Evolutionary Computation with Islands

Extending EvoLP.jl for Parallel Computing

Xavier F. C. Sánchez Díaz Ole Jakob Mengshoel



Norwegian ICT conference for research and education **NIKT2023**



2

- The **operators**
 - The tests
 - Conclusion and future work

Outline

NOTIC CONFERENCE Norwegian ICT conference for research and education NIKT2023

norwegian

- A brief **overview** of evolutionary computation
- Our framework: EvoLP.jl
- The problem and our motivation
- . The **feete**

Evolutionary Computation



It is a branch of **Computational Intelligence** that solves **optimisation** problems using **evolution-inspired** algorithms.

- A **population** of candidate solutions is **evolved iteratively**
- **Exploration**: mutation (stochastic perturbation)
- **Exploitation**: recombination (of promising solutions)



Our Framework



We made EvoLP.jl!

"A playground for evolutionary computation—for prototyping evolutionary solvers such that one can swap a component or two and see how it affects the run"





EvolP.jl

EvoLP.jl

A playground for Evolutionary Computation in Julia



Project repo at https://github.com/ntnu-ai-lab/EvoLP.il



Documentation at https://ntnu-ai-lab.github.io/EvoLP.jl/stable



julia> import Pkg julia> Pkg.add("EvoLP")



Partially funded by NFR

Proudly supported by NAIL





An **open-source** framework to **code** and analyse **evolutionary computation** solvers with:

- Test functions for optimisation
- Result reporting and Statistics logging
- Built-in algorithms and support for custom operators



- Challenge: To find multiple optima in multimodal functions, we need diversity preservation mechanisms.
- Fitness sharing
- Niching
- Clustering
- Speciation



Jacques Descloitres MODIS Land Rapid Response Team NASA GSFC





- Challenge: To find multiple
 optima in multimodal
 functions, we need diversity
 preservation mechanisms.
 - We can preserve diversity by evolving multiple populations and occasionally exchange individuals
 - The Island model of the Genetic Algorithm



Jacques Descloitres MODIS Land Rapid Response Team NASA GSFC





- **Opportunity**: we have **HPC** capabilities
 - Can we **expand EvoLP.jl** to run easily in **HPC clusters**?
 - **Opportunity**: there is **little support** for **parallel** execution in **EC software** in Julia.
 - Evolutionary.jl and Metaheuristics.jl include only distributed evaluation using threads
 - No registered package supports island models



-



Parallel architectures provide many benefits:

- Faster runtime by distributing the workload

- Ease of modelling island approaches

- Perfect match for MPI
- Convergence benefits of the communication
- Communication functions as a **diversity preservation** technique



EvoLP – The Taxonomy



A **block** for each *step* in the evolution process:

- Population Generators
- Selectors
- Recombinators
- Mutators





EvoLP – The Taxonomy



A **block** for each *step* in the evolution process:

- Population Generators
- Selectors
- Recombinators
- Mutators
- Island operators







Copernicus Sentinel (2020), ESA

drift operator

Send individuals from one island to another





NOTIFICATION NOTIF

Copernicus Sentinel (2020), ESA

strand operator

Receive individuals from one another island





Copernicus Sentinel (2020), ESA

reinsert operator

Reinsert individuals into population, replacing some of the old ones according to a given policy



Tests Setup



We **compare islands** vs single population (serial).

For the islands:

- An archipelago with 64 islands
 - 1-way ring topology
 - All islands are identical
- drift selects deme uniformly at random
- reinsert replaces worst deme by the received deme



Tests Setup



- Each population is a **Generational Genetic Algorithm**:
 - **Population size**: 30 individuals
 - Random initialisation
 - Rank based selection of parents
 - Uniform crossover
 - Gaussian **mutation** with $\sigma = 0.1$
 - Limited to 100 generations
 - Tested on Idun using Julia 1.7.2 and EvoLP.jl 1.2



Tests – Ackley

Norwegian ICT conference for research and education **NIKT2023**

20

15

10

5





Parallel

d = 10

Serial

Tests – Rosenbrock









Fewer outliers on parallel approach, but distribution is similar



Tests – Michalewicz







Harder to traverse for d=10, similar distributions but parallel *seems* better



Tests – Eggholder

300

-300

NOTIVE NOTICE AND A CONTRACT AND A CONT



Parallel approach converges more consistently than the serial approach



300

v -300

Tests – Rana

300

0

-300





Parallel approach converges more consistently than the serial approach



300

v -300

Ν

Conclusions



- We studied island models in multimodal functions
- We found island models **provide diversity** and show **convergence** benefits over the serial approach
- Extended EvoLP.jl with new communication blocks:
 - drift
 - strand
 - reinsert



Future Work



- Ship this extension in EvoLP 2.0 (Q1 2024)
- Expand analysis of **diversity preservation** mechanisms in **multimodal landscapes**
- Experimenting with other components:
 - MPI topologies and overheads
 - Other migration selection/replacement policies



EvoLP.jl

Thank you! **Evolutionary Computation with Islands** Extending EvoLP.jl for Parallel Computing

An **open-source** framework to **code** and analyse evolutionary computation solvers with:

- Test functions for optimisation
- **Result reporting** and **Statistics logging**
- Built-in algorithms and support for custom operators



Partially funded by NFR

Proudly supported by NAIL











Project repo at

Documentation at https://ntnu-ai-lab.github.io/EvoLP.jl/stable



julia> import Pkg julia> Pkg.add("EvoLP")

