# CSE/ECE 848 Introduction to Evolutionary Computation

## Module 3 - Lecture 14 - Part 1

# Comparison of EC Methods: Motivation

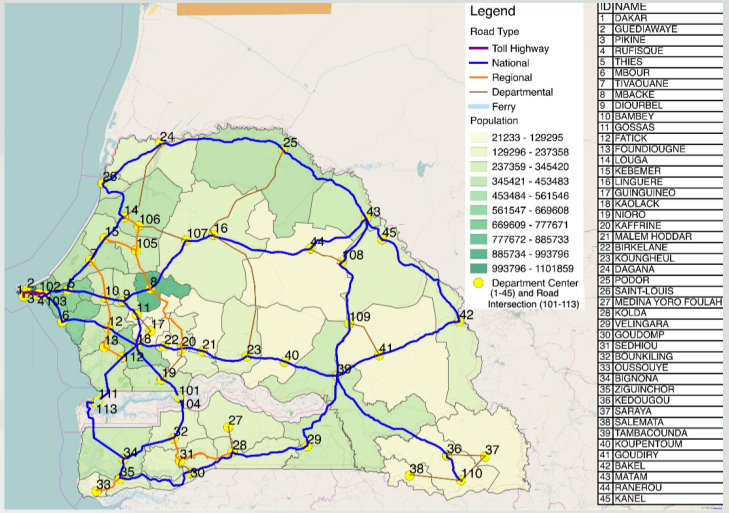
Wolfgang Banzhaf, CSE
John R. Koza Chair in Genetic Programming

# What do we want?

- What do we want an EA to do for us?
  - We want them to solve our problem
- Applications
  - Design (one-off) problems
  - Repetitive problems, including online problems
- Academic Research
  - Method exploration

# A Design Application

- Example: Optimization of extensions of an existing road network
- Complex, multi-objective problem, subject to many constraints
- One excellent solution created once is sufficient
  - Solution quality is priority
  - Time to solution not important
  - OK, if algorithm does not reach solution for some initial conditions
- Specific aspects allow to research special operators/representations not tranferable to other problems




From: Y. Wang et al., *"National and Regional Road Network Optimization for Senegal using Mobile Phone Data"*, NetMob 2015

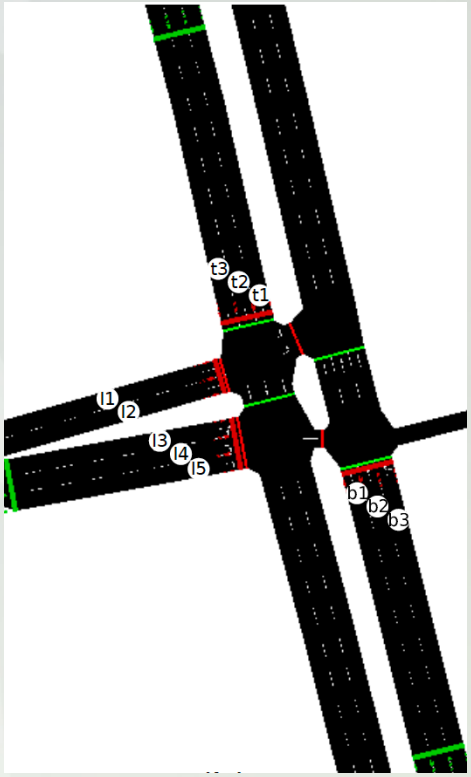
# A Repetitive Application

- Example: Domestic transportation firm with several trucks and drivers, given daily schedule every morning
- Pick-up and delivery plan for each truck and driver
- EA must find a good solution quickly, reliably, and repeatedly, i.e. for different instances
  - Speed is very important
  - Solutions must be good, but not necessarily optimal
  - There might be only time for one run, therefore, quality should have low variance
- Algorithm might be used in many different companies


From: [caliper.com](caliper.com)

# A Online Application

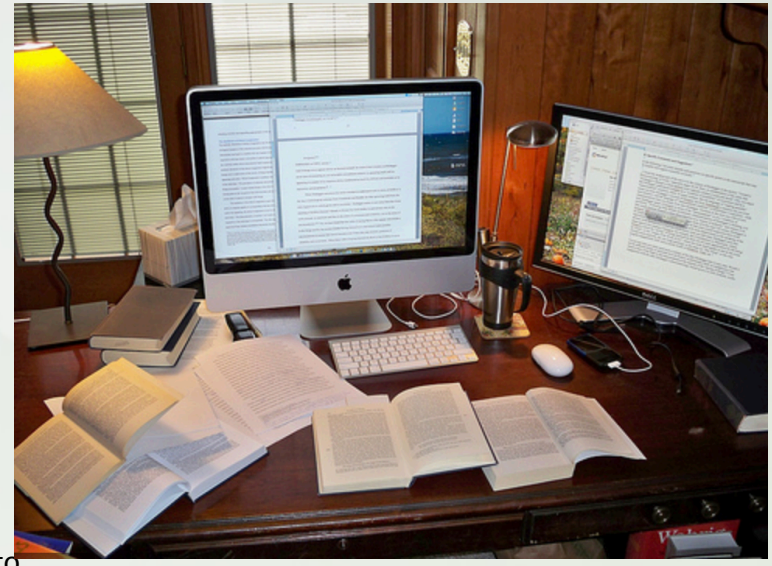
- Example: Traffic light controller
- Set up traffic signals to optimize traffic flow under different conditions (weather, daytime, etc.)
- Sensors allow to calculate density
- Online mode, optimizing a baseline schedule to adapt to current (hour to hour) conditions
- Consecutive problems are similar
- Repetitive problem solving
- Must be fault tolerant and robust (e.g., noise in the data or missing data should not prohibit solution)




From: SUMO traffic control simulator illustration as taken from E. Ricalde, PhD thesis, 2019

# Academic Research

- Get a good solution for a given challenging problem
- Show EAs are applicable in a novel problem domain
- Show that an EA with some new features is better than another EA
- Show that the EA is better than an existing non-evolutionary algorithm
- Find best parameter set-up, given conditions
- Obtain insight into the workings of the algorithm
- See how algorithm scales with problem size



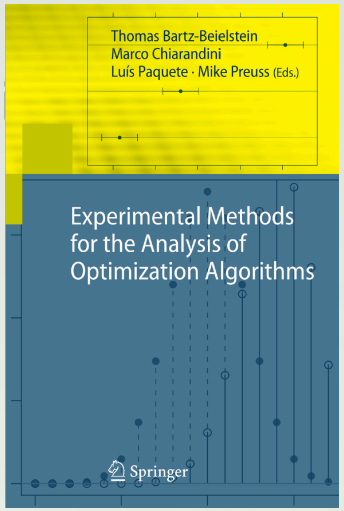
# Design of Computational Experiments

- A generic problem (e.g. TSP) and its instances (particular numerical cases)
- Experiments involve running a set of algorithms on a number of instances of the problem
- It starts with a set of questions about the heuristics under study
  - Problem characteristics, e.g. size, #constraints …
  - Algorithm components, e.g. stopping criteria, move selection
- Often: Pr… …d, instances… …es)

| | Algorithm 1 | Algorithm 2 | … | Algorithm n |
|---|---|---|---|---|
| Instance 1 | | | | |
| Instance 2 | | | | |
| … | | | | |
| Instance m | | | | |

# Combinatorial Design

- Randomized algorithms (like GAs) require on top of that: Repeated runs under same parameter conditions, as results are different from run to run
- Often experimental factors are run with just 2 conditions: high and low, but to get clarity whether there is a linear relationship, we need at least 3 conditions!
- Example:
  - 8 problem factors, with only 2 conditions -> 256 comb
  - 3 replicates -> 768 combinations
  - 3 algorithms factors with only 2 conditions -> 6144 combinations
  - 5 times with different random seeds -> 30,720 runs !!



# Experimental Analysis Myths
## (from Wineberg and Christensen)

- Results from 1 run is all that is needed
  - No, shows only proof of concept
- The best value achieved in a set of runs tells you something about the population distribution
  - No
- Using the same RNG seed for both systems provides a fairer comparison
  - It doesn't - it's the statistical properties of the system that we are looking for
- One system is obviously better than the other when looking at the data/graphs - no statistics necessary
  - If it is so obvious, one might as well show with statistics
- My average is better than your's means my algorithm is better than your's
  - In the best case, you would need to take variance into account