

CS3910 Computational Intelligence

Computational Intelligence Solution for Market Pricing

Damon Tran

Student Number: 160111190

Justification of Solution

­­­The two solutions that have been chosen to solve the given problem are, Particle Swarm Optimisation(PSO) and Artificial Immune System (AIS).

The reason I have chosen these two approaches is because, I think these algorithms are easily applicable to the context and able to represent the problem well.

PSO works to find the best solution by using special set of operators with self-adjustment and discovery. PSO as its name suggests uses particles, these entities will search through the problem space and discover parameters of their own which evaluate to a given fitness level, based on that it will be checked against the particles personal best fitness level and the global best fitness level that the swarm has found. Other particles will do the same, and eventually all change their values based on the swarms best.

How this can be applied to the Pricing problem is simple, each particle will search for a set of prices and evaluate it, and all other particles will do the same. Then they all check whether the prices they have found produces a revenue greater than that of the swarm’s global best. If it has, store that particle’s price list and revenue as the new global best.

AIS works with the goal in mind to mutate itself, until it produces something that works very well for the given problem i.e. (blood-cells mass producing effective antibodies to destroy pathogens). In the context of the given problem the AIS algorithm will mutate the prices until it produces a “good” revenue and clone itself and continuously mutate.

Novelty and insight of solution (report, code)

PSO seems like it should work well on this given problem, because regardless of the initial values it initialises (good or bad), the swarm will always generate larger and larger revenues based off the global best. In addition, if the particle generates an invalid price list that cannot be evaluated to a revenue, it does not matter because the particle’s attraction to its personal best and global best will cause it to return to the feasible region anyway. Meaning eventually, it will produce a valid price list which is a benefit because it means the algorithm is quite flexible and robust guaranteeing it will provide a solution.

There are a few methods for PSO constraint handling such as boundary walls, solution repairs and velocity clamping. Including the one mentioned above as invisible wall.

I could use the PricingProblem’s isValid method where I create initial random price lists to ensure the initial population is valid and add it to the particle, but it is not needed if the invisible wall works.

The performance of the algorithm will vary depending on the parameters that have been set for it, being number of particles and number of iterations. There is a specific formula that is known for setting the swarm size to produce good solutions for problems.



I will be running various test on the swarm size with the same number of iterations and compare the results to see which parameters perform better.

Like PSO, AIS also has many benefits such as its robustness and use of memory as well as it being autonomous (finding solutions by themselves). In relation to the problem I also believe it will perform well and produce good solutions. Due to the nature of how it works (selective cloning). Where it only clones the best solutions, ‘best’ meaning solutions which produce the biggest revenues.

The algorithm aims to apply a low mutation rate to good solutions and conversely high mutation rates to bad solutions. Therefore, every good solution that is produced should not change much and will be kept, as well as the bad solutions changing drastically and producing better results which provides better chance of generating good solutions.

And at the end of the mutations the mutated clones are collected and put with the parents, where only the best x number of solutions are kept. The rest being discarded and changed for new random solutions. Therefore, you only clone and mutate the best solutions gradually getting better and better values.

**Evaluation**

PSO tests:

Here in this chart you can see that the overall average revenue produced varies vastly depending on the swarm size factor. All these tests were done with same number of iterations (200).

Swarm size of 5 did the poorest and only managed to produce an average revenue of 3307.24, conversely the greatest revenue average at 3547.958 was produced by a population of 50 surpassing even the swarm factor size formula mentioned earlier known for producing good solutions.

PSO’s population of particles is a very important factor when it comes to finding good solutions, and that it is dependent on the problem and problem size.

This graph shows the comparison of values across 5 tests with the same swarm population but different number of iterations. Number of iterations shows it is an important parameter in the PSO algorithm and is positively correlated with higher revenue solutions. Producing an average, greater than that of 200 iterations by 247.12.

AIS Tests: