Motivation

Creating an experimental test bed

- Experiment with graphs
  Currently being used by MLStar, another project of ours that performs auto-ML
- Good software engineering
  Avoid spaghetti code from previous versions
- Flexible experimentation
  Inflection points where a user can provide their own behaviors
- Modular implementation
  Components that interact with one another

Framework ≠ Library

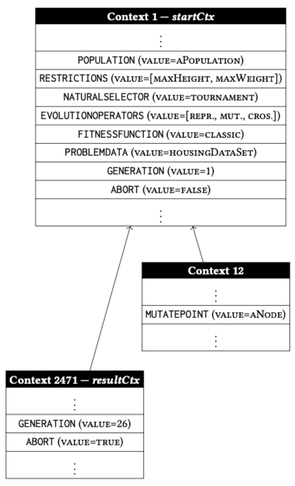
GPStar4 is a framework but also provides a library

- GPStar4 as a framework
  Top level is a loop that performs genetic programming, providing various inflection points
  1. Generate Individuals
  2. Evolve
     a. Select "fittest" individuals
     b. Evolve them into a new population
     c. Continue loop?
  3. Show results
- GPStar4's library
  Provides pre-defined variants of components, e.g.,
  - Individuals as expression trees that can be evaluated
  - Usual evolution operators
  - …

Context System

A state sharing mechanism

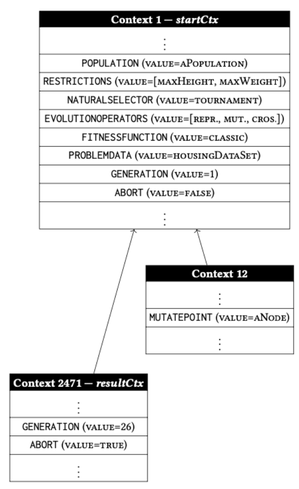
- Stores user-defined behaviors and data
  To be retrieved at inflection points
- Not global
  Local contexts can be used to provide relevant details at localized points



Context System

Implementation — hierarchy

- A context is a map: {tag→value}
- Contexts are hierarchical, usually created by extending another context
  Derived contexts share information from common ancestors
- Lower-level contexts can locally overwrite a parent mapping
  Local state information can then be shared between components
- Parent contexts can be accessed via their *label*



Context System

Implementation — registry

- Tags can be typed
- Tags can have a default value or function
- Value change listeners can be added

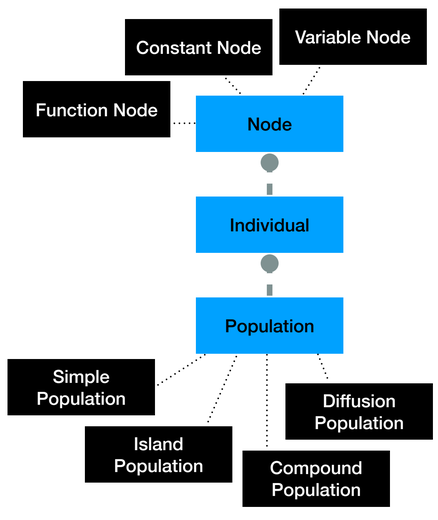
Tag	Type	Default value	Default function	Listeners
BUILDER	Builder	—	—	[]
POPULATION	Pop.	—	—	[]
POPULATIONSIZE	int	—	f1	[]
RESTRICTIONS	Rest.[]	—	—	[]
NATUALSELECTOR	NatSel.	—	—	[]
EVOLUTIONOPERATORS	EvOp.[]	—	—	[]
FITNESSFUNCTION	FitFun.	—	—	[]
PROBLEMDATA	DataSet	—	—	[]
GENERATION	int	0	—	[f2]
ABORT	boolean	false	—	[]
MUTATEPOINT	Node	—	—	[]

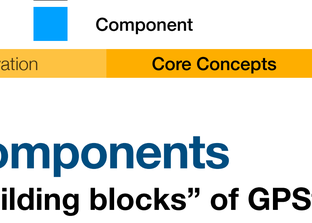
Context Registry

Components

"Building blocks" of GPStar4

- GPStar4 defines a hierarchy from populations to nodes
- Created individuals must conform to the *language*
  Enforced by *restrictions*
  - Max node depth
  - Max node weight
  Enforced on selection of constituent nodes by the *graph builder*

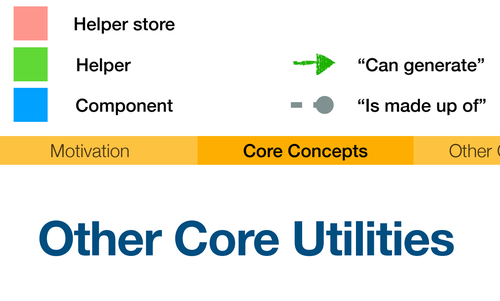


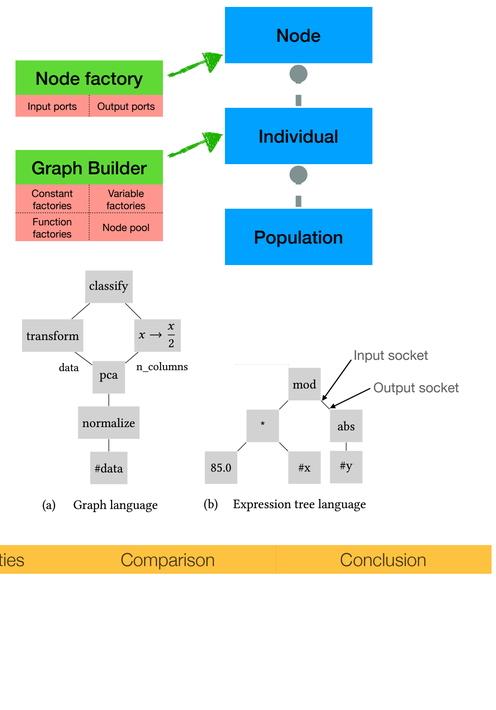


Components

"Building blocks" of GPStar4

- The graph builder coordinates the creation of individuals
  Type-driven
  Holds a set of *factories* that can create various kinds of nodes
  Builds structures of nodes top-down
- Factories create nodes
  They define input and output *ports*
  On instantiation, abstract ports become concrete *sockets*

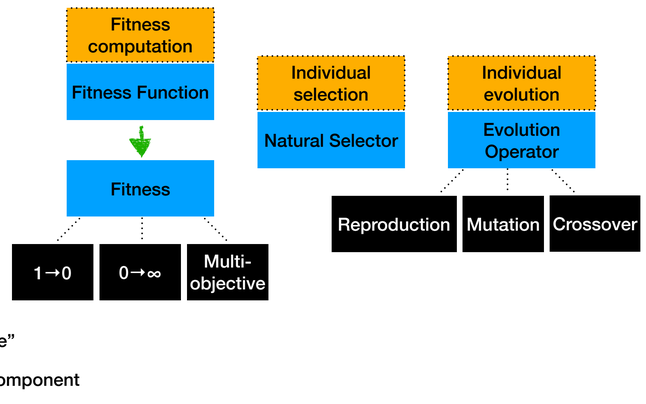




Components

"Building blocks" of GPStar4

- Fitness functions
  Optional
  Fitness allows comparing individuals
- Natural selectors
  Fitness-based
  Competition-based
- Evolution operators



Other Core Utilities

- Choosers
  Pick from a set of options
  - Choosing a node factory with a given type in the graph builder
  - Choosing an evolution operator to apply
  - …
- Caches
  Memoize information
  - Single cache
  - Disjunctive weak cache
- Diversity mechanism
  Includes a variety of pre-defined diversity metrics to apply on a population
  - Number of unique fitness values
  - Keijzer's unique subgraph ratio (num_unique_subgraphs ÷ num_subgraphs)

Comparison

System	GPStar4	ECJ	Jenetics	EC-KitY	deap
Language	Java	Java	Java	Python	Python
Individual representations	tree, DAG	tree, DAG, image	tree, vector	tree, vector	many
Generic interface	✓		✓	✓	
Strongly-typed	✓	✓	✓		✓
Multi-objective		✓	✓	✓	✓
GUI application	✓	✓	(✓)[1]		

[1]for image processing only

Conclusion

Experiments and applications

- A few experiments
  Identifying digits using OCR
  - Used MNIST dataset
  - Required heavily pre-processing images
  - In the end an eigen-classifier performed much better
  Multiple-instance learning
  - Input data consist of bags, each one containing instances (individuals)
  - The challenge is to learn from the instances how to label bags
  - Application: keysets contain keys, and the goal is to learn which keysets allow access to a (nested) room
  Nim
  - A two-player game where players can remove objects from one or more heaps
  - A player wins or loses when they are the one to remove the last object, depending on game configuration
  - Successful strategies are based on integer manipulation
  - Tried different representations for moves: div-mod direct encoding, two-zone bit sequence, scoring of next moves
- Currently being extended by MLStar
  Auto-ML system using GP for supervised learning