PYCON 2016
ROSE CITY
PORTLAND, OREGON
MAY 28TH - JUNE 5TH

EVOLUTIONARY DESIGN OF PARTICLE SWARMS

Science with Python, Celery, and PyGame

by Benjamin Bengfort

DistrictDataLabs
pycon.districtdatalabs.com

THE STORY OF HOW WE USED PYTHON TO DO SCIENTIFIC EXPERIMENTS

Evolutionary computation is inspired by biological evolution and utilizes metaphors of reproduction, crossover, recombination, mutation, and selection by fitness to perform optimization. Evolutionary techniques are primarily used in academia, but also in industrial design where they are used to optimize the parameters or specifications of physical objects. Of note is a quarter-sized antenna that was designed by evolutionary algorithms (EAs) and whose design was far better than similar human designs (that antenna is now used in a variety of satellites).

The idea that genetic and evolutionary algorithms can be used for computational creativity and design is an interesting one. More specifically, we considered whether or not we could use EAs to design a particle swarm system that outperformed a human designed one. The result was a research project that lasted several months and resulted in a publication – yes, EAs can in fact outperform human design for these types of applications!

However, actually conducting the experimentation required a lot of computation - the result was achieved after evolution of 183 generations that took over 160 hours to complete! The use of Python tools to aid in our scientific investigation was critical, or we would have never had a result. We used Celery to multiprocess the evolution so that 16 simulations were running simultaneously on an m1.2xlarge Amazon EC2 instance. We utilized Numpy and SciPy SciKits to perform vector and matrix computations as efficiently as possible to simulate Newtonian physics, and finally we used Pygame to visualize the simulation and study the results in real time.

In this poster, we will show how Python tools were critical to our research. The clever application of Python to research extends beyond SciPy - it requires multiprocessing, simulation, visualization and more!

Craig Reynolds was originally inspired to design simulated flocks by sitting on a bench watching the behavior of birds flying together.

He designed **BOIDS** (http://www.red3d.com/cwr/boids/) as simple agents whose actions in concert led to globally emergent behavior. In both algorithms and nature, large flocks can move together in complex formations and patterns with only a few simple local rules.

THE EXPERIMENT:

CAN WE USE EVOLUTIONARY ALGORITHMS TO DESIGN A PARTICLE SWARM SYSTEM THAT CAN OUTPERFORM A HUMAN DESIGN?

NumPy

PYTHON FOR SCIENCE

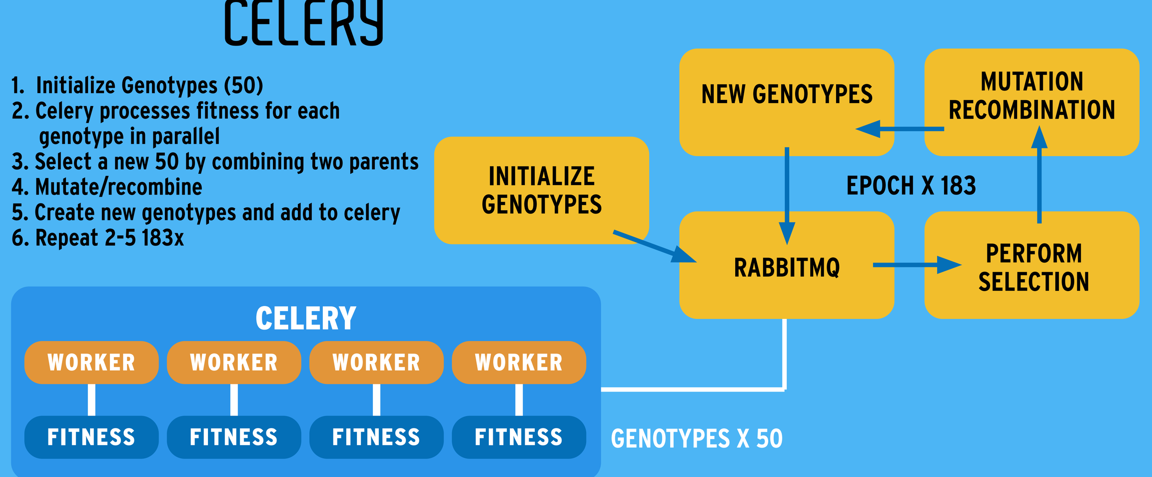
- Used Numpy and SciPy to implement visual and simulated particle swarms and our evolutionary process. (Python with C/Fortran is fast!)
- Used Celery to multiprocess evolution so that each generation was computed in parallel.
- Used PyGame to visualize and analyze the results of the experiments and the behaviors of the swarms.

THIS IS HARD !

- **273 seconds** average simulation (compute fitness)
- Optimization: Celery distributed process queue
  - **tasks**: compute fitness in parallel
  - **controller**: manages population, evolution and submits simulation tasks to the queue
- 8 processes on an Amazon c3.2xlarge:
  - ~160 hours to complete (183 generations total)
  - ~52 minutes per generation

CELERY

1. Initialize Genotypes (50)
2. Celery processes fitness for each genotype in parallel
3. Select a new 50 by combining two parents
4. Mutate/recombine
5. Create new genotypes and add to celery
6. Repeat 2-5 183x



This methodology is about **design** not about real-time problem solving or simulated adaptation.

THE HUMAN TEAM: RODRÍGUEZ, A., & REGGIA, J. A. (2004). EXTENDING SELF-ORGANIZING PARTICLE SYSTEMS TO PROBLEM SOLVING. ARTIFICIAL LIFE, 10(4), 379-395.

THE EVOLVED TEAM: BENGFORT, B., KIM, P. Y., HARRISON, K., & REGGIA, J. A. (2014, DECEMBER). EVOLUTIONARY DESIGN OF SELF-ORGANIZING PARTICLE SYSTEMS FOR COLLECTIVE PROBLEM SOLVING. IN SWARM INTELLIGENCE (SIS), 2014 IEEE SYMPOSIUM ON (PP. 1-8). IEEE.

PARTICLE SWARMS

PROBLEMS IN THE PHYSICAL WORLD CAN BE EFFICIENTLY SOLVED USING SIMPLE AGENTS THAT ACT IN CONCERT TO DEMONSTRATE GLOBALLY EMERGENT INTELLIGENCE

COMPLEX NAVIGATION
SENSOR DEPLOYMENT
FIRE FIGHTING
CONSTRUCTION

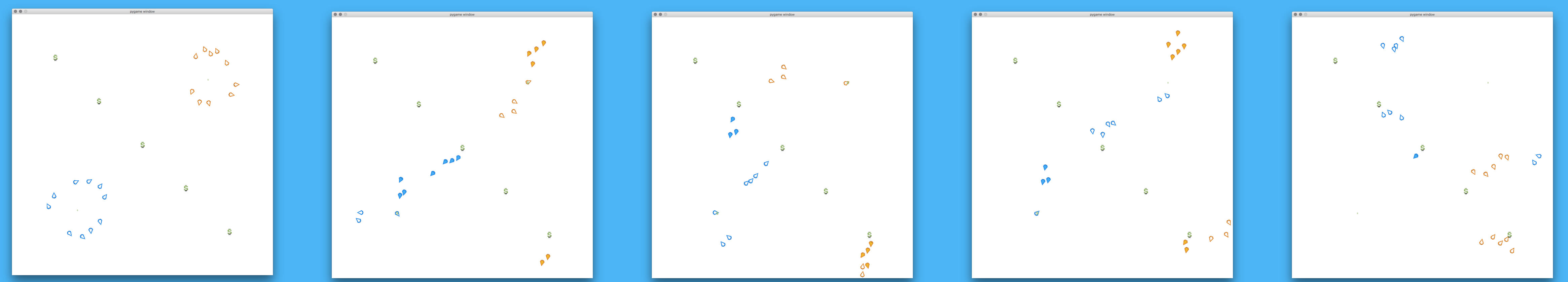
MANY ROBOTIC IMPLEMENTATIONS ARE NOW BECOMING CHEAP ENOUGH TO WARRANT THE USE OF ROBOTIC FLOCKS

OUR EXPERIMENT: SEARCH AND RESCUE

CHARACTERISTICS:
- Two teams and bases
- Resources depots
- Find, collect, and return
- Large world
- Periodic boundaries

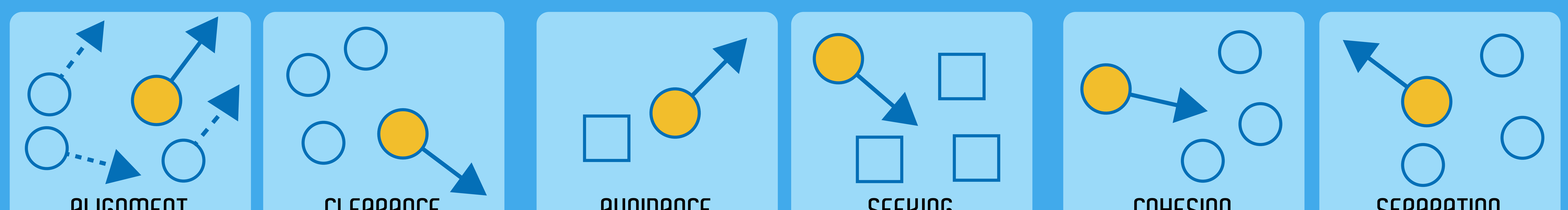
CONSIDERATIONS:
- Exploit vs. Explore?
- Competitive - limited resources
- Theft and collisions
- Economy of design
- Adapt to environment

HUMAN DESIGN: MOVEMENT BEHAVIORS



INITIAL SPREAD · CARAVANS · COLLECT & GUARD · GATHERING & EXPLORING · EXPLORING FLOCKS

BEHAVIOR EMERGES FROM LOCAL RULES



ALIGNMENT · CLEARANCE · AVOIDANCE · SEEKING · COHESION · SEPARATION

EVOLUTION

MODIFY BEHAVIORS OF A SWARM TO FIT A SPECIFIC DOMAIN TASK BY EVOLVING THE UNDERLYING COMPONENTS

- The dynamics parameters of velocity components
- The structure of the finite state machine

EVOLUTIONARY STRATEGIES (ES)
optimization of a genotype of real valued numbers.

EVOLUTIONARY PROGRAMMING (EP)
the evolution of the structure of Finite State Machines (an older definition of evolutionary programming) by modifying transitions and states.