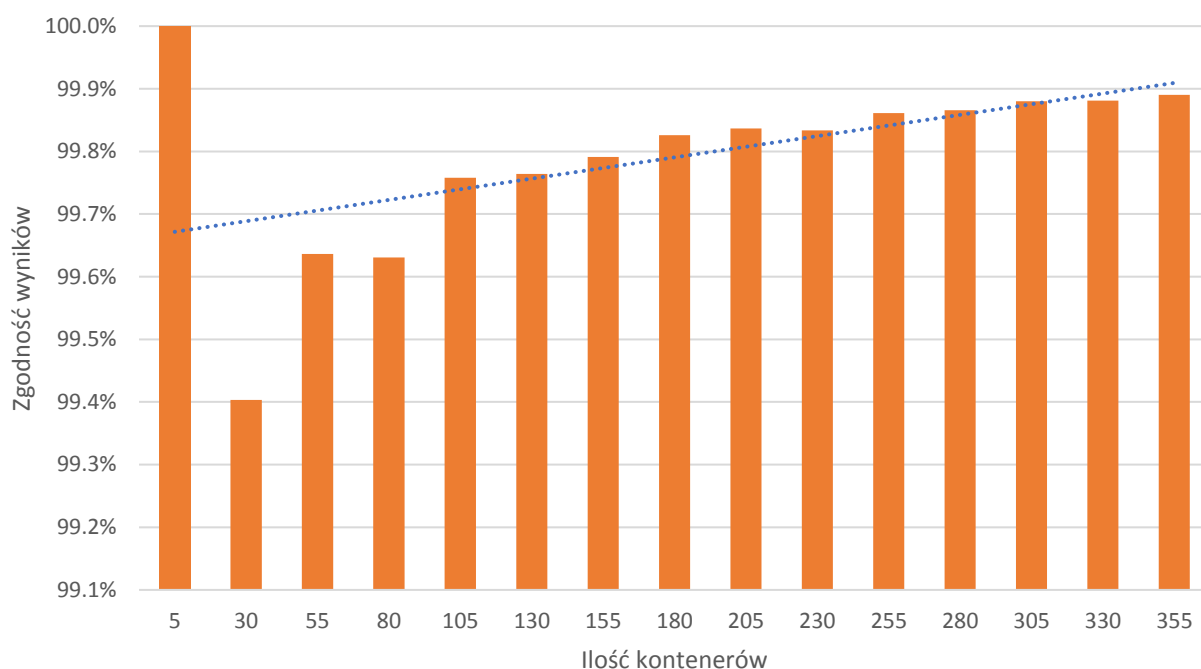
Złożoność obliczeniowa algorytmu dynamicznego wynosi $O(nb)$, ponieważ najbardziej złożoną operacją jest iteracja po tablicy $n \times b$ elementowej, gdzie n jest ilością kontenerów, a b ładownością. Złożoność algorytmu zachłannego wynosi $O(n \log n)$, z powodu zawartego weń sortowania, którego złożoność wynosi właśnie tyle. Pierwszy wykres pokazuje bardzo ciekawą różnicę między tymi algorytmami. Obie funkcje są zależne liniowo od n oraz drugiego współczynnika. W przypadku algorytmu dynamicznego jest to stałe b , które mimo swej wielkości zostaje w pewnym momencie przewyższone przez wolno, lecz stale rosnące $\log n$. Perfekcyjnie widać to na wykresie, gdzie z początku szybsza jest funkcja zachłanna, jednak wraz z ogromnym wzrostem liczby elementów, $\log n$ zaczyna powoli prześcigać b , w efekcie na prowadzenie wychodzi funkcja dynamiczna.

To co natomiast okazało się w pierwszym badaniu zaletą algorytmu dynamicznego, tj. zależność od dwóch zmiennych w zamian za słabsze związanie z n , jest zgubne dla niego w drugim badaniu, czyli przy zmiennej ładowności, a stałej liczbie elementów. W takich warunkach czas wykonywania algorytmu zachłannego jest teoretycznie stały, ponieważ funkcja ograniczająca złożoność tego algorytmu jest zależna tylko i wyłącznie od ilości elementów. Zupełnie inaczej sprawa wygląda z algorytmem dynamicznym, gdzie widoczny na wykresie liniowy wzrost czasu wykonywania jest powiązany z zależnością funkcji O tego algorytmu od ładowności.

