Autonomous Systems Evolution

Why plan when you can just try?

Damien Anderson, Petros Papadopoulos, Marc Roper

Dept. Computer and Information Sciences

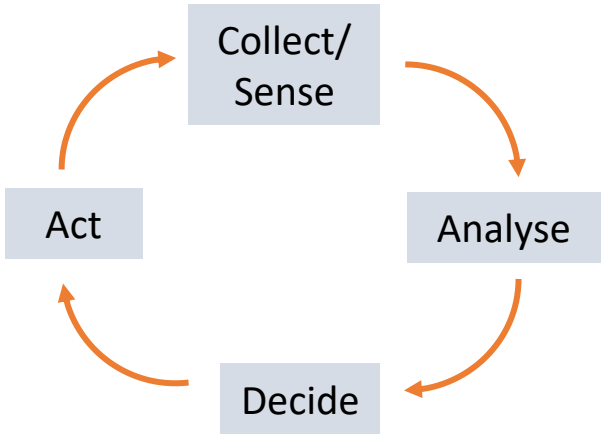
University of Strathclyde

Glasgow, UK

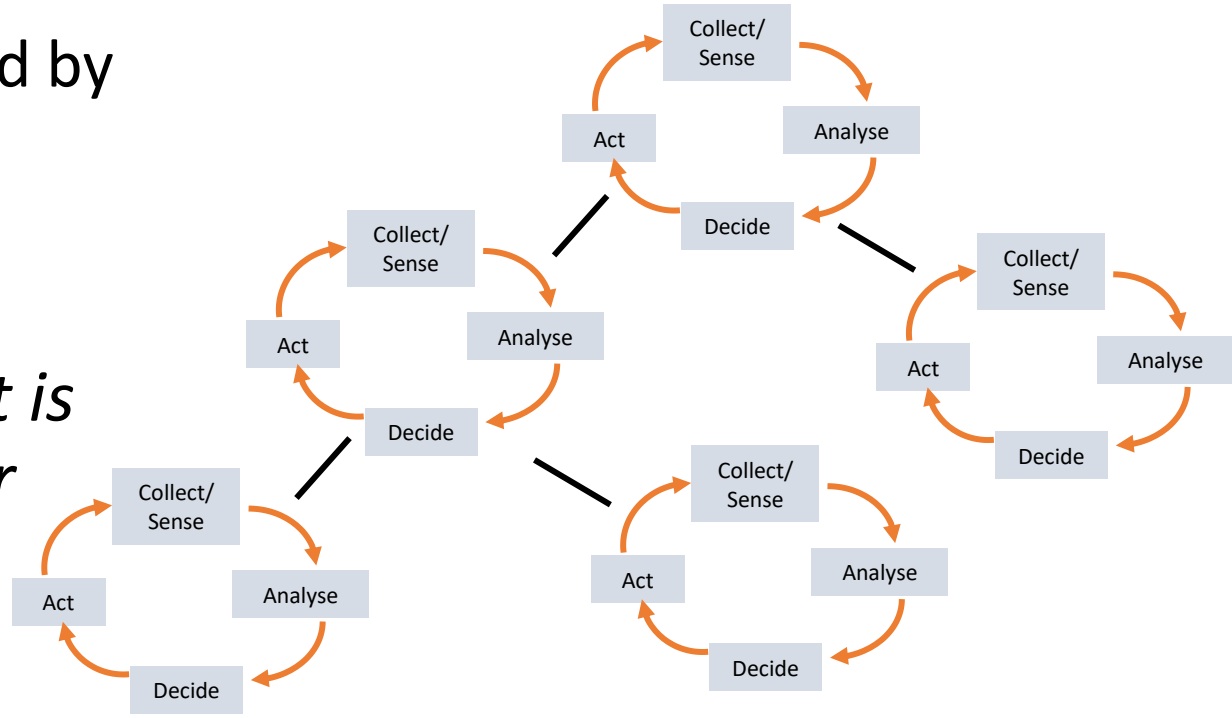
SANS 12-13th December 2022

Problem Context

Complex control systems, such as Mobile Network Architectures, are typically characterised by a generic process monitoring loop

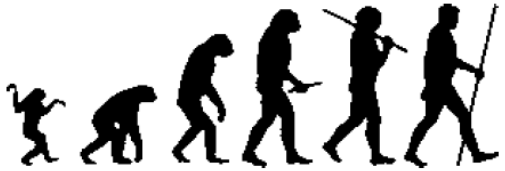


- Systems often too complex to be controlled by one loop
  - Require hierarchies of control loops



