

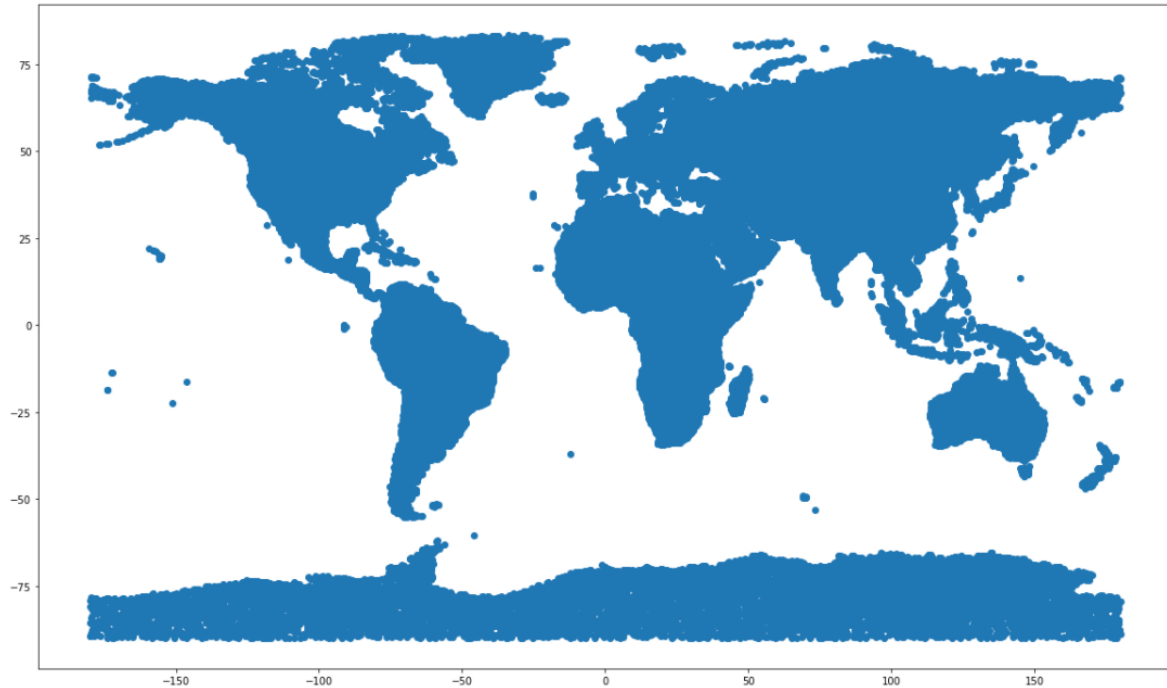
Projekt POP

Santa's Stolen Sleigh

Oskar Bartosz, Ewa Roszczyk

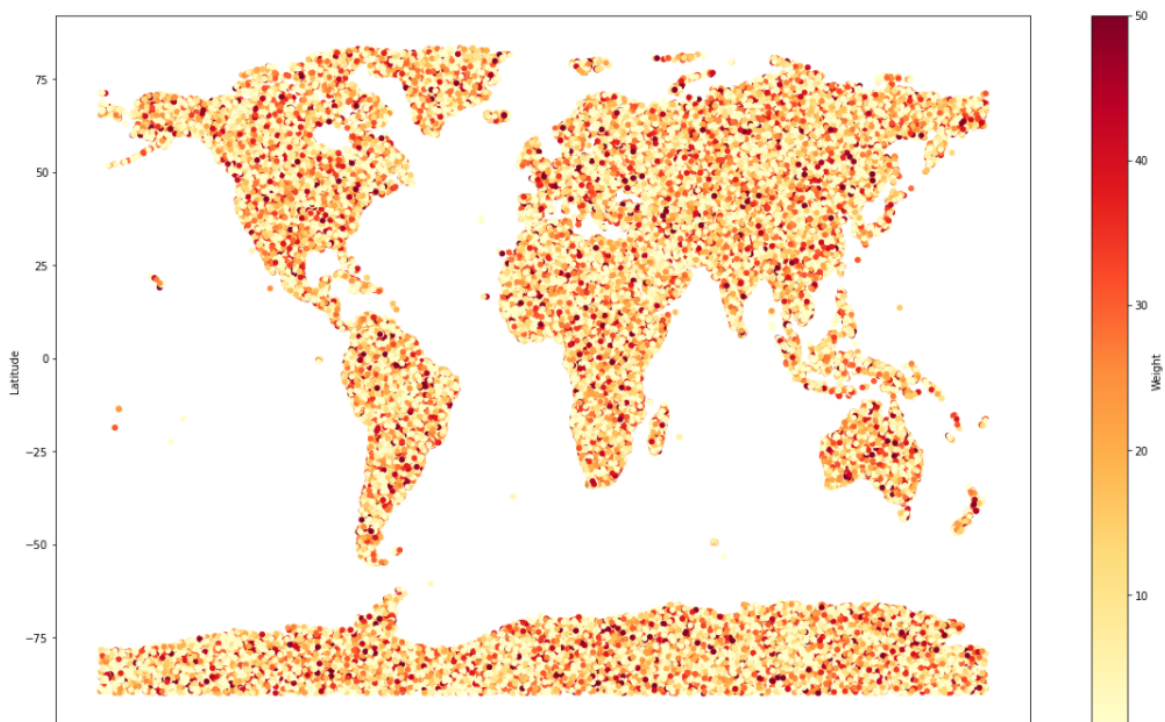
Analiza Danych

Zacznijmy od stworzenia reprezentacji graficznej naszych danych:



Niebieskimi punktami wypełniono miejsca na mapie do których mamy dostarczyć prezenty. Widzimy że nie jest to rozkład równomierny. Wynikający z tego problem to fakt że będziemy musieli znaleźć sposób aby nasz algorytm docierał na inne kontynenty tworząc optymalne trasy.

Spójrzmy teraz na to jak rozkładają się wagi prezentów.



Widzimy że w przestrzeni punktów do których dostarczamy prezenty, ich waga rozkłada się równomiernie w całości.

Wnioskiem jest że będziemy mieć większy problem z dopasowaniem drogi w makro-skali (podróż z bieguna północnego na antarktydę) niż w mikro (wybór następnego miasta do odwiedzenia).

Co do samych wag prezentów i ich wpływu na sanie, zakładamy że Sanie mają pojemność 1000 jednostek. Prezenty ważą do pięćdziesięciu jednostek. Dla 100 000 punktów będzie to około 5000 różnych tras do zaplanowania dla Mikołaja.

Wstępna propozycja rozwiązania

Opis pomysłu: Przeszukiwanie przestrzeni składającej się z 100000 punktów w celu znalezienia najbardziej optymalnej drogi jest problemem NP-trudnym. Zamiast przeszukiwać całą przestrzeń, proponujemy algorytm genetyczny, oparty o pomysł "kolonii mrówek". Pojedyncza mrówka wyruszająca z jednego punktu, do przemieszczenia ma do wyboru punkty z sąsiedztwa o zadanej odległości. Do każdego z tych punktów ma szansę się przemieścić, kierując się funkcją prawdopodobieństwa dla tego punktu i parametrów mrówki.

Wypuszczamy wiele mrówek z bieguna, które tworzą własne trasy. Na koniec pokolenia, liczymy wartość takiej trasy. Następnie wkładając je do kolejki priorytetowej, każemy położyć mrówkom na tych trasach feromon, im lepsza trasa tym więcej. Powtarzamy aż na kolejnych generacjach poprawa jakości będzie niższa niż nasz threshold. Otrzymując najlepszą trasę z x pokoleń wykreślamy wszystkie punkty do niej należące z mapy, i zaczynamy jeszcze raz.

W kontekście zadania, mrówka zaczyna podróż z bieguna północnego z pustym plecakiem. Po drodze plecak będzie się zapelniał, gdy po drodze będziemy "podnosić" prezenty. Gdy plecak się zapelni, mrówka musi wrócić na biegun. Będzie to miało wpływ na wartość funkcji prawdopodobieństwa podróży do danego punktu.

Na tę funkcję prawdopodobieństwa wpływ ma:

- **Stopień załadowania plecaka:** Mając plecak pusty preferujemy punkty leżące daleko od bieguna, wraz z zapelnianiem go chcemy punkty coraz bliżej bieguna.
- **Feromon:** Mrówka preferować będzie te drogi, na których wyczuwa feromon. Będzie to dla niej znak że ta droga była wysoko oceniana w poprzednich generacjach.
- **Ciężar prezentu a pojemność plecaka:** Mając pusty plecak chcemy aby pierwsze wylądowały w nim ciężkie prezenty. Przełoży się to na to że później renifery pozbędą się ich na początku trasy
- **Odległość:** Im punkt jest dalej, tym mniej prawdopodobne żeby była to najlepsza możliwa ścieżka.

Problem odległości w wyborze następnego punktu jest istotny, ponieważ będziemy zmuszeni do pokonywania oceanów jako następny punkt. Problemem także jest sąsiedztwo punktu dla którego liczymy prawdopodobieństwa. W idealnym przypadku liczymy prawdopodobieństwo każdego punktu na mapie od każdego, ale jest to złożoność $O(n^2)$, na którą nie możemy sobie pozwolić przy tak dużej liczbie punktów. Rozważmy najlepszy sposób generowania sąsiedztwa aby rozpoznać do których punktów mamy generować prawdopodobieństwo podróży:

- Genetyczne rozwiązanie komiwojagera jako porównywanie kolejno odległości pomiędzy trasami. Rozwiązanie to jest bardzo skomplikowane obliczeniowo, ponieważ mamy 100 000 miejsc i jeśli mielibyśmy porównać odległość każdego punktu z każdym to...
- Myśleliśmy, aby powyższy problem rozwiązać poprzez jednorazowe obliczenie odległości między każdymi punktami, dzięki czemu zaoszczędziliśmy czas na obliczaniu odległości za każdym razem, gdy mrówka podejmowałaby decyzję, gdzie iść dalej. Obliczenia te zapisali byśmy do pliku, z którego później wystarczyłoby tę odległość odczytać. Problemem w tym rozwiązaniu jest ilość miejsca w pamięci jaką te dane by zajmowały (~160GB).
- Rozważaliśmy również rozwiązania najpierw najlepszy albo najbliższy/najcięższy, jednak w tej sytuacji również musimy porównać każdy punkt na mapie z tym, w którym obecnie się znajdujemy, aby móc zdecydować, który jest najbliższy lub który jest najlepszy.
- Najlepszym rozwiązaniem wydaje się nam podzielenie wszystkich lokalizacji na grupy (np. zgodnie z południkami i równoleżnikami) i obliczenie współrzędnych środka każdej grupy. Dzięki temu będziemy mogli porównywać dany punkt z grupą, a dopiero, gdy wybierzemy grupę docelową będziemy mogli zdecydować, do którego punktu chcemy polecieć (np. zgodnie z tym czy chcemy zostawić lżejszy czy cięższy prezent). Podział na grupy sprawi, iż nasze rozwiązanie stanie się mniej skomplikowane obliczeniowo

Dzieląc problem na grupy, możemy dodawać do sąsiedztwa punkty z coraz kolejnych grup graniczących dopóki nie osiągniemy minimalnej liczby punktów.

Matematyczny opis modelu i jego parametry:

Przykładowa funkcja chęci mrówki:

$$d(s, c, w, v) = \frac{f(s,c)^{h1}}{hav(s,c)^{h2}} + (w \cdot \frac{1}{hav(o, c)} + (W - w) \cdot hav(o, c))^{h3} + (\frac{v}{w})^{h4}$$

Gdzie:

o - punkt startowy (biegun)

s - punkt z którego wychodzimy

c - punkt do którego się udajemy

w - obecny ciężar plecaka

v - ciężar prezentu w miejscu **c**

W - maksymalny ciężar plecaka

d - ocena jakości punktu **c** z punktu **s**, z ciężarem **w** w plecaku.

f - wartość feromonu na trasie **s**, **c**

hav - odległość między **s**, **c**

h1, h2, h3, h4 - wagi poszczególnych składowych funkcji.

Tę funkcję obliczamy dla wszystkich punktów z sąsiedztwa **S**. Następnie aby otrzymać prawdopodobieństwo poruszenia się w tym kierunku przepuszczamy funkcję chęci dla każdego punktu przez funkcję **Softmax**.

Algorytm Genetyczny

osobnik - wektor binarny włączający krawędzie drogi w grafie.

selekcja - Wybrana pula najlepszych zostawi za sobą feromon. Możemy eksperymentować z różnymi metodami selekcji.

mutacja - Mrówki nie podążające za feromonem. (losujące inną drogę)

Kryteria jakości:

- **Średnia ilość pustego miejsca w saniach na każdej drodze** - pozwoli nam zaobserwować czy maksymalnie efektywnie planujemy drogę pod względem pojemności sań.
- **Odchylenie standardowe odległości między punktami w drodze** - Pokaże nam czy nie robimy "pustych przelotów" np. zaczynając w Afryce i nie rozdając nic po drodze.

Definicja funkcji celu:

$$WRW = \sum_{j=1}^m \sum_{i=1}^n \left[\left(\sum_{k=1}^n w_{kj} - \sum_{k=1}^i w_{kj} \right) \cdot Dist(Loc_i, Loc_{i-1}) \right]_j,$$

n - ilość prezentów, które Mikołaj zabiera ze sobą w jedną podróż

m - ilość podróży, które odbywa Mikołaj

Wagę mnożymy razy odległość, którą przebywa Mikołaj i reniferki. Odległość obliczamy z formuły Haversine:

$$\text{hav}(\theta) = \text{hav}(\varphi_2 - \varphi_1) + (1 - \text{hav}(\varphi_1 - \varphi_2) - \text{hav}(\varphi_1 + \varphi_2)) \cdot \text{hav}(\lambda_2 - \lambda_1)$$

- szerokość geograficzna (odpowiednio punktu 2 i 1)
- długość geograficzna (odpowiednio punktu 2 i 1)

Założenia:

- W funkcji celu lokalizacja początkowa i końcowa to Biegun Północny.
- Maksymalna waga jaką możemy zabrać w podróż to 1000 (nie wliczając sani, które ważą 10)

Sposób mierzenia jakości rozwiązania (podsumowania wyników)

Jakość rozwiązania możemy mierzyć jako zmęczenie reniferów po przebyciu całej trasy. (Możemy porównywać to do sumy iloczynu podróży i wagi prezentu, gdy by Mikołaj miał zawsze po każdym dostarczeniu prezentu wracać na Biegun Północny - zabierał by ze sobą w jedną podróż tylko jeden prezent.)

Za najlepsze rozwiązanie możemy również przyjąć to, które wygrało konkurs organizowany przez serwis Kaggle i porównywać nasze wyniki do niego.

W trakcie działania programu, możemy również sprawdzić jakość naszego rozwiązania porównując ilość wolnego miejsca w saniach, które zostaje gdy wyruszamy z prezentami w drogę - jeśli ilość wolnego miejsca na wielu trasach będzie duża to należy zastanowić się, czy rozwiązanie to na pewno jest optymalne.

Należy zwrócić uwagę na fakt, iż problem jest NP-trudny, w związku z czym znalezienie najlepszego rozwiązania nie jest możliwe w czasie wielomianowym. Nie uda nam się znaleźć rozwiązania idealnego, jednak powinniśmy znaleźć rozwiązanie satysfakcjonujące.