README

What is this repository for?

This is simply a showcase of my first Haskell project. Feel free to try it, use it if you think there's anything valuable in there or send me comments if you have any - it's best to get my attention on twitter https://twitter.com/rickdzekman or look me up on LinkedIn.

How do I get set up?

- The code will automatically generate a new (and random) TSP-OPTIMIZE problem - and also a problem.gv graphviz file. If you have GraphViz installed simply run this to generate an image output: circo -Tpng output.gv -o tspCirco.png
- It will then run the Ant Colony algorithm over it. The end result of the algorithm will get printed in the console.
- It will actually print (w/ Show) the state of the "AntSolver" in the last round of execution. The key part to look for is the Leaderboard and it will show something like this: Leaderboard: >> 1. (39,[Cycle w/ score: 39 and path: [Edge (5): 6–10,Edge (3): 2–10,Edge (4): 0–2,Edge (5): 0–5,Edge (5): 5–8,Edge (4): 3–8,Edge (2): 3–4,Edge (4): 4–7,Edge (2): 1–7,Edge (3): 1–9,Edge (2): 6–9]]) >> 2. (46,[Cycle w/ score: 46 and path: [Edge (3): 2–10,Edge (5): 6–10,Edge (5): 6–7,Edge (5): 7–9,Edge (3): 1–9,Edge (5): 1–4,Edge (2): 3–4,Edge (4): 3–8,Edge (5): 5–8,Edge (5): 0–5,Edge (4): 0–2]])

(in this example #1 = best score: a Hamiltonian cycle with a total weight of 39)

Testing Different Configuration Options

The module `ProbableAlgorithms.AntColonyCompare` provides an additional facility to compare different types of configurations and score their performance. An AntConfigCreator takes a "Country" (TSP Graph) and returns an AntConfig. There is a list of test configs to trial.

NB. The AntColonyCompare function `compareConfigs` will take some time to execute depending on the number of configs being tested. Printing the compareTest function will output a table with the IDs of the different configs along with their scores.

fromList [(2238,[4]),(2262,[6]),(2419,[2]),(2437,[5]),(2516,[1]),(100917,[3])]

License

This is available under an MIT License along with the usual caveats - no warranties (expressed or implied) or guarantees or promises of any kind.