

first define the weight and value of each box

Then define the chromosome class, each chromosome will have its own state, value and weight.

Then use three global value, these value will change as the function are called

max_last is the max value until now

diff_last is the difference between last two crossovers

chromosome_max is the max chromosome

times is the times of satisfy the finish condition, if it is the first time, using mutation

There are 5 function.

Init(), fitness(), filter(), crossover(), is_finished().

Init(): initialize the chromosome with random state

fitness(): calculate the values and weight for each chromosome

filter(): filter the chromosome we don't want: if its weight is larger than 120, change its value to zero, sort the class based on it values, and discard last 50%

crossover(): each time the chromosome in the list will randomly choose another chromosome to crossover, each time will produce a random value as position, and change wither each other, finally return a list with 12 chromosome, including the father generation and child generation

is_finished(): judge whether it has finished, if the last two difference is both smaller or equal to 1, the meet the requirement and return true. Also, in this function, it will store the global max value.

If it is the first time meet is_finished(), then it may be not the best result, it may have discarded some useful gene before, so I used mutation, randomly change a gene and change it. Then call the function above and run again.

Because this algorithm is randomly choose gene to crossover, so its result depends on the initial chromosome state and some times it is not stable, so we may get some sub-optimize result like 17 or 19, but typically we will get 20.

Reference:

<https://blog.csdn.net/u010451580/article/details/51178225>

<https://zh.wikipedia.org/wiki/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>

<https://blog.csdn.net/lilong117194/article/details/78906488>