# Report

## Shu Zhang 001255490

**Problem Statement:**

Business location choosing is perhaps the most important decision for a business owner, so it requires precise planning and research. It involves many factors such as proximity to living area, shop rent, population flow and much more. In order to make the most profits, we need to generate an algorithm to measure different locations hence find the best location.

**Problem Solution:**

In this project, I combined **Reilly's Law of Retail Gravitation** with **Particle Swarm Optimization**.

**Reilly's Law of Retail Gravitation :**

Suppose C is a main living area, the attractiveness of two retail centers A and B is proportional to their population and inversely proportional to the square of the distance to C. Here is the formula:

Ba / Bb = (Pa / Pb)(Db / Da)2.

(Pa, Pb represent the population of B and A, Db, Da represent the distance to C)

So I set the fitness value as Ba / Bb, cause B is my basic point, the larger the fitness value is, the better the location is.

**Particle Swarm Optimization:**

In this part, I treat each particle as an independent thread and I generate 50 threads totally.

Here is my object model:

PublicInfo

MethodSet

Particle

PSOProcess

PBest

Velocity

Location

My main process is in PSOProcess class which implements PublicInfo class. In PublicInfo class there are some constants: swarm size, max iteration and etc. MethodSet class provides some concrete methods that could be called for by PSOProcess.

There are four steps in PSOProcess:

1. Calculate the fitness value of current particle.
2. Compare the fitness value with particles nearby, update PBest.
3. Calculate GBest.
4. Update location and velocity.

**Thread:**

In my project, each particle is a thread. In the PSOProcess, particles only care about their own PBest. The initialized swarm is defined in PublicInfo as public variable, so every particle can use it.

Here is a simple graph that can explain it:

Swarm:

Particle

PSOProcess

And here is the whole flow of my project:

Particle thread begins

Main thread begins

Initialization

Calculate fitness value

Compare it to pBest, gBest

Execute and search

update

End

Meeting ending condition?

End

And here are some screenshots of my project:





