

ALHE

Dokumentacja

Dawid Brzozowski
Konrad Opaliński

Temat:

Santa's Uncertain Bags

1. Ogólny opis zadania:

Pomóż św. Mikołajowi zapakować przygotowane prezenty w odpowiednie worki. Do dyspozycji mamy 1000 worków i 9 typów prezentów. Wszystko byłoby proste, gdyby nie fakt, że ręcznie robione przez elfy prezenty są zupełnie różne. Każdy typ prezentu tylko w przybliżeniu definiuje jego wagę, a pomyłki mogą być bardzo kosztowne. Nad wszystkim czuwa ministerstwo pracy, które konfiskuje wszystkie worki których waga przewyższa dopuszczalne 50 funtów.

2. Sposoby obliczania wag prezentów:

Każdy prezent ma stałą wagę, ale poszczególne wagi są nieznane. Wagi dla każdego rodzaju prezentu nie są identyczne, ponieważ elfy wykonują je w wielu rodzajach i rozmiarach. Chociaż wagi zostały usunięte z bazy danych, elfy nadal mają plany każdej zabawki. Po kilku skomplikowanych integracjach objętości, elfom udało się dać Mikołajowi rozkład prawdopodobieństwa dla wagi każdego typu zabawki. Aby symulować wagę pojedynczego prezentu w funtach, wymyślili następujące parametry rozkładu:

- Horse
 - Parametry rozkładu: $\max(0, \text{np.random.normal}(5, 2, 1)[0])$
 - Ilość: 1000
- Ball
 - Parametry rozkładu: $\max(0, 1 + \text{np.random.normal}(1, 0.3, 1)[0])$
 - Ilość: 1100
- Bike
 - Parametry rozkładu: $\max(0, \text{np.random.normal}(20, 10, 1)[0])$
 - Ilość: 500
- Train
 - Parametry rozkładu: $\max(0, \text{np.random.normal}(10, 5, 1)[0])$
 - Ilość: 1000

- Coal
 - Parametry rozkładu: $47 * \text{np.random.beta}(0.5, 0.5, 1)[0]$
 - Ilość: 166
- Book
 - Parametry rozkładu: $\text{np.random.chisquare}(2, 1)[0]$
 - Ilość: 1200
- Doll
 - Parametry rozkładu: $\text{np.random.gamma}(5, 1, 1)[0]$
 - Ilość: 1000
- Block
 - Parametry rozkładu: $\text{np.random.triangular}(5, 10, 20, 1)[0]$
 - Ilość: 1000
- Gloves
 - Parametry rozkładu: $3.0 + \text{np.random.rand}(1)[0]$ if $\text{np.random.rand}(1) < 0.3$ else $\text{np.random.rand}(1)[0]$
 - Ilość: 200

3. Sposób oceniania:

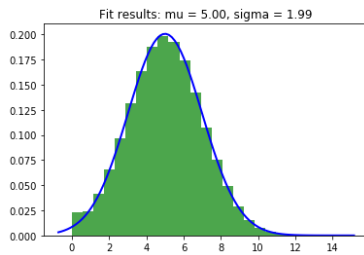
Zgłoszenia są oceniane na podstawie całkowitej wagi, jaką uda się zmieścić w 1000 torbach Mikołaja. Zasady dotyczące pakowania prezentów są następujące:

- Przepelnienie worka powyżej limitu 50 funtów spowoduje, że cały worek nie będzie się liczył
- Żaden prezent nie może być użyty więcej niż raz (chodzi tutaj o prezenty o tym samym ID)
- Każda torba musi mieć 3 lub więcej prezentów (nie typów prezentów)

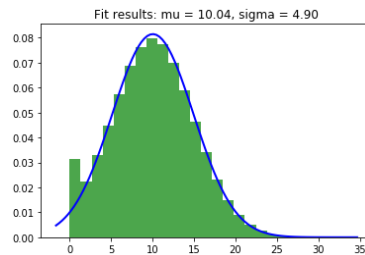
Graficzna interpretacja rozkładów wag prezentów:

Po zapoznaniu się z zadaniem oraz z kryterium oceniania rozwiązania problemu, postanowiliśmy przyjrzeć się dokładniej podanym rozkładom wag prezentów. Sporządziliśmy histogramy podanych rozkładów oraz dopasowaliśmy do nich rozkłady normalne.

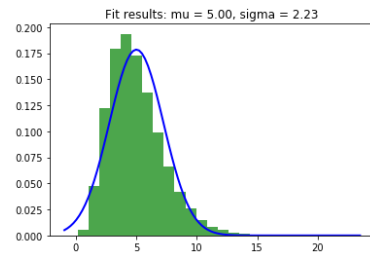
Horse



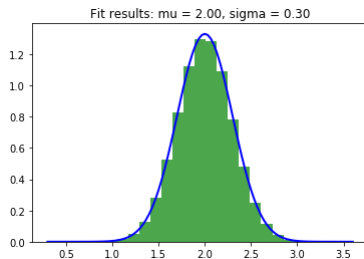
Train



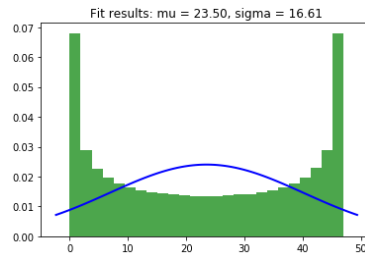
Doll



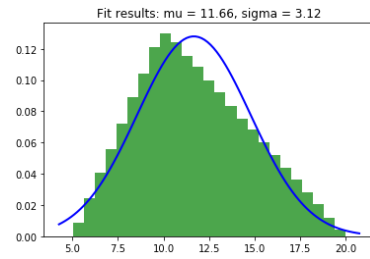
Ball



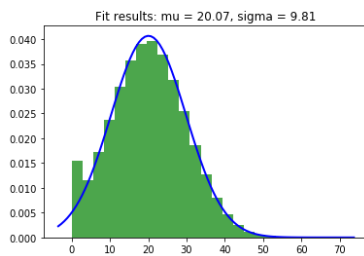
Coal



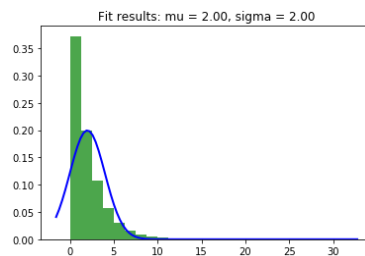
Block



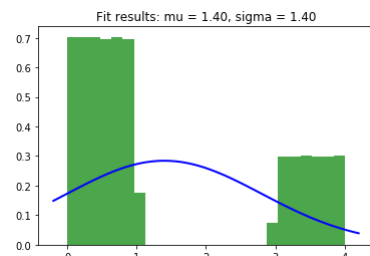
Bike



Book



Gloves



Definicja funkcji celu oraz sposobu mierzenia jakości rozwiązania:

Jakość rozwiązania mierzymy poprzez sumę mas worków, o masie nieprzekraczającej 50 funtów.

Im większa sumaryczna masa, tym lepsze rozwiązanie. Platforma Kaggle oferuje "obiektywne" wyliczenie jakości i z tej ewaluacji będziemy korzystać do obliczenia jakości zoptymalizowanego rozwiązania.

Podczas optymalizacji parametrów, na bieżąco będziemy ewaluować nasze rozwiązanie poprzez wylosowanie mas przedmiotów z podanych rozkładów i wyliczenie wartości funkcji celu.

Rozwiązanie:

W ramach rozwiązywania problemu skorzystaliśmy z propozycji zamieszczonych w dokumentacji wstępnej - algorytmu plecakowego przyjmującego ilość prezentów oraz ich wartości oczekiwane i odchylenie standardowe do ewaluacji sposobu zapakowania.

1. Moduł odpowiedzialny za pakowanie prezentów (algorytm plecakowy)

Algorytm generuje kombinacje typów prezentów oraz ich ilości jakie można zapakować do danego plecaka tak aby nie przekroczyć maksymalnej dopuszczalnej jego wagi. Uwzględnia on też aktualną ilość dostępnych typów prezentów. Algorytm rozpoczyna pakowanie prezentów od najcięższego do najlżejszego. W przypadku, gdy maksymalną pojemność worka wypełniły mniej niż 3 prezenty, wypakowujemy ostatni z nich oraz próbujemy dopakować następny typ prezentu z kolejki (w przypadku, gdy algorytm dojdzie do końca kolejki i nie znajdzie odpowiedniego prezentu, wypakowuje ostatnio dodany prezent oraz zaczyna pakować od nowa, ale z zaktualizowaną listą dostępnych typów prezentów - bez typu, dla którego nie udało się znaleźć co najmniej 2 innych typów prezentów tak aby spełnić wymogi zadania). Proces ten wykonywany jest tyle razy ile jest dostępnych plecaków, w tym przypadku jest to 1000 razy. Ostatecznie zwracana jest lista zapakowanych plecaków, która następnie trafia do ewaluacji.

