

# Final Project Checkpoint 2

Sam Kim<sup>1</sup>, Nick Merrill<sup>2</sup>, Crystal Stowell<sup>3</sup>

<sup>1</sup>samuelkim@college.harvard.edu

<sup>2</sup>merrill@college.harvard.edu

<sup>3</sup>cstowell@college.harvard.edu

## Progress

- The Cuckoo Search has been tested on the Fence Problem, Box Problem, Michalewicz Problem, and Manufacturing Problem. The algorithm seems to be well-polished.
- The random walk has been tested via the Cuckoo Search. It appears to be working as expected.
- Particle Swarm Optimization (PSO) has been implemented and seems to be able to find successively better fitness solutions over numerous iterations, but is not currently coming close to the globally optimal solution. More work is required to improve this algorithm.
- The Nurse Scheduling problem has been coded. It is still in its initial stages so there are very few constraints. The fitness is ranked solely on employee happiness, and does not yet account for the cost of “over-scheduling.” There are also currently no restrictions on the maximum number of shifts that can be worked in a 24 hours period or the maximum shifts that can be worked back-to-back.
- A basic user interface has been created in the terminal so that users can choose which problem they would like to solve, which algorithm they would like to use, and what they would like to input for the various variable(s).

## Problems

- Since the inputs and output of the Nurse Scheduling Problem appears to be so different from that of previous problems, lots of time and thought was put into how the problem should be structured. It was decided that the solution class would not change (it remains an ArrayList of Doubles), so appropriate alterations to the solutions are made within the Nurse Scheduling Problem to test for constraints and fitness. Namely, each value in the solution which is a double between 0 and 1 is converted to an int through appropriate rounding, and the single arraylist is converted to a matrix with rows and columns that can be easily assessed by the rest of the problem.
- Generalization of the Nurse Scheduling Problem is proving to be very difficult. If the

shift-requirements are different from job to job, how can these be accounted for in the problem? It may be that certain requirements will have to be “hard-coded,” limiting the generalizability of the problem.

- PSO currently is not globally optimizing the fitness function. Debugging is underway to find the culprit.

## **Teamwork**

This week, the task of formatting the nurse scheduling problem required the brainpower of all three teammates. Since the implementation of this problem requires both the algorithms and solutions classes, the knowledge of all three individuals was needed to determine the best possible design.

- Sam
  - Completion and debugging of Particle Swarm Optimization algorithm
- Nick
  - user interface to enter inputs, including a soon-to-be CSV file for the nurse scheduling problem.
- Crystal
  - testing of the manufacturing problem
  - implementing the nurse scheduling problem, tested with minimal constraints

## **Plan**

- Resolve any outstanding issues with PSO and the various problems that it may be passed. Begin modifications/hybridizations of both Cuckoo Search and PSO. (Sam)

- Enhance the usability of the program by simplifying user input requirements as much as possible. (Nick)

- Continue to add constraints and generalizability to the nurse scheduling problem. (Crystal)

- Begin running tests to compare the performance (in terms of both convergence to the globally optimal solution and time/number of generations) of the two optimization algorithms and several different problems. (All)

## **How to Use Our Program:**

- After loading the JAR file . . .

- Choose the problem you would like to solve and the algorithm you would like to use to solve it by following the command prompts and entering the appropriate integer.
- Enter appropriate inputs for the variable(s) of that problem.
- Watch it get optimized!