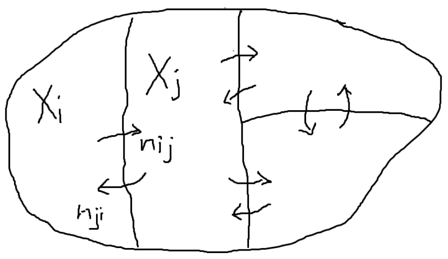
Partitioning problem

by Pascal CHEN.

Contents

1 Introduction



Partitioning

In this review, I will introduce the problem of partitioning.

The context is the following: we want to divide a work into several sub-tasks, in order to parallelize all this and to make the work as quickly as possible.

However some tasks are interconnected by a boundary, ie a partition is very strongly related to its neighbors, for example in the case of the mesh.
If we cut a mesh so that each partition performs a calculation on it, we will cut so that each mesh is a united block and not pieces of meshes scattered.

Here is the problem: we have N partitions where each partition has a workload, but the workload is different for partitions, so some computer cores are forced to wait for their neighbors to complete the program.

We will therefore seek to find a way to balance the workload with the least possible exchange between neighbors so that the exchange is done as quickly as possible. This is therefore a linear optimization problem where the number of workload exchanges must be minimized.

It can be modeled as follows:

Let i and j be in $[1; N]$.
Let N_i be the workload of partition number i and X_i the normalized workload of N_i, ie

$$X_i = N_i - \tilde{N}$$

where :

$$\tilde{N} = \frac{1}{N} \sum_{i=1}^{N} N_i.$$

Let n_{ij} the workload exchange from i to j.
Let $\mathcal{N}(i)$ the set of neighbors of i.
Note that $\forall i,\, n_{ii} = 0$ and $\forall i,\, \forall j \notin \mathcal{N}(i),\, n_{ij} = 0$.

$$\text{Minimize} \sum_{i,j} n_{ij}$$
$$\text{Subject to } \forall i,\, X_i - \sum_j n_{ij} + \sum_j n_{ji} = 0.$$

Note that

$$\sum_i X_i = 0$$

and one of the equations is redundant in the constraints.

2 Results

In order to solve the problem, there are many algorithms for linear programming referred at this link:

https://en.wikipedia.org/wiki/Linear_programming

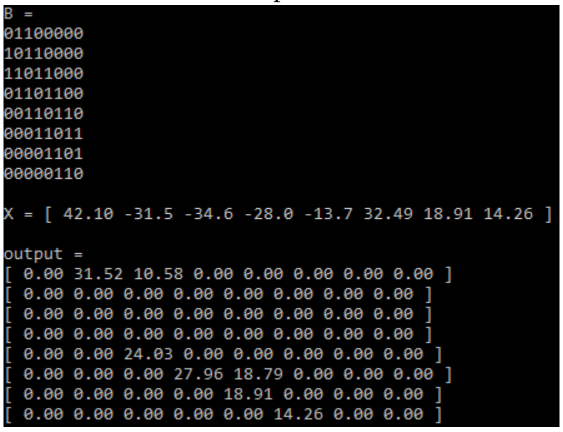
Obviously, the best algorithm for our case is simplex method.

Of course, a constraint equation can be seen as two inequalities, but this is a trap for the simplex method, because you multiply the number of lines by two and add columns to the simplex matrix, whereas the method of the simplex is exponential although on average is polynomial.

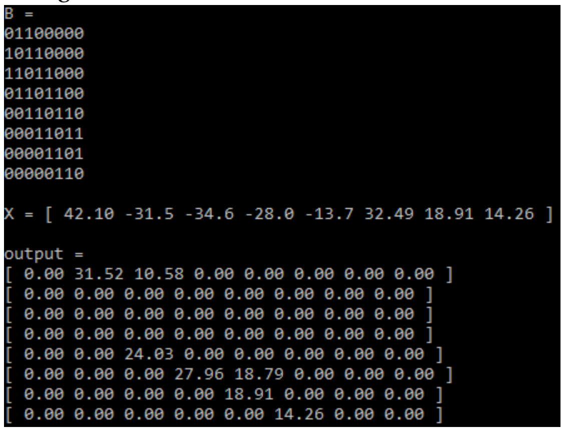
Moreover if you run the simplex algorithm on inequalities to describe an equation, you will notice that the two lines will be very similar.
In this case, it is better to use two phases method.

I have tried to implement simplex method in C with dense matrix in order to compare with the library glpk (gnu linear programming kit).

here are the results for 8 partitions and maximum of 4 neighbors:



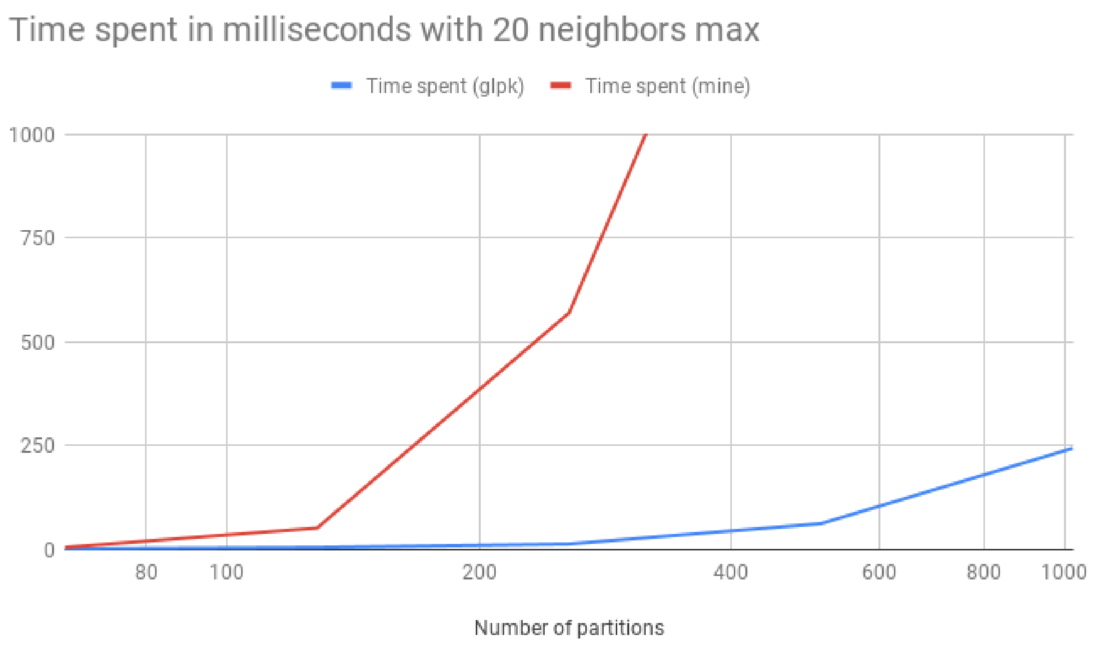
```
B =
01100000
10110000
11011000
01101100
00110110
00011011
00001101
00000110

X = [ 42.10 -31.5 -34.6 -28.0 -13.7 32.49 18.91 14.26 ]

output =
[ 0.00 31.52 10.58 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 24.03 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 27.96 18.79 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 18.91 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 14.26 0.00 0.00 ]
```

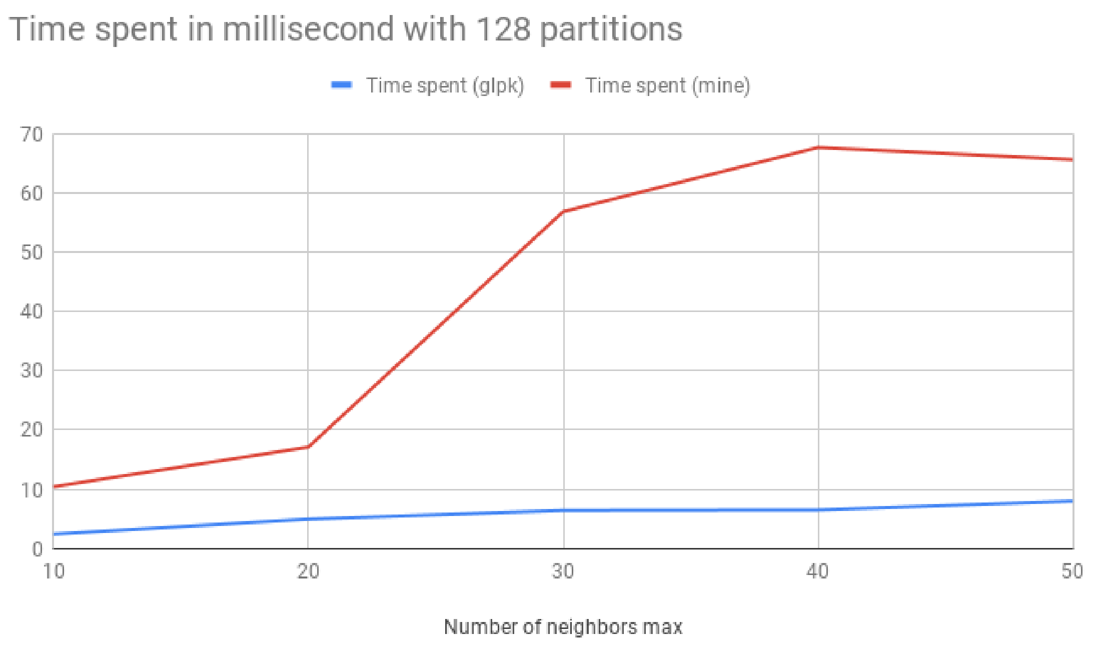


```
B =
01100000
10110000
11011000
01101100
00110110
00011011
00001101
00000110

X = [ 42.10 -31.5 -34.6 -28.0 -13.7 32.49 18.91 14.26 ]

output =
[ 0.00 31.52 10.58 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 24.03 0.00 0.00 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 27.96 18.79 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 18.91 0.00 0.00 0.00 ]
[ 0.00 0.00 0.00 0.00 0.00 14.26 0.00 0.00 ]
```

Left : glpk, right : mine

The values are generated randomly, I have copied the X output for my X, and the result are both acceptable, the objective sum is equal to 146.05 for one part and 146.1 for the other part due to the imprecision.





Due to the high performance of glpk, using sparse matrix, it is large better to use this library for memory and time complexity.

This kind of program can be linked to an input "ai" that can predict the workload.