Democractic 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 it's own set of constraints.

$$\begin{aligned} Maximize/Minimize : f_{\mathrm{m}}(x) \qquad & m = 1, 2 \ldots, M \qquad (1) \\ constraints : g_{\mathrm{j}}(x) \geq 0 \qquad & j = 1, 2 \ldots, J \\ h_{\mathrm{k}}(x) = 0 \qquad & k = 1, 2 \ldots, K \\ {x_{\mathrm{i}}}^{\mathrm{L}} \leq x_{\mathrm{i}} \leq {x_{\mathrm{i}}}^{\mathrm{U}} \qquad & i = 1, 2 \ldots, n. \end{aligned}$$

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 $z = (z_1, z_2, z_3, \ldots, z_M)$. 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 it's 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 $F_{\mathrm{m}}(x)$ into *m* single-objective optimization problems and weight vector w_{m}, subject to constraints mentioned in Equation 1.

$$Maximize/Minimize : F_{\mathrm{m}}(x) = \sum_{m=1}^{M} w_{\mathrm{m}} f_{\mathrm{m}}(x) \qquad (2)$$

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.