Particle Swarm Optimisation for Portfolio Optimisation Problems in Finance

Project Plan

Anthony S. Chapman

Supervisor: Dr. Wei Pang,
Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, UK

Introduction

Optimisation models play an increasingly important role in portfolio management decisions. Many computational portfolio problems ranging from asset allocation to risk management can be solved using modern optimisation techniques [1]. I have been interested in the analysis of trading for a very long time and I believe that with the help of statistical mathematics and a strong and safe computing language, a “smart” system can be created and given the correct guidelines, it can independently manage a portfolio [2, 3] and arrive at the optimal (maximum profit with minimum risk) trading decisions. Particle Swarm Optimisation is at the frontier of both linear and non-linear optimisation methods yet there are not that many applications to finance[4].

Goals

The goal of this project is to make an application to optimise a portfolio using PSO methods. For example, give a budget and some assets you are interested in, the system will decide what to buy or sell in order to maximise profit and minimise risk. If I have enough time I would like to give it a nice user friendly GUI and maybe even run it on the cloud, but this are unlikely.

Methodology

The following point illustrate how I will carry out the project:

- Familiarise myself with PSO and apply it to integer programming problems.
- Verify whether PSO is more efficient than KKT [5] and Simplex [6].
- Research portfolio optimisation, example, when measuring risk, would be better to use variance or quantile-based risk measures[7].

- Implement simple portfolio test solutions, for example, implement a PSO algorithm to test whether it be financially beneficial to buy or sell a collection of assets in a portfolio[8].
- Learning about PSO in Haskell and possibly parallel programming modules in Haskell.

Resources Required

I aim to make an application which can run on a standard PC or laptop.

- Dual core processor with 1.6GHz or more.
- Haskell: Parallel Haskell Compilation System, version 7.4.2 or higher.

Risk Assessment

I will need to be most efficient in my time management given that I also have to study for other modules in mathematics. Finding the balance and rhythm for working is essential to the success of my proposed project.

A possible problem would be that the constraints for a given optimisation process don't make a compact set. Meaning that it won't necessarily have an optimal solution and the program might crash or give incorrect results. To tackle this, I will implement safe guard measures to determine whether the region constraints is well defined, thanks to Haskell's lazy evaluation, it won't try to work out the solution before the verification has been carried out.

Timetable

- 2 weeks: Literature review on PSO, portfolio optimisation and risk assessment.
- 2 weeks: Testing what I learnt from literature and experiment with ideas.
- 4 weeks: Implement main system.
- 3 weeks: Write report.
- 1 week as buffer in case something takes longer than expected.

References

[1] Konstantinos P. Anagostopoulos. Mutiobjective evolutionary algorithms for complex optimisation problems. 2009.

[2] Wikipedia. Portfolio optimisation. http://en.wikipedia.org/wiki/Portfolio_optimization. Accessed: February,2014.

[3] Gerard Cornuejols. Optimisation methods in finance, 2005.

[4] James Kennedy and Russel Eberhart. Particle swarm optimisation. 2001.

[5] Wikipedia. Karush-kuhn-tucker condition. http://en.wikipedia.org/wiki/Karush-Kuhn-Tucker_conditions. Accessed: February,2014.

[6] Wikipedia. Simplex method. http://en.wikipedia.org/wiki/Simplex_algorithm. Accessed: February,2014.

[7] Konstantinos P. Anagnostopoulos. Multiobjective evolutionary algorithms for complex portfolio optimization problems, 2009.

[8] E.C. Laskari. Particle swarm optimisation for integer programming. 2005.