# Selekcja i ekstrakcja danych

Wczytanie obiektu i jego cech do pamięci. Następnie przekonwertowanie do postaci, w jakiej obliczane będą odległości. Każda cecha rozpatrywana jest indywidualnie.

## Wejście

* Zbiór danych które zostaną poddane przetwarzaniu
* Lista cech
* Opis danych, pojedynczych rekordów i ich cech
  + **Numeryczne**  
    Konwersja przebiega natychmiast, ponieważ liczba jest po prostu kopiowana.
  + **Stałe**  
    Cechy w postaci identyfikatorów, stringów, czy bloków binarnych danych.
    - **Sortowalne**   
      Stałe, którym można przypisać jednolitą hierarchię, oraz ich ilość jest ograniczona. Tzn. dla każdych dwóch obiektów ze zbioru wartości danej cechy istnieje relacja między nimi. Odpowiednio jest to większość lub mniejszość.
    - **Porównywalne**   
      Cechy, dla których wartości możemy powiedzieć jedynie czy są równe, czy nie.
      * **Ograniczone**   
        Zbiór wartości danej cechy jest ograniczony.
      * **Nieograniczone**   
        Zbiór wartości jest teoretycznie nieograniczony, jednak na tyle powtarzalny, aby był jeszcze przydatny. Jego postać jest konwertowana funkją skrótu.

## Kernel

* **Blok**:Pojedyńczy rekord
* **Wątek**: Pojedyńcza cecha

## Wyjście

Tabela rekordów z wybranymi cechami w postaci liczbowej.

# Odległości

Zadaniem tej fazy jest wygenerowanie macierzy odległości pomiędzy każdą parą obiektów przetwarzanego zbioru.

## Wejście

* Tabela rekordów z cechami
* Opis cech
  + Typ
  + Waga

W specjalnych przypadkach, jak np. obraz, ekstrakcja i selekcja nie są potrzebne. Jednakże wtedy potrzebna jest zdolność czytania tych danych w tym etapie.

## Kernel

Pary mogą być obliczane niezależnie więc można przypisać je do bloków. Jeśli maksymalna ilość bloków jest mniejsza niż ilość par do obliczenia, wtedy bloke będą obliczały kolejne pary.

*[TBD. Program powinien wrócić do hosta po każdej parze czy kontynuować, aż wszystkie pary zostaną przeliczone]*

* **Blok:**  pojedyńcza para (lub kilka kolejnych)
* **Kernel:** Pojedyńcza cecha w parze rekordów
* Wczytać parę
* Obliczyć odległość (zależnie od typu I wagi)
* Zapisać

## Wyjście

* Macierz (a tak naprawdę jej połowa) odległości.

# Grupowanie

## Funkcja oceny rozwiązań

Dla każdego rozwiązania i każdego rekordu

### Wejście

* Lista rozwiązań
* Kryteria
* Macierz odległości

### Kernel

* **Blok**: jedno rozwiązanie
* **Wątek**: jedno kryterium

### Wyjście

* Listy (po jednej dla każdego kryterium) przypisujące danemu rozwiązaniu jego wynik

## Sortowanie rozwiązań

Sortowanie dominacji w rozwiązaniach.

* Każdemy rozwiązaniu przypisuje się dwa atrybuty
  + N – ilość rozwiązań dominujących to rozwiązanie.
  + S – zbiór rozwiązań zdominowanych przez to rozwiązanie

### Sortowanie list kryteriów

Przy krótkich listach rozwiązań, nie opłaca się tego robić. Przeszukiwanie po czterech listach będzie odpowiednio szybkie I oszczędniejsze w pamięci

### Generowanie atrybutów

#### Wejście

* listy wyników kryterium

#### Kernel:

* **Blok**:
* **Wątek**: Rozwiązanie  
  Przeszukaj kolejno listy wyników. Wylistuj wszystkie rozwiązania, które są gorsze od tego w każdym kryterium. Wynikowy zbiór zapisz jako atrybut S a wszystkim zwiększ o 1 atrybut N

## Selekcja rozwiązań

Według algorytmu selekcji NSGA-II

## Krzyżowanie rozwiązań

### Wejście:

* Zbiór “najlepszych” rozwiązań przeznaczonych do krzyżowania

### Kernel:

* **Blok**:
* **Wątek**: Para  
  Krzyżowanie pojedyńczej pary rozwiązań.

## Mutacja rozwiązań

Indywidualnie rozpatrzeć mutację jażdego rozwiązania.

### Wejście:

* Zbiór rozwiązań

### Kernel:

* **Blok**:
* **Wątek**: Pojedyńcze rozwiązanie