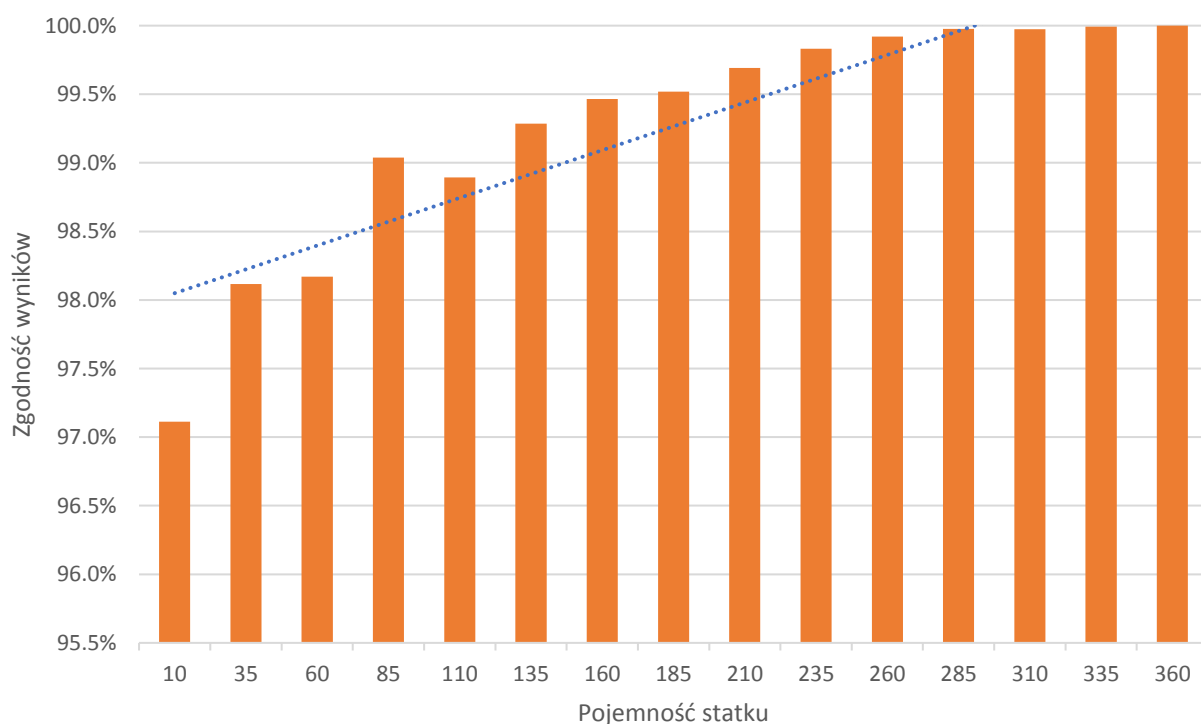
Złożoność pamięciowa zdecydowanie wypada na korzyść algorytmu zachłannego, dla którego wynosi ona $2n$, podczas gdy w dynamicznym jest równa nb . Algorytm zachłanny posiada jednak jedną znaczącą wadę, która może go dyskwalifikować w wielu przypadkach, mianowicie nie gwarantuje optymalnego wyniku.

Algorytm dynamiczny jest dokładny, jednak jeśli nie zależy nam na idealnym rozwiązaniu to dużo lepszym wyborem może okazać się algorytm zachłanny, w szczególności jeśli instancja jest określona dla dużej ładowności. Pozwoli nam to oszczędzić dużo czasu oraz pamięci.

Wykres jakości wyników algorytmu zachłannego w porównaniu z optymalnymi rozwiązaniami dla zmiennej ilości kontenerów (a stałej ładowności)



Wykres jakości wyników algorytmu zachłannego w porównaniu z optymalnymi rozwiązaniami dla zmiennej pojemności statku (a stałej ilości kontenerów)



Jak już wyżej wspomnieliśmy, algorytm zachłanny nie gwarantuje najoptymalniejszego rozwiązania problemu. W powyższych testach sprawdziliśmy jak bardzo średnio odbiegają jego rezultaty dla zmiennych ilości kontenerów, a stałej pojemności oraz odwrotnie, uśredniając wyniki 100 pomiarów dla każdej instancji.

Pierwszy wykres pokazuje pewien wzrost dokładności wyników wraz ze wzrostem ilości kontenerów przy zachowaniu stałej pojemności. Przy przeanalizowaniu poszczególnych wyników można wyciągnąć wnioski, że ma to przyczynę w tym, iż większa ilość kontenerów skutkuje ogólnie lepszymi wynikami. Wynika to z tego, że gdy mamy więcej elementów do wyboru, statystycznie więcej jest także elementów bardzo opłacalnych (także tych bardzo lekkich), co przy zachowanej takiej samej pojemności gwarantuje lepszy, czyli większy wynik. Większe rezultaty osiągane przez oba algorytmy skutkują mniejszą względną różnicą. Jakość rozwiązań będzie rosła więc wraz ze wzrostem ilości elementów, aczkolwiek będzie to wzrost coraz bardziej subtelny, który nie dojdzie do równych 100%. Skrajnym przypadkiem, który zdecydowaliśmy się zaprezentować na wykresie, jest na tyle niewielka ilość elementów przy sporej pojemności i wagach elementów w pewnym zakresie (nie przekraczających pewnego maksimum), że wszystkie elementy wchodzą do plecaka. Dla takich wyjątkowych przypadków mamy pewność co do 100% dokładności algorytmu zachłannego. W naszych testach zachodził on dla pojemności kilku (np. 5) elementów, ponieważ przyjmowaliśmy pojemność statku równą 200, a wagi przedmiotów z zakresu 1-30 (czyli maksymalna możliwa waga wszystkich przedmiotów wynosi w takich warunkach 150).

Drugi wykres również ukazuje wzrost dokładności, tym razem przy wzroście pojemności statku a zachowaniu stałej ilości elementów. Pomimo że przypadek jest odwrotny, to wzrost ten ma bardzo podobną bezpośrednią przyczynę – większa pojemność powoduje, że możemy zmieścić więcej elementów, co gwarantuje statystycznie znacznie lepsze, większe rezultaty, a więc mniejszą względną różnicę między wynikami algorytmu zachłannego a tymi optymalnymi. Dla bardzo małych pojemności (np. 10), tak jak często będą wychodziły idealne rezultaty, tak każde odchylenie o jeden, będzie skutkowało obniżeniem jakości rozwiązania o 10%, co będzie znacząco obniżało średnią dokładność, co widać na wykresie. Skoro natomiast wraz ze wzrostem pojemności do plecaka będzie wchodziło coraz więcej przedmiotów, to tutaj również może zachodzić, celowo przez nas rozważony, skrajny przypadek dla którego do plecaka zawsze wchodzić będą wszystkie elementy. Z racji że będzie on zachodził dla wyjątkowo wielkich pojemności, 100% dokładność widzimy na końcu drugiego wykresu. Z wyżej wymienionych powodów, przy założeniu zachowania pewnych ram wagowych przedmiotów, średnia jakość rozwiązań będzie zawsze od pewnej dużej liczby wyznaczającej pojemność wynosić dokładnie 100% (a wcześniej zapakowywanie wszystkich elementów będzie zachodzić dla coraz to większej liczby pojedynczych przypadków).

Problem plecakowy to dosyć ciekawy problem z punktu widzenia przynależności klasowej. Należy do klasy NP, a ściśle jest to problem NP zupełny bądź NP trudny, w zależności od tego czy przedstawimy go w formie decyzyjnej czy optymalizacyjnej. Nie istnieje zatem algorytm wielomianowy gwarantujący optymalne rozwiązanie w każdym przypadku, ponieważ algorytm programowania dynamicznego jest uznawany jedynie za algorytm pseudowielomianowy, tzn. jego czas działania jest ograniczony od wielkości wejścia i maksymalnej wartości liczbowej, którą w tym wypadku jest pojemność b .