

MASON and ECJ Integration

AKM Khaled Ahsan Talukder

George Mason University

atalukde@gmu.edu

June 16, 2013

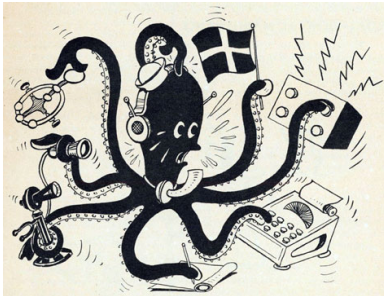


MASON + ECJ

- ▶ MASON: A multiagent simulation toolkit
 - ▶ ECJ: A powerful research framework for evolutionary optimization (which also supports massively parallel optimization)
- both were developed at the Autonomous Robotics Lab, GMU.



Setting parameters can be a nightmare !



- ▶ Scientists make **models**.
- ▶ Complicated models (i.e. ABM) requires a large number of **multiple inter-dependent parameters** to be tuned manually – prohibitive.
- ▶ Can we automate this ?

Preliminaries

Model Parameter Types

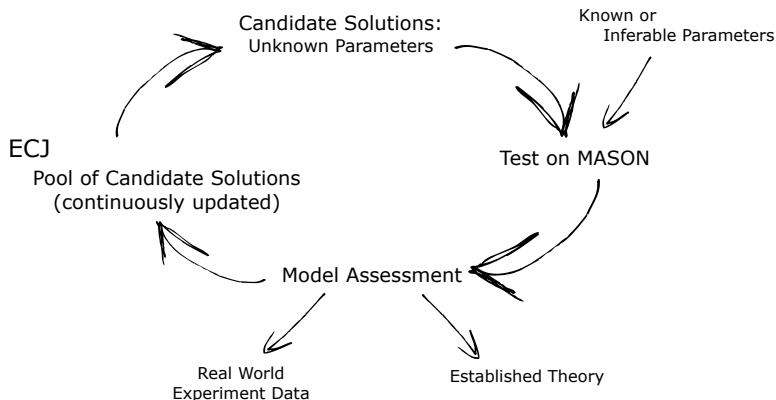
- ▶ Known Parameters – whose settings already known for a fact
- ▶ Inferable Parameters – whose settings should be set in a certain way (*according to a particular theory*)
- ▶ Insensitive Parameters – over whose settings the model is expected to be insensitive
- ▶ **Unknown Parameters – on which an experimenter has no idea/control over.**
(*most interesting ?*)

Goals

- ▶ To produce a certain kind of output which is predicted by a theory, or
- ▶ to match and validate against known real-world results.



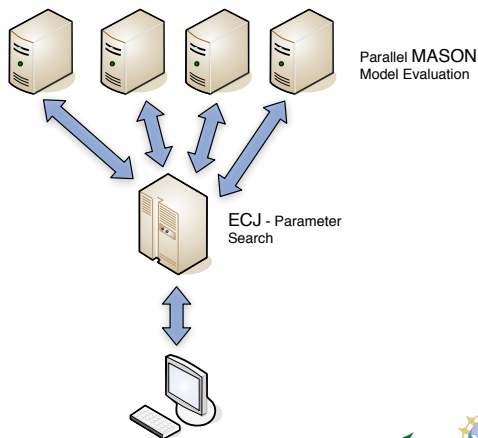
Optimization as An Automated Procedure



– What if an **expected** parameter setting is not found ?

Parallel Evolutionary Optimization – A Bird's-Eye View

- ▶ Extremely expensive – need to optimize the model by running it many times.
- ▶ Ought to be done in parallel.



First Example – *RebeLand*

RebeLand – Simulating a *Society*

Model of “Socio-Economic Stability” developed on top of MASON at the Center of Social Complexity, GMU.

Objectives to Maximize (a.k.a *Multi-objective Optimization Problem*)

- ▶ Population satisfaction
- ▶ Amount of money skimmed from the populace through corruption



Parameters/Knobs

Unknown Parameters to Set (*all scaled to 0...1*)

- ▶ State Corruption Rate
- ▶ State Tax Rate
- ▶ Maximum State Reserve
- ▶ Minimum State Reserve
- ▶ Minimum Spending on Populace
- ▶ Maximum Police Per Capita
- ▶ Initial Reserve Army Ratio
- ▶ Standing Army Size
- ▶ State Attack Interval

Details You Don't Care About

Optimization Algorithm

NSGA-II, 6000 evaluations.

Testing (i.e. *Generalization*)

Individual *Parameter Settings* are tested by running RebeLand on MASON 8 times – mean results were considered.

