A More Efficient MOPSO for Optimization

Walid Elloumi
REGIM: Research Group
on Intelligent Machines,
University of Sfax,
ENIS, BP - 1173
Sfax 3038, Tunisia
Email: elloumiwalid@ieee.org

Adel M. Alimi
REGIM: Research Group
on Intelligent Machines,
University of Sfax,
ENIS, BP - 1173
Sfax 3038, Tunisia
Email: Adel.alimi@ieee.org

***Abstract*—Swarm-inspired optimization has become very popular in recent years. The multiple criteria nature of most real world problems has boosted research on multi-objective algorithms that can tackle such problems effectively, with the computational burden and colonies. Particle Swarm Optimization (PSO) and Ant colony Optimization (ACO) have attracted the interest of researchers due to its simplicity, effectiveness and efficiency in solving optimization problems. We use the notion of multi-objective Particle Swarm Optimization (MOPSO) for few methods; and we find in most of the results; more the number of the swarm increases more the accuracy of object is achieved with greater accuracy. Performance of the basic swarm for small problems with moderate dimensions and searching space is satisfactory.**

Keywords-Swarm intelligence, Particle swarm Optimization, Multi-objective Optimization

I. Introduction

Solving Multi-Objective (MO) problems using Computational Intelligence (CI) techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Artificial Immune Systems (AIS), are a fast developing field of research. Similar to other optimization, MO algorithms using CI techniques.

Problems are examples of a special class of optimization problems called multi-objective optimization problems (MOPs). The question is; what is an optimal solution for a multi-objective problem? In general, it is called a Pareto optimal solution if there exists no other feasible solution [1].

However, users practically need only one solution from the set of optimal trade-off solutions. Therefore, solving MOPs can be seen as the combination of both searching and decision-making [2].

Evolutionary algorithms [3][4] have emerged as heuristic and global alternatives with their most striking characteristic being: using a population for the search in each iteration. This makes them suitable for solving multi-objective problems. Today, the rise of evolutionary multi-objective optimization can be seen by the number of publications produced over time [5]. It is worthwhile to note that there are several paradigms that have emerged as alternatives for the conventional EAs, such as; Particle Swarm Optimization (PSO) [6], Ant Colony Optimization (ACO) [7], Differential Evolution (DE) [8].

A bibliographical survey on methods reveals that various numerical optimization techniques have been employed to approach the problem over the last three decades. Among these methods, priority list methods [9] are very fast; however, they are highly heuristic and generate schedules with relatively higher operation cost.

From the traffic management on a foraging network [10] to the building of efficient structures, along with the dynamic task allocation between workers, examples of complex and sophisticated behaviors are numerous and diverse among social insects [11].

Swarm intelligence, as a scientific discipline including research fields such as; swarm optimization or distributed control in collective robotics, is born from the incredible abilities of social insects to solve their everyday-life problems. Their colonies ranging from few animals to millions of individuals, display fascinating behaviors that combine efficiency with both flexibility and robustness [12].

We are working toward a model that describes peoples' thoughts as a social phenomenon. Thinking differs from the choreographed behaviors of fish and birds in two major ways. First, thinking takes place in a space of many more than three dimensions, as we have seen it in graphs or matrices and high-dimensional analogues of language. Second, when two minds converge on the same point in cognitive space, we call it agreement, not collision.

In flocking simulation the important thing to simulate is coordinated movement of the organisms, whether flocks, herds, or schools. Some motives for studying such a topic include the desire to understand biological aspects of social behavior and the wish to create interesting and lucrative graphical effects.

This paper is organized as follows: the second section gives an overview of accomplishments of the social insects in general. Different problems solving devices inspired by the collective behavior of birds called MOPSO is proposed in section 3. Section 4 provides concise description of the necessary background material, namely the basic MOPSO concepts. In section 5 we detail our approaches tested. Section 6 presents the experimental work. The paper is concluded in section 6.

II. Accomplishments of the Social insects

The optimization potential of simple behaviors has been most noted in studies of insects, and in particular in the behaviors of the social insects. An insect may have only a few hundred brain cells, but insect organizations are capable of architectural marvels, elaborate communication systems, and terrific resistance to the threats of nature.

Wilson [14] began his systematic study of the social behaviors of ants that showed the phenomenon of imprinting in animals: some kinds of baby birds adopt the first thing they see when they hatch as their parents and follow them everywhere they goe. In some famous photographs, Lorenz is seen leading a line of happy goslings who had adopted him as their mother. Imprinting is a form of instinctive behavior Lorenz called a fixed action pattern. A fixed action pattern is a behavior that an organism emits in response to a particular, often very specific, stimulus.

In the introduction of this paper, we note that the term swarm intelligence is originally used to describe a particular paradigm in research.

In fact, all their published models derive from the activities of the social insects. Our use of the term is even less restrictive than [15]. We note, for instance, that the term swarm intelligence has been used in the field of semiotics to describe the kind of irrational buzz of ideas in the mind that underlies the communication of signs between two individuals.

We use the term swarm in a general sense to refer to any such loosely structured collection of interacting agents. The classic example of a swarm is a swarm of bees, but the metaphor of a swarm can be extended to other systems with a similar architecture. An ant colony can be thought as a swarm whose individual agents are ants, a flock of birds is a swarm whose agents are birds, traffic is a swarm of cars, a crowd is a swarm of people, an immune system is a swarm of cells and molecules, and an economy is a swarm of economic agents. Although the notion of a swarm suggests an aspect of collective motion in space, as in the swarm of a flock of birds, we are interested only in types of collective behavior, such as ACO and PSO [16].

III. Multi-objective PSO: MOPSO

This is a Multi-objective Evolutionary Algorithm (MOEA) which incorporates Pareto dominance into a particle swarm optimization algorithm in order to allow the PSO algorithm to handle problems with several objective functions [13]. In PSO, a population of solutions (particles) are used without neither crossover nor mutation operators. Each solution is assigned a velocity and uses this velocity to make a move in the search space. The determination of the velocity of a particle is dependent on both the best position the particle has achieved (the local best) and the best position, the population has found so far (the global best). Applying PSO to multi-objective optimization very much relies on how to define the local and global best positions. MOPSO keeps tracking the local best for every solution over time. In order to find the global best position for each solution, MOPSO uses an external archive (secondary repository) of particles to store all non-dominated particles. Each particle will be assigned for a selected one in the archive (as the global best). The selection of a particle in the archive is dependent on the density of the areas surrounding the particle. Further, the archive is updated continuously and its size is controlled by using the grid technique where a hyper-grid is built in the area of the objective occupied by the archive, and all solutions in the archive will belong to different hyper-cells of the grid depending on their locations.

IV. Basic Multi-objective optimization concepts

Be m inequality constraints. Then, we are interested in finding a solution, $x^* = (x_1^*, x_2^*, ..., x_n^*)$, that minimizes f(x). The objective functions $f_i(x)$ may be conflicting with each other, thereby rendering the detection of a single global minimum at the same point in S, impossible. For this purpose, optimality of a solution in multi-objective problems needs to be redefined properly. Let $u = (u_1, ..., u_k)$ and $v = (v_1, ..., v_k)$ be two vectors of the search space S. Then, u dominates v, if and only if, $u_i \leq vi$. for all; $i = 1, 2, ..., k$, and $u_i < v_i$ for at least one component. A solution, x, of the multi-objective problem is said to be Pareto optimal, if and only if there is no other solution, y, in S such that f(y) dominates f(x). In this case; we also say that x non-dominated with respect to S. The set of all Pareto optimal solutions of a problem is called the Pareto optimal set.

The special nature of multi-objective problems makes necessary the determination of new goals for the optimization procedure, since the detection of a single solution, which is adequate in the single objective case, is not valid in cases of many, possibly conflicting objective functions. Based on the definition of Pareto optimality, the detection of all Pareto optimal solutions is the main goal in multi-objective optimization problems. However, since the Pareto optimal set can be infinite and our computations adhere to time and space limitations, we are compelled to set more realistic goals. Thus, we can state as the main goal of the multi-objective optimization procedure.

V. Approaches tested

We have used two approaches; one consist of combinatory optimization of ACO and PSO and the other of Ant Supervised by PSO.

A. Combinatory Optimization

The combination between Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). This phenomenon is called combinatory optimization. It is a multi-objective function that has the meaning of looking at the initial values aiming to minimize the outcome of our function. We will give an overview on the status of the theoretical analysis of ACO algorithms, with a special focus on the analytical investigation of the runtime required to find an optimal solution to a given combinatorial optimization problem.

PSO, on the other hand constitutes a crucial element to serve the purpose of our study. It includes mainly the algorithm variations. On the next stage PSO will be coupled with ACO to achieve the combinatory optimization. A key consideration in our research is the Travelling Salesperson Problem (TSP). A closer examination of this concept will be given throughout our study. To reach our aim we have to carry out different steps. First, a general framework of the system of nodes that stands for different functions and the set of the nodes translating these functions and constituting entire charts representing localizations. Second, to study the execution time and the number of cycles established. Finally, the ultimate objective function which is to minimize the distances of the paths that relate [16].

We have used 1000 iteration because when we tested 2000 and 3000 iterations we conclude that the cycle number and time are proportionate to that of 1000 iterations. Due to this reason, our combination consists of using 1000 iterations for the sake of time consuming.

The table I resumes the results about the combination and comparison between ACO and PSO.

TABLE I
Combination between ACO and PSO

Number of nodes	Time for Our approach	Time for ACO
22	37,47	44,93
29	2,84	3,05
48	294,77	318,04
52	478,02	479,01
70	764,98	952,54
76	210,31	251,46
96	1678,75	1846,62
150	772,46	1485,50

1) The time for our approach of PSO is better than the ACO for the route of the shortest path to the problem of TSP; therefore we can say we won at :
   - Time
   - Number of cycle
   - Cost
   - Work Space
   - Energy
2) It is certain that our approach of PSO is better than the ACO because the behaviour of PSO is socio-cognitive while the ACO is self-organized.

B. Hybridation of both techniques

In this work, we propose an Ant colony algorithms supervised by Particle Swarm Optimization to solve continuous optimization problems. Traditional ACO are used for discrete optimization while PSO is for continuous optimization problems. Separately, PSO and ACO shown great potential in solving a wide range of optimization problems. Aimed at solving continuous problems effectively, we develop a novel ant algorithm ”Ant Supervised by PSO” (A.S.PSO) the proposed algorithm can reduce the probability of being trapped in local optima and enhance the global search capability and accuracy. An elitist strategy is also employed to reserve the most valuable points. Pheromone deposit by the ants' mechanisms would be used by the PSO as a weight of its particles ensuring a better global search strategy. By using the A.S.PSO design method, ants supervised by PSO in the feasible domain can explore their chosen regions rapidly and efficiently.

Our proposal is to make PSO supervising an ant optimizer. We propose an Ant colony algorithms supervised by Particle Swarm Optimization to solve continuous optimization problems. Traditional ACO are used for discrete optimization while PSO is for continuous optimization problems. Separately, PSO and ACO shown great potential in solving a wide range of optimization problems.

First, we start by initializing the search in light of research in space, the number of iterations for PSO and the cost function. The second step is to initialize the ant, the cost of this function and assimilate an ant for each PSO particle. Thereafter, we proceed to simulate the ant algorithm and integrate the best solution to the ant than the PSO. Finally, we will evaluate the best local solutions and the best position of the PSO. The last step is to change the speed and position of particles PSO, and the positions that of Ant. If conditions are satisfied we reach the end, if not we go back to the third stage.

Since walk has symmetry characteristics, we assign sub-Ant, respectively to the left foot and to the right one. Both must cooperate to ensure a generation of the joints trajectories ensuring a stable walking. The joins considered here are the hip, the knee and the ankle joints [17].

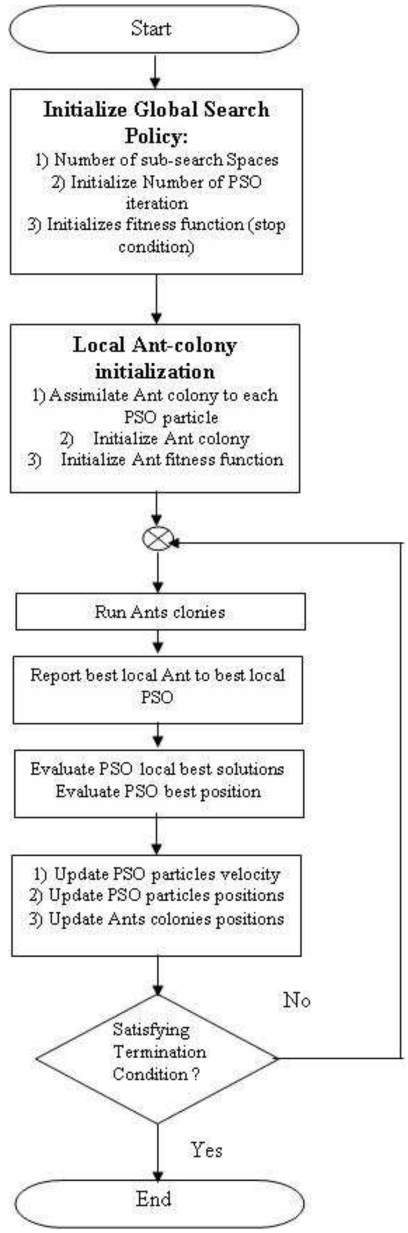

Fig. 1. Procedure A.S.PSO

VI. Simulation results

It is clear that the search ability of a particle is heavily dependent on the best position $P^i_{j,t}$ and $P^{i,g}_{j,t}$, involved in its velocity update equation.

$$V^i_{j,t+1} = WV^i_{j,t} + C_1R_1(P^i_{j,t} - X^i_{j,t}) + C_2R_2(P^{i,g}_{j,t} - X^i_{j,t}) \quad (1)$$

$$X^i_{j,t+1} = X^i_{j,t} + V^i_{j,t+1} \quad (2)$$

Where j = 1,...,n, i = 1,...,N, and are two positive constants, and are random values in range [0, 1].

- W is the called **inertia weight** of the particle. This is employed to control the impact of previous history of velocities on the current velocity, thus to influence he trade-off between global and local exploration abilities of the particles. A larger inertia weight w facilitates global exploration while a smaller inertia weight tends to facilitate global exploration while a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. Suitable selection of the inertia weight w can provide a balance between global and local exploration abilities requiring less iteration for finding the optimum on average. A nonzero inertia weight introduces.

  The