Concurrency in evolutionary algorithms

JJ Merelo
University of Granada
Granada (Spain)
jmerelo@ugr.es

http://gecco-2019.sigevo.org/

1

Instructors

❖ Juan J. Merelo is Professor of Computer Architecture at the University of Granada. He's also an active open source developer, and as such has released free software libraries for evolutionary algorithms in Perl 6, Perl and JavaScript. He supports open science through extensive use of repositories. His repo can be found at https://github.com/JJ

2

Course Agenda

❖ Why it is now the moment of concurrent algorit languages.

❖ Basic concepts of concurrency.

❖ Functional programming and what it's got to do with concurrent programming.

❖ The concept of state and how it's related to concurrent programming.

❖ Stateless algorithms HOWTO

❖ How to eliminate state from evolutionary algorithms

❖ Levels of concurrency and programming patterns for concurrent evolutionary algorithms

❖ Issues and challenges in concurrent evolutionary algorithms.

3

Concurrency

❖ High-level abstract interface to low-level threac

❖ Several types of concurrency
  • Channel-based
  • Actor-based

❖ Modern languages including concurrency :
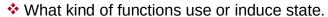 • Perl 6
  • Julia
  • Scala
  • Go

4

Functional programming

❖ Functions as first-class entities.

❖ No side effects → Stateless algorithms

❖ Functional data structures.

❖ Reactive programming: Asynchronous events that *react* to events.

• Messages or events from channels.

5

Working with no state

❖ What kind of functions use or induce state.

❖ Stateless algorithms

❖ Functional equivalence and *ab initio* design of algorithms.

❖ Lambda functions and serverless architectures.

6

Stateless evolutionary algorithms

❖ Granularity of functions.

❖ Escaping functional equivalence

❖ Dataflow design for concurrent evolutionary algorithms.

❖ The importante of communication.

7

Concurrent programming patterns

❖ Thread clusters.

❖ Thread scaling patterns.

❖ Message design for concurrent evolutionary algorithms.

❖ Debugging and logging.

❖ Auto-threading.

8

Implementations of concurrent evolutionary algorithms

- ❖ Implementations in Perl 6 using Algorithm::Evolutionary::Simple
- ❖ Some concurrent libraries that can be used.
- ❖ Cloud-native implementations
  - Use of message brokers.
- ❖ Examples in other languages.

9

References

- ❖ Gregory R Andrews. 1991. Concurrent programming: principles and practice. Benjamin/Cummings Publishing Company San Francisco.
- ❖ John Hawkins and Ali Abdallah. 2001. A Generic Functional Genetic Algorithm. In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA '01). IEEE Computer Society, Washington, DC, USA, 11–. http://dl.acm.org/citation.cfm?id=872017.872197
- ❖ Kittisak Kerdprasop and Nittaya Kerdprasop. 2012. Concurrent Computation for Genetic Algorithms. In Proceedings of the 1st International Conference on SoftwareTechnology. 79–84.
- ❖ J. L. J. Laredo, P. A. Castillo, A. M. Mora, and J. J. Merelo. 2008. Exploring population structures for locally concurrent and massively parallel Evolutionary Algorithms. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). 2605–2612. https://doi.org/10.1109/CEC.2008.4631148

10

References

- ❖ Kiyoharu Tagawa. 2012. Concurrent Differential Evolution Based on Generational Model for Multi-core CPUs. In Proceedings of the 9th International Conference on Simulated Evolution and Learning (SEAL'12). Springer-Verlag, Berlin, Heidelberg, 12–21. https://doi.org/10.1007/978-3-642-34859-4 ₂
- ❖ Juan Julián Merelo Guervós, Juan Luis Jiménez Laredo, Pedro A. Castillo, José Mario García Valdez, Sergio Rojas Galeano:Exploring Concurrent and Stateless Evolutionary Algorithms. EvoApplications 2019: 405-412
- ❖ José Mario García Valdez, Juan Julián Merelo Guervós: A modern, event-based architecture for distributed evolutionary algorithms. GECCO (Companion) 2018: 233-234
- ❖ Juan Julián Merelo Guervós, José Mario García Valdez: Going Stateless in Concurrent Evolutionary Algorithms. WEA (1) 2018: 17-29

11