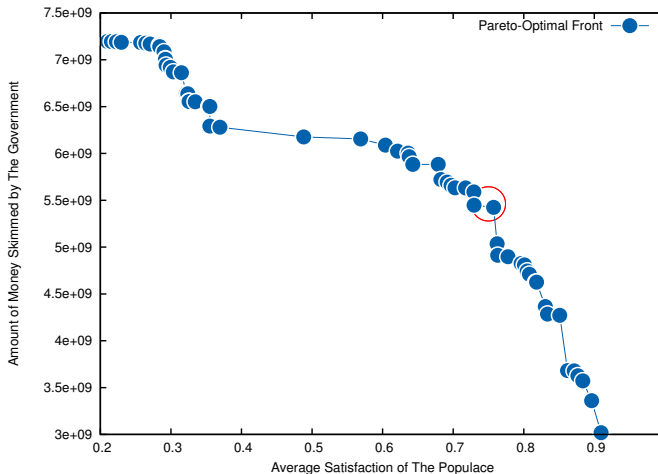
Parallelization

Master-Slave Evaluation, 30 Slave Units on Hydra

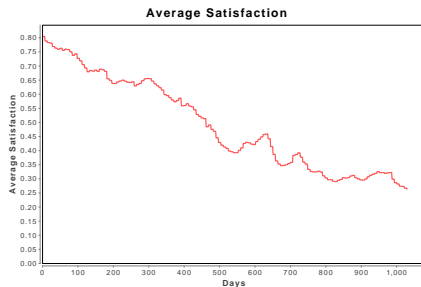
Total evaluation time: about 1 hour



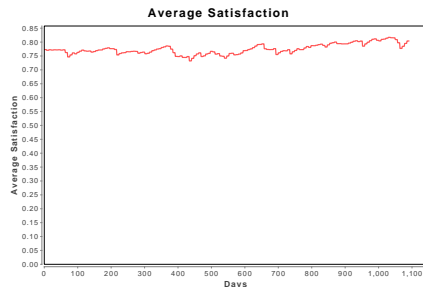
Results: The *Pareto Front*



Results: Population Satisfaction



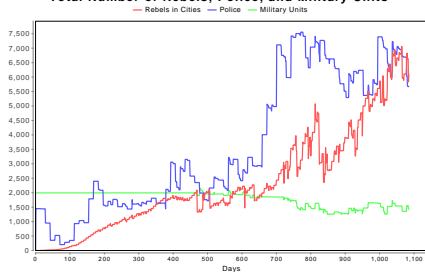
Before Optimization
(Original Parameters)



After
Optimization

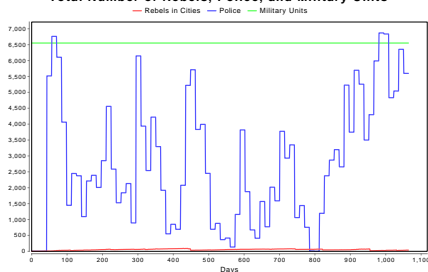
Results: The Overall Situation

Total Number of Rebels, Police, and Military Units



Before Optimization
(Original Parameters)

Total Number of Rebels, Police, and Military Units



After
Optimization

Results: Some (interesting) numbers

- ▶ State's corruption rate – **86%**
- ▶ State's tax rate – **77%**
- ▶ State's maximum reserve rate – 84%
- ▶ State's minimum reserve rate – 77%
- ▶ Minimum benefit share to the populace (i.e. public spending) – **98%**
- ▶ Maximum number of police force per capita – **31%**
- ▶ Initial reserve army ratio – 0.01%
- ▶ Standing army size – 0.06%
- ▶ State's attack frequency (to suppress rebels) – **78%**

Conclusions

Lesson ?

It's **fine** to run a **corrupt** government and keep its people **happy** – *only* if you have a **high** public spending, **large police force** and a very **frequent attack** on rebels.

Utopia ?

These results are “interesting” – i.e. *communist dictatorship* ?

Revelations ?

Does this reveal cracks in the *RebeLand* model semantics? Or a bug in the code?

Next: PacMan – Evolving Agent Behaviour

- ▶ MASON → **Behaviour** specifications
- ▶ ECJ → **Optimization** by **Evolution**
- ▶ MASON + ECJ → Evolving Optimized Behaviour
- ▶ **Target:** The game of PacMan

An Evolved Pac Behaviour



Demo

Summary



- ▶ Parameter optimization could be very important for assessing a **model's validity**, finding **more interesting insights**.
- ▶ Surely, symbiosis between MASON and ECJ – has a lot more to offer.
- ▶ Building a unified/standardized APIs to help plugging ECJ capabilities into MASON models – could just be a tip of the iceberg.

Questions?

Suggestions?

