Evolutionary Multi-Objective Optimization: A Distributed Computing approach

Rahul Krishna, George Mathew
Computer Science, North Carolina State University, USA
{rkrish11, george2}ncsu.edu

Abstract—
Index Terms—

I. Introduction

Optimization is the task of finding one or more solutions which satisfy one or more specified objectives. While a single-objective optimization involves a single objective function and a single solutions, a multi-objective optimization considers several objectives simultaneously. In such a case, a multi-objective optimizer generates a set of alternate solution with certain trade-offs. These are called Pareto optimal solutions.

Multi-objective problems are usually complex, NP-Hard, and resource intensive. Although exact methods can be used, they consume prohibitively large amounts of time and memory. An alternative approach would be to make use of meta-heuristic algorithms, which approximate the Pareto frontier in a reasonable amount of time. Even so, these meta-heuristic algorithms consume a significant amount of time.

Parallel and distributed computing used in design and implementation of these algorithms may offer significant speed-ups. In addition to this, they may be used to improve the quality, increase the robustness of the the obtained solutions, and may also allow the algorithms to be scaled to solve large problems.

In this project, we aim to present parallel models for two evolutionary multi-objective optimizers: (1) Differential Evolution (DE) [1]; and (2) Geometric Active Learning (GALE) [2]. From the implementation point of view, we focus on using the *henry2 Linux cluster* offered by NC State with distributed programming environments such as message passing (Open-MPI) [3].

This report is organized as follows. The following section presents a brief description of the algorithms being studied. In §**??**, we discuss various methods for evaluating the performance of the parallelized algorithms. In section §**??**, we highlight challenges we expect to overcome during implementation. Finally, section §**??** highlights the tools we use.

II. Methods and Materials

An Evolutionary Optimization(EO) begins its search with a population of solutions usually created at random within a specified lower and upper bound on each variable. If bounds are not supplied in an optimization problem, suitable values can be assumed only for the initialization purpose. Thereafter, the EO procedure enters into an iterative operation of updating the current population to create a new population by the use of four main operators: selection, crossover, mutation and elite-preservation. The operation stops when one or more termination criteria are met. The following evolutionary algorithms shall be parallelized.

Setting	Value
Population Size	100
Number of Generations	100
Mutation Rate	0.75
Crossover Probability	0.3

Fig. 1: Settings for DE

Setting	Value
Population Size	100
Number of Generations	100
Domination Factor	0.15

Fig. 2: Settings for GALE

III. Experimental Setup

IV. Results

References

[1] Rainer Storn and Kenneth Price. Differential evolution &ndash; a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, December 1997.

[2] J. Krall, T. Menzies, and M. Davies. Gale: Geometric active learning for search-based software engineering. *Software Engineering, IEEE Transactions on*, PP(99):1–1, 2015.

[3] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

Algorithm	Runtime	Convergence	Divers
DE	Min:A Med:B Max:C	Min:A Med:B Max:C	100

Fig. 3: Settings for GALE