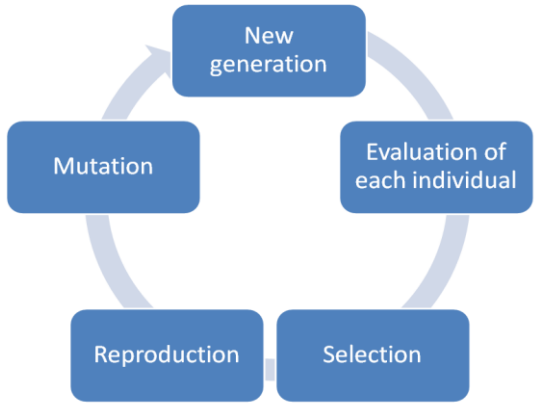
- *How do we design such a hierarchy that is able to respond to change? In particular unforeseen changes?*

- Answer: *Evolution*



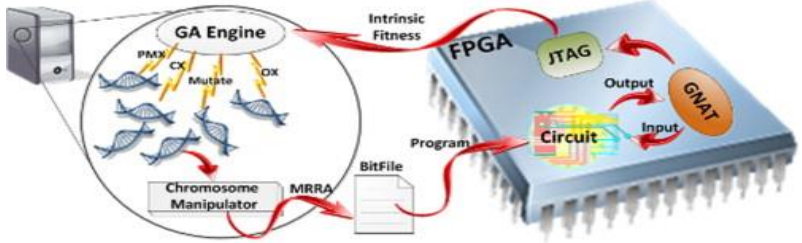
Evolutionary Algorithms

- Population-based metaheuristic optimization algorithms.
- Use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection, and guided by a fitness function
- Large number of variants but all follow similar principles.
- Global optimisation algorithms which are are very effective at finding the best, or close to best, solution in a vast and irregular search space
- Characterised by their abilities to efficiently *explore* the search space, while *exploiting* existing solutions.

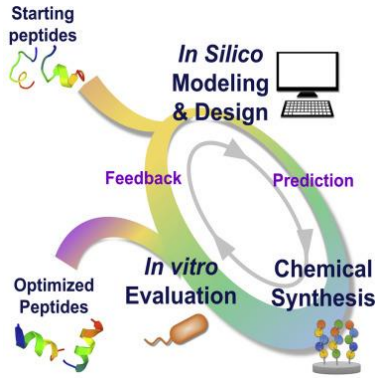


Evolutionary Algorithms

- Employed in a large number of domains:
  - Parameter optimisation
  - Circuit design
  - Code optimisation/refactoring/repair
  - Robotics
  - Financial forecasting
  - Job-shop scheduling
  - Antenna design
  - Drug discovery
  - …




```
template ← Existing valid solution
solution ← Solution being repaired
errorList ← Ordered list of defective genes in solution
missingList ← List of genes missing from solution
for each item in template do:
        if template[item] in missingList then
            repairList.add(template[item])
repairCount ← 0
for each item in solution do:
        if solution[item] in errorList do:
            errorList.remove(item)
            solution[item] ← repairList[repairCount]
            repairCount ← repairCount + 1
```



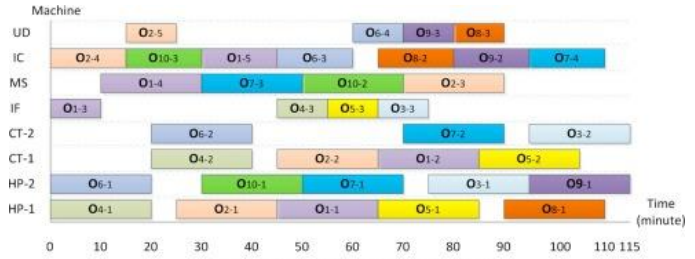
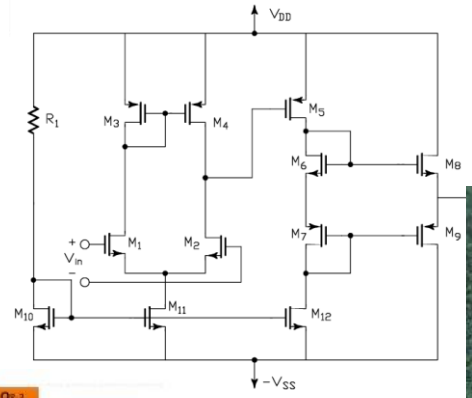
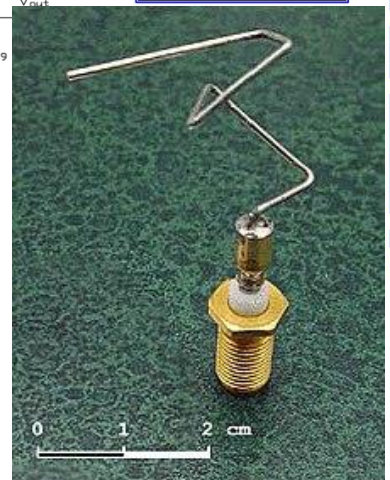


Fig. 9. Gantt chart of the solution for case 1



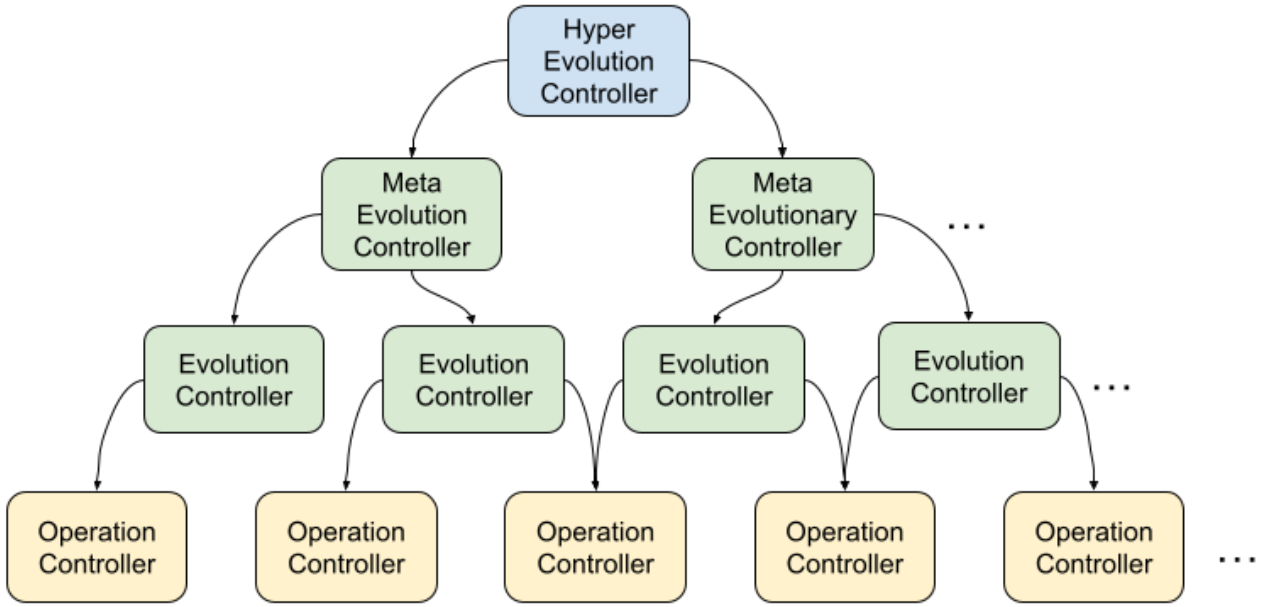


Forecast Performance

Symbol	Forecast June 17th	% Change September 17th	Accuracy
USD/EGP	↑	-2.86%	✗
USD/TRY	↑	10.39%	✓
USD/UAH	↑	5.14%	✓
AUD/JPY	↑	3.63%	✓
AUD/USD	↑	6.22%	✓
AUD/CHF	↑	1.33%	✓
NZD/USD	↑	4.76%	✓
EUR/USD	↑	5.15%	✓
AUD/CAD	↑	3.13%	✓
EUR/JPY	↑	2.68%	✓
AUD/GBP	↑	2.81%	✓
CHF/JPY	↑	2.18%	✓
EUR/GBP	↑	1.97%	✓
GBP/JPY	↑	0.67%	✓
CAD/JPY	↑	3.44%	✓
EUR/CAD	↑	2.35%	✓
AUD/NZD	↑	1.47%	✓
AUD/EUR	↑	0.81%	✓
GBP/USD	↑	3.17%	✓
CAD/CHF	↑	1.74%	✗
USD/INR	↑	-3.76%	✗
CHF/GBP	↑	1.57%	✓
USD/JPY	↑	-2.41%	✗
USD/HKD	↑	0.09%	✓
EUR/RUB	↓	13.75%	✗
NOK/SEK	↓	-0.87%	✓
USD/MYR	↓	3.73%	✓
EUR/SEK	↓	-1.42%	✓
GBP/CHF	↓	1.50%	✓
GBP/CAD	↓	0.76%	✗
USD/SGD	↓	-2.72%	✓
USD/CNY	↓	4.56%	✓
USD/KRW	↓	-3.51%	✓
USD/CHF	↓	-4.56%	✓
USD/MXN	↓	6.54%	✓
USD/CAD	↓	-2.76%	✓
JPY/CNY	↓	2.25%	✓
USD/THB	↓	0.13%	✗
EUR/NOK	↓	0.48%	✓
USD/ILS	↓	-0.95%	✓
EUR/AUD	↓	-0.82%	✓
USD/IDR	↓	5.71%	✗
USD/BRL	↓	-0.07%	✓
GBP/AUD	↓	2.87%	✓
USD/SEK	↓	-6.37%	✓
USD/PLN	↓	-4.81%	✓
USD/DKK	↓	5.14%	✓
USD/CZK	↓	-4.48%	✓
USD/ZAR	↓	6.35%	✓
USD/RUB	↓	7.16%	✗
USD/HUF	↓	-0.74%	✓
USD/ARS	↓	8.15%	✗

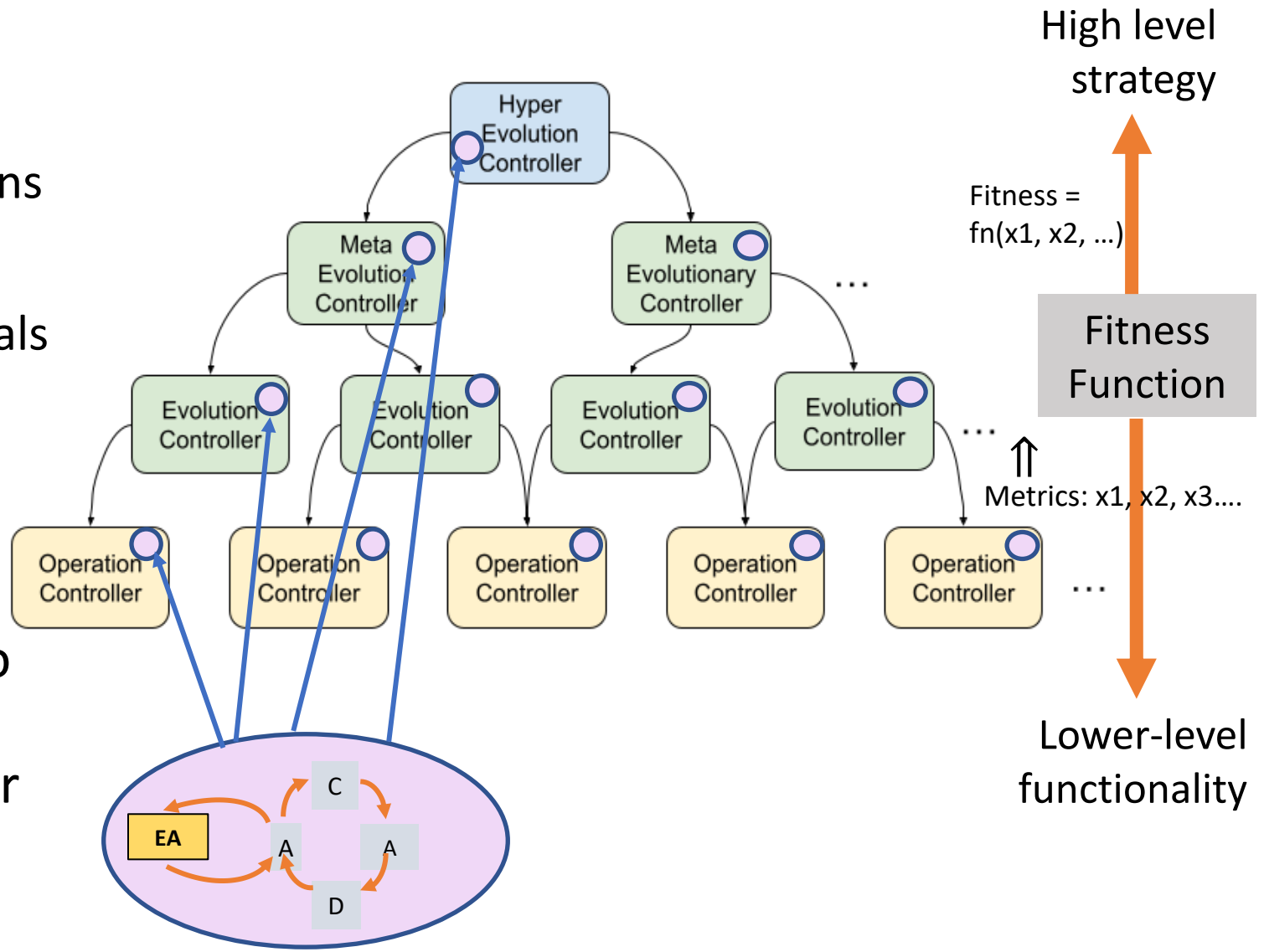
Evolutionary Algorithms in Control Systems

- Each control component in the hierarchy has an EA at its core
- Adapt to changes in environment (e.g. latency, response time, thoughput…)
  - E.g. manipulate parameters of operation controllers
- Allow *new* configurations
  - Different combinations of Operation and Evolution Controllers
- Look for *better* configurations
  - Explore new configurations and compositions of both Operation Controllers and Evolution Controllers

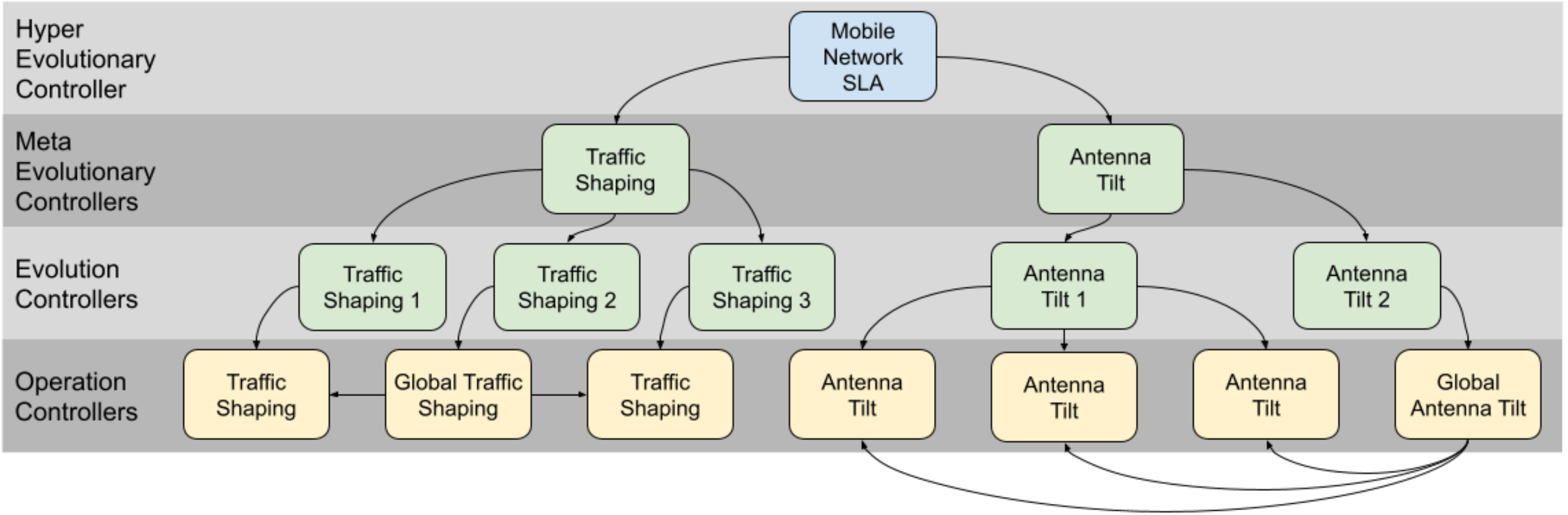


Maintaining order and achieving change

- Driven by the utility/fitness function
  - Mechanism for evaluating solutions
- Hierarchically delegated strategy
  - Higher levels capture strategic goals
  - Lower levels are focused on function performance
  - Strategic goals decomposed into lower-level fitness functions
- Control/adaptation is devolved to lower-level controllers, all powered by EAs and running their own CADA loop



Mobile Network Controller Illustration



Why Evolutionary Algorithms?

- Potential number of configurations to explore in response to a decision to act is huge
  - Which Operation Controllers should be composed?
  - And with what parameters…?
  - Can very efficiently explore this large and irregular search space
- Generic
  - What will future systems look like? What new Operation Controllers might arise? How are changes in strategy accommodated?
  - EAs are generic approaches and can accommodate previously unanticipated changes in system configurations.
- These elements combine to enable EAs to *create* new and effective solutions

Have you Implemented This?

- Funny you should ask…

Questions?