2. Moduł odpowiedzialny za dobieranie parametrów.

- a. W początkowej implementacji, naszymi jedynymi parametrami były wagi typów przedmiotów oraz maksymalna pojemność plecaka. Stosowaliśmy zasadę taką, że początkowo braliśmy wartości oczekiwane z rozkładów oraz daną pojemność i za pomocą algorytmu plecakowego generowaliśmy wynik. Manipulując pojemnością, mogliśmy dostosować zapakowanie w ten sposób, aby plecak rzadko był "przeładowany" (nie przekroczył 50 funtów), ale jednocześnie miał dużą masę, co bezpośrednio przekłada się na wynik ewaluacji.
- b. W kolejnym etapie stwierdziliśmy, że warto byłoby uwzględnić nie tylko oczekiwaną masę prezentu lecz także jego odchylenie standardowe. Zdecydowaliśmy, że uwzględnimy to przy pakowaniu prezentów w ten sposób, że oprócz manipulowania pojemnością plecaka oraz wagami typów przedmiotów, manewrować będziemy także wpływem odchylenia standardowego na wypełnienie plecaka pewnym współczynnikiem. Od tego momentu, dodając przedmiot do plecaka uwzględniamy również jego odchylenie przemnożone o podany współczynnik.
- c. Do optymalizacji wag, zastosowaliśmy odmianę algorytmu 1+1. Polega ona na tym, że generujemy wagi typów przedmiotów dodając do ich wartości oczekiwaną losową wartość z określonego zakresu (nieodległą od obecnych), który zmniejsza się wraz z każdą iteracją. Z pomocą tych wag uruchamiamy algorytm pakujący prezenty, a następnie ewaluujemy wynik pakowania. W przypadku, gdy otrzymamy rezultat przewyższający poprzedni najlepszy wynik - aktualizujemy wagi oraz obecny najlepszy wynik.

3. Moduł odpowiedzialny za ewaluację zapakowania

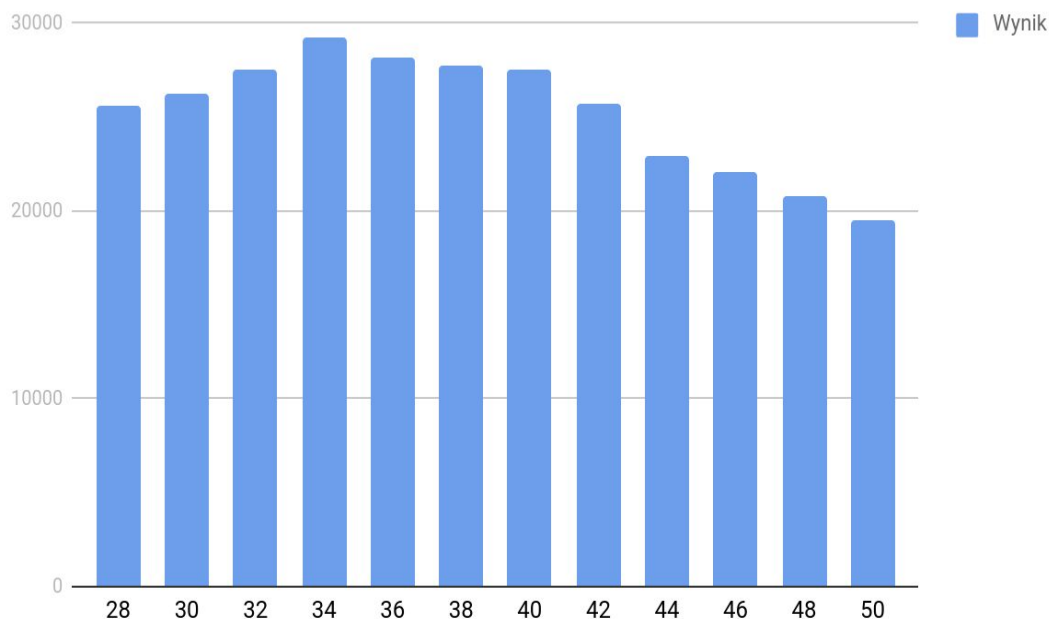
- a. Ewaluacja wyniku zbliżona jest do faktycznej ewaluacji przeprowadzanej na stronie konkursowej Kaggle. Jako argument, przyjmuje zapakowaną listę plecaków. Polega na tym, że generujemy wagi typów prezentów z rozkładów określonych w treści zadania, a następnie obliczamy wynik w ten sposób, że dla każdego worka obliczamy jego masę - jeżeli nie przekracza ona 50 funtów, dodajemy do całkowitej masy zapakowania. Aby uniknąć rozbieżnych wyników dla tego samego zbioru związanego z losowością zadania, tą operację przeprowadzamy kilkadziesiąt razy a następnie wynik uśredniamy.
- b. Ewaluacja ta, różni się od tej z konkursu Kaggle, ponieważ Kaggle przechowuje zapisaną masę dla każdego prezentu z osobna i korzystając z nich sumuje masy plecaków. My tych mas niestety nie znamy i aby uniknąć rozbieżności w wynikach spowodowanych nadmierną losowością zastosowaliśmy powyższe "uproszczenie" - które daje bardzo zbliżone wyniki.

4. Wyniki oraz wnioski

A. Tabela zależności dla początkowego rozwiązania biorącego pod uwagę oczekiwane wagi prezentów oraz manipulującego maksymalną pojemnością plecaka.

Pojemność plecaka	Wynik
28	25630
30	26225
32	27497
34	29277
36	28131
38	27737
40	27557
42	25674
44	22961
46	22123
48	20825
50	19483

Wynik w zależności tylko od pojemności plecaka

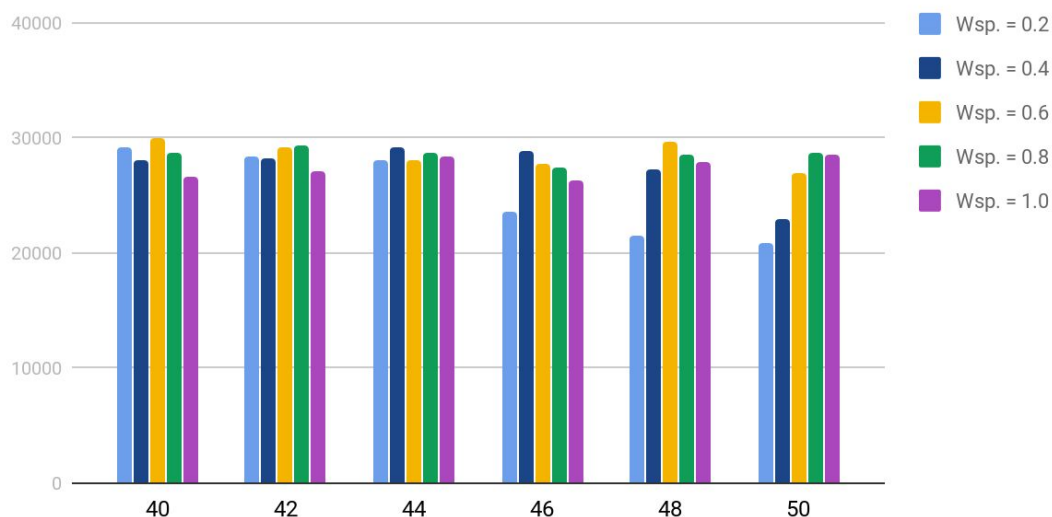


Wnioski: Wyniki przedstawione powyżej zgadzają się z naszymi oczekiwaniami. Widać, że dla pojemności plecaka równej maksymalnej pojemności (50 funtów) wynik jest bardzo niski. Jest to spowodowane tym, że znaczna część plecaków zostaje odrzucona z powodu przepełnienia. Wraz z malejącą pojemnością plecaka, wynik początkowo rośnie aż do momentu, gdy pojemność ta znacznie ogranicza ilość zapakowanych prezentów (wtedy wynik z powrotem maleje).

B. Tabela zależności dla rozwiązania biorącego pod uwagę oczekiwane wagi prezentów oraz ich odchylenia standardowe. Rozwiązanie dodatkowo manipuluje maksymalną pojemnością plecaka oraz współczynnikiem, przez który zostaje pomnożone odchylenie standardowe.

Pojemność	Wynik				
	Wsp. = 0.2	Wsp. = 0.4	Wsp. = 0.6	Wsp. = 0.8	Wsp. = 1.0
40	29116	28063	30020	28767	26693
42	28363	28158	29169	29383	27074
44	28097	29143	28094	28697	28379
46	23637	28916	27754	27507	26372
48	21574	27220	29652	28610	27901
50	20867	22913	26949	28744	28607

Wynik w zależności od pojemności plecaka i współczynnika odchylenia standardowego



Wnioski: W tym przypadku można dostrzec, że wynik dla podanego współczynnika odchylenia standardowego za jednym razem rośnie wraz ze wzrostem pakowności plecaka, a za drugim maleje. Tendencję malejącego wyniku można zauważyć np. dla współczynnika równego 0.2. Jest to spowodowane tym, że na początku mamy zapas pakowności wynikający z wolnych 8 funtów (dla pakowności równej 42 funty). W sytuacji, gdy takiego zapasu nie mamy, wówczas okazuje się, że współczynnik ten jest zbyt mały i znaczna część plecaków zostaje odrzucona. Przeciwną sytuację można zaobserwować dla współczynnika równego 1.0. W tym przypadku wynik rośnie wraz ze wzrostem pakowności (nie jest to co prawda tak mocny wzrost, co spadek w poprzedniej sytuacji). Jest to spowodowane tym, że na początku przesacowujemy ilość wolnego miejsca w plecaku (dla pakowności 42 funtów mamy 8 funtów wolnego miejsca oraz duży współczynnik, który dodatkowo zmniejsza maksymalną ilość prezentów, które można zapakować). Dla maksymalnej pojemności plecaka równej 50 funtów okazuje się, że współczynnik ten już dość dobrze sobie radzi. Widać również, że wyniki średnio są wyższe niż dla poprzedniej wersji. Wynika to z tego, że uwzględniając odchylenie standardowe dla poszczególnych rodzajów prezentów, lepiej dopasowujemy zapas, który potrzebny jest do tego aby bezpiecznie zapakować plecak.

C. Rozwiązanie z poprzedniego podpunktu po zastosowaniu optymalizacji 1+1 dla wartości oczekiwanych typów prezentów.

Pojemność	Wynik				
	Wsp. = 0.2	Wsp. = 0.4	Wsp. = 0.6	Wsp. = 0.8	Wsp. = 1.0
40	29116 31982	28063 31949	30020 31259	28767 31890	26693 30196

42	28363 30994	28158 29969	29169 30918	29383 30915	27074 32452
44	28097 29769	29143 34004	28094 31568	28697 28939	28379 31577
46	23637 31794	28916 31407	27754 32617	27507 31664	26372 31647
48	21574 31429	27220 32174	29652 31876	28610 29380	27901 31662
50	20867 32827	22913 31832	26949 31323	28744 28915	28607 29486

* Wartości otrzymane w tym punkcie zostały zapisane czcionką pogrubioną, pozostałe wartości są wartościami z poprzedniego punktu i zostały podane w celu porównania.

** Zapakowanie zaznaczone na szaro zostało sprawdzone w serwisie Kaggle oraz otrzymaliśmy taki wynik (jest to najwyższy wynik jaki udało nam się otrzymać):

44wsp0.4.csv
just now by opalkonrad
44, wsp=0.4

33740.50900 33740.50900



Wnioski: W tym przypadku dzięki skorzystaniu z optymalizacji 1+1 udało nam się polepszyć wynik w każdej sytuacji. Ciekawą rzeczą, którą można zauważyć jest to, że optymalizacja prawie w każdym z przypadków osiągała wynik powyżej 30000 punktów. Jest to o tyle interesujące, że nawet dla pakowania o współczynniku równym 0.2 oraz pakowności 50 funtów, algorytm umiał na tyle dopasować wagi przedmiotów, że końcowy wynik był na bardzo wysokim poziomie. Różnica punktów w tym przypadku pomiędzy algorytmem bez optymalizacji, a tym z optymalizacją 1+1 wynosi ok. 12000 punktów! Warto również zwrócić uwagę na to, że pomimo iż Kaggle ma na stałe przypisane wagi poszczególnych prezentów, a w naszym ewaluatorze wagi te są za każdym razem inne (stąd kilkudziesięciokrotne wywołanie obliczenia wyniku) udało się otrzymać bardzo podobne wyniki (34004, a 33740 punktów).

Podsumowanie

Każdy kolejny etap projektu wprowadzał poprawki do już istniejącego algorytmu, które miały bezpośredni wpływ na wynik ewaluacji. Elementem problematycznym był brak możliwości testowania naszego optymalizatora 1+1 idealnej ewaluacji. Przybliżenie jakie zastosowaliśmy w ewaluatorze pozwoliło na znaczne poprawienie rezultatów niezależnie od konkretnych wag, które przypisane są na platformie Kaggle. Skutkowałoby to dobrym rezultatem na nowych, nie widzianych wcześniej danych.