**Literature Review**

**Title:** Multifactorial Evolution: Toward Evolutionary Multitasking

**Review of the Literature:**

This paper introduces the use of evolutionary algorithms [1] in multitasking. It uses the inherent parallelism in the evolutionary algorithms to solve multiple optimization tasks across domains. Traditionally, Evolutionary algorithms have been applied to solve two groups of problems. They are 1) Single Objective Optimization (SOO)[2] or 2) Multi objective Optimization[3]. This Paper introduces a third group of problems called as multifactorial optimization. The paper collates the search space of various tasks across problems into a unified representation scheme. This paper assumes that all the tasks are minimization tasks. It uses five properties to define every individual in the population P. The five properties include 1) Factorial cost, 2) Factorial Rank, 3) Scalar Fitness, 4) Skill Factor, 5) Multifactorial Optimality. The optimization is performed in unified search space and the genetic operations [4] such as crossover, mutation and assortative mating is performed. These genetic operations prevent the evolutionary solver to get stuck in local optima. The paper introduces the concept of Multifactorial Evolutionary Algorithm and it does not address scalability which it faces when being deployed in cloud environments.

**Title:** A Group based Approach to improve Multifactorial Evolutionary Algorithm

**Review of the Literature:**

This paper improves the multifactorial Evolutionary Algorithm [5] by introducing grouping. In order to circumvent the challenges associated with the dimensionality where several tasks with multi dimensional search spaces are to be solved simultaneously, similar tasks are grouped, and genetic information is exchanged between groups. This paper uses bisecting K-means algorithm [6] in order to group tasks based on Manhattan distance. The grouping mechanism is evaluated against other multifactorial evolutionary algorithm variants such as Single Objective Evolutionary algorithm(SA), Multifactorial Evolutionary Algorithm without grouping(MFEA), and Multifactorial Evolutionary algorithm with a selection factor(GMF+S). Four benchmark optimization functions [7] are used to evaluate performance. The functions include Griewank function, Ackley Function, Rastrigan Function, Weierstrass Function. The performance evaluation reveals that all the MFEA variants defeat ST after several generations because ST is quickly trapped in a local optimum. GMFEA variants exhibit a better convergence compared to MF, while the performance of GMF+S is between GMF and MF. The grouping mechanism has enabled the MFEA to solve optimization tasks of a higher scale previously unsolvable by plain MFEA. This group-based approach also fails when the optimization tasks are highly variable.

**Title:** Evolutionary Computation: Comments on the History and Current State

**Review of the Literature:**

This paper collates and reviews the works on evolutionary algorithms and defines the genetic operations involved in the evolutionary algorithms. three variants of evolutionary algorithms must be distinguished: genetic algorithms, evolutionary programming, and evolution strategies. From these (“canonical”) approaches innumerable variants have been derived. Their main differences lie in 1) the representation of individuals, 2) the design of the variation operators (mutation and/or recombination), 3) the selection/reproduction mechanism. The paper explores the various representation schemes [8] used to represent data in the evolutionary context. It then moves on to define a genetic operation called as mutation [9], the motivation to employ this operation and explores the various types of mutation operations which can be introduced in the data. It then defines Self- adaptation [10] which involves searching the solution space and strategy parameters simultaneously. The paper defines the Recombination [11] genetic operation and its variants. It is essentially the exchange of genetic information between two individuals in a population in order to create an offspring.

This paper also defines the selection [12] mechanism as a part of the evolutionary algorithm. This has roots in Darwinian evolution where the superior or survivable genes are selected. The paper defines this mechanism in the context of evolutionary algorithms and uses selection to transfer optimal genetic information across generations. This paper explores the theoretical aspects of evolutionary algorithms and summarizes the work in evolutionary algorithms during the time of publication.

**References:**

[1] T. Back, U. Hammel, and H. P. Schwefel, “Evolutionary computation: Comments on the history and current state,” IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 3–17, Apr. 1997

[2] 2] S. M. Guo and C. C. Yang, “Enhancing differential evolution using eigenvector-based crossover operator,” IEEE Trans. Evol. Comput., vol. 19, no. 1, pp. 31–49, Feb. 2014.

[3] C. A. C. Coello, “Evolutionary multi-objective optimization: A historical view of the field,” IEEE Comput. Intell. Mag., vol. 1, no. 1, pp. 28–36, Feb. 2006.

[4] J. Rice, C. R. Cloninger, and T. Reich, “Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models,” Amer. J. Hum. Genet., vol. 30, pp. 618–643, Nov. 1978.

[5] Abhishek Gupta, Yew Soon Ong, and Liang Feng. Multifactorial Evolution: Toward Evolutionary Multitasking. IEEE Transactions on Evolutionary Computation, 20(3):343–357, 2016

[6] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. Proceedings of KDD-2000 workshop on text mining, pages 1–20, 2000

[7] Mei-Ying Cheng, Abhishek Gupta, Yew-Soon Ong, and Zhi-Wei Ni. Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. Engineering Applications of Artificial Intelligence, 64:13–24, 2017.

[8] Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press, 1992.

[9] K. A. De Jong, “An analysis of the behavior of a class of genetic adaptive systems,” Ph.D. dissertation, Univ. of Michigan, Ann Arbor, 1975, Diss. Abstr. Int. 36(10), 5140B, University Microfilms no. 76- 9381.

[10] H.-P. Schwefel, Numerical Optimization of Computer Models. Chichester: Wiley, 1981.

[11] L. J. Eshelman, R. A. Caruna, and J. D. Schaffer, “Biases in the crossover landscape,” in

Proc. 3rd Int. Conf. on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1989, pp. 10–19

[12] J. E. Baker, “Adaptive selection methods for genetic algorithms,” in Proc. 1st Int. Conf. on Genetic Algorithms and Their Applications. Hillsdale, NJ: Lawrence Erlbaum, 1985, pp. 101–111.