

Democratic Algorithm: A Novel Approach to Hierarchical Swarm Intelligence

Abstract—Computational intelligence has produced nature inspired algorithms as the new and popular problem solving tool in diverse fields such as soft computing, functional optimization, data science and machine learning. These algorithms can be further sub-divided, with swarm intelligence, physics-inspired, social, evolutionary algorithms a few noteworthy. In this paper we present Democratic Algorithm (DA), a social-construct based algorithm heavily inspired from hierarchy swarm intelligence and real life social establishments. Our proposal extends the existing methodology for hierarchical structures by framing experience into a viable component for future applications. A comprehensive study is conducted with 16 benchmark functions and real-life data to cement the competence of the algorithm with DA providing a markup of 2-7.4X of convergence rate.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

Optimization lies at the very core of majority of the problems ranging from machine learning to business planning. The level of importance awarded to optimization is due to the limited computational resources at hand. To compensate for the lack of resources, smarter algorithms are developed on an almost daily basis. These algorithms are capable of working efficiently in a constrained environment.

A. Multi-Objective Optimization Problem Definition

As the name suggests, Multi-Objective Optimization Problems (MOOPs) require simultaneous solutions to multiple single-objective optimization problems each with its own set of constraints.

$$\begin{aligned} \text{Maximize/Minimize : } f_m(x) \quad & m = 1, 2, \dots, M \\ \text{constraints : } g_j(x) \geq 0 \quad & j = 1, 2, \dots, J \\ h_k(x) = 0 \quad & k = 1, 2, \dots, K \\ x_i^L \leq x_i \leq x_i^U \quad & i = 1, 2, \dots, n. \end{aligned} \quad (1)$$

Equation 1 describes a general MOOP consisting of M single-objective functions having K equality bounds and J inequality bounds [1]. The n input vector components x_i are also bounded within a lower x_i^L and upper x_i^U bound.

Usually MOOPs are solved by decomposing them into multiple single-objective optimization problems. However, the striking difference in this method and directly solving a MOOP lies in the decision space. For a pure MOOP, the multi-dimensional decision space is accompanied with an objective space Z . For each solution x , there is a point in the objective space denoted by $z = (z_1, z_2, z_3, \dots, z_M)$. Hence pure MOOPs optimize a vector whereas multiple single-objective optimization problems optimize singular variables.

B. Orthodox Methods of Solving MOOPs

The techniques developed in this section describe classical techniques which are based on mathematical and statistical reasoning. These algorithms are rather simplistic and quite intuitive. Some of the noteworthy techniques are discussed here:

- **Weighting Sum Method:**

In this algorithm, the MOOP is decomposed into multiple singular problems and each problem is assigned a weight corresponding to its respective importance to the global front. The weights must be quantified based on prior information and the singular objective functions must be normalized for proper setting of the weight vector. Equation 2 decomposes the MOOP $F_m(x)$ into m single-objective optimization problems and weight vector w_m , subject to constraints mentioned in Equation 1.

$$\text{Optimize : } F_m(x) = \sum_{m=1}^M w_m f_m(x) \quad (2)$$

– Advantages:

1) Simple and intuitive

– Disadvantages:

1) Required conversion of minmax problems into 1 type

2) Difficulty in proper weight vector setting

- **Weighted Metric Method:**

This technique is a general extension of weighting sum method with a variant of combination of multiple objectives. Equation 3 represents this method where l_p is the distance measure between candidate solution x and solution z^* , parameter $p \in [1, \infty]$. For $p = 1$, weighted metric decomposes into weighting sum method.

$$\text{Optimize : } l_p(x) = \left(\sum_{m=1}^M w_m |f_m(x) - z_m^*|^p \right)^{\frac{1}{p}} \quad (3)$$

– Advantages:

1) Simple yet generic distance metric

– Disadvantages:

1) Prerequisite knowledge of individual minima and maxima of each function is required

- **Value Function Method:**

In this method, the user must provide a value mapping function $\mathbb{U} : \mathbb{R}^M \rightarrow \mathbb{R}$ for all M functions. The optimization process is then reduced to Equation 4, subject to the same constraints as Equation 1

$$\text{Optimize} : \mathbb{U}(f(x)) \quad (4)$$

- Advantages:
 - 1) Ideal when prior information is available
- Disadvantages:
 - 1) Problem of over-simplified value mapping function

Solving MOOPs using classical methods requires either prior information about the subproblems or assumptions for decomposition. These algorithms have been put to use in multiple practical applications involving huge sets of internal parameters, which incur an overhead of optimal configuration setting.

II. MODERN TECHNIQUES FOR SOLVING MOOPs

In this section, we provide a brief overview of modern techniques which provide a computationally cheaper substitute to classical computation for finding solutions in artificial landscapes.

A. Quantified problems in Classical Computation

The algorithms described in the preceeding section can be classified as either direct search or gradient based search techniques. Direct search algorithms use described constraints are utilized in their raw format to guide the solution. Gradient based search techniques use derivatives of the objective function and constraints to guide the solution. However, both these techniques encounter similar problems as described:

- Result is dependent on selection of initial state
- Tendency to get stuck in local optima
- Lack of generalization due to dependence on problem statement
- Requirement of mathematical representation
- Inefficient when it comes to non-differentiable or discontinuous problems
- Lack of parallelization in majority of the algorithms
- Time incurred is high due to single search guidance system

These problems resulted in the development of new fields in Computational Intelligence, namely Evolutionary Algorithms and Swarm Intelligence. The main motivation for the development of these algorithms is to alleviate the hurdles faced by classical algorithms.

REFERENCES

- [1] Chankong V., Haimes Y., Thadathil J., Zionts S. (1985), Multiple Criteria Optimization: A State of the Art Review, Decision Making with Multiple Objectives, Lecture Notes in Economics and Mathematical Systems 242, Edited by Y.Y. Haimes, V. Chankong, SpringerVerlag, 36.