(Biological) Life-Cycle Algorithm*

First Author[1[0000−1111−2222−3333]], Second Author[2,3[1111−2222−3333−4444]], and Third Author[3[2222−−3333−4444−5555]]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. Several bio-inspired algorithms use population evolution as analogies of nature. In this paper, we followed a strategy that takes a similar approach as the study of bacteria growth in microbiology, where using a microscope, we can observe and analyze the evolution of a population over time. We present an algorithm inspired by the biological life-cycle of animal species, which consists of several stages: birth, growth, reproduction, and death. As in nature, we intend to execute all these stages in parallel and asynchronously on a population that evolves constantly.

In this paper, we present a novel distributed nature-inspired algorithm for solving optimization problems. From the ground up, we designed the algorithm as a cloud-native solution using the cloud available resources to divide the processing work-load, among several computers or running the algorithm as a cloud service. The algorithm works concurrently and asynchronously on a constantly evolving population, using different computers (or containers) independently, eliminating waiting times between processes.

This algorithm seeks to imitate the natural life cycle, where new individuals are born at any moment and mature over time, where they age and suffer mutations throughout their lives. In reproduction, couples match by mutual attraction, where they may have offspring. Death can happen to everyone: from a newborn to an aged adult, where the individual's fitness will impact their longevity. As a proof-of-concept, we implemented the algorithm with Docker containers by solving the OneMax problem comparing it with a basic GA (sequential) algorithm, where it showed favorable and promising results.

Keywords: First keyword · Second keyword · Another keyword.

* Supported by organization x.

1 First Section

1.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], and a homepage [5]. Multiple citations are grouped [1–3], [1, 3–5].

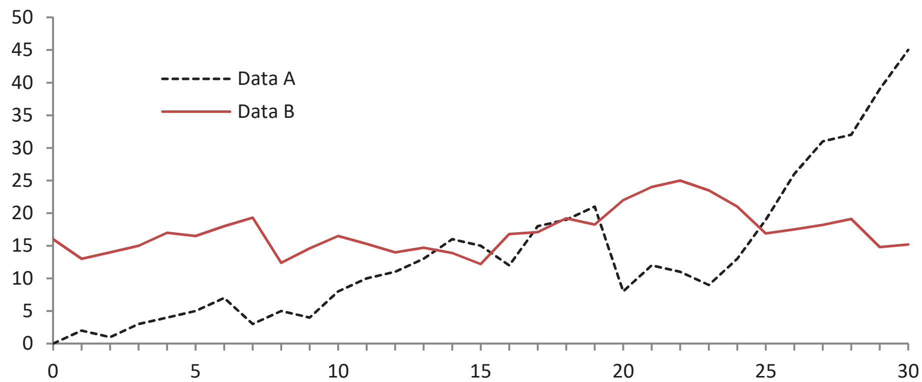

Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

References

1. Valdez, M.G., Guervós, J.J.M.: A container-based cloud-native architecture for the reproducible execution of multi-population optimization algorithms. Future Generation Computer Systems **116**, 234–252 (2021)
2. Valdez, M.G., Guervós, J.J.M.: A container-based cloud-native architecture for the reproducible execution of multi-population optimization algorithms. Future Generation Computer Systems **116**, 234–252 (2021)
3. Valdez, M.G., Guervós, J.J.M.: A container-based cloud-native architecture for the reproducible execution of multi-population optimization algorithms. Future Generation Computer Systems **116**, 234–252 (2021)
4. Valdez, M.G., Guervós, J.J.M.: A container-based cloud-native architecture for the reproducible execution of multi-population optimization algorithms. Future Generation Computer Systems **116**, 234–252 (2021)
5. Valdez, M.G., Guervós, J.J.M.: A container-based cloud-native architecture for the reproducible execution of multi-population optimization algorithms. Future Generation Computer Systems **116**, 234–252 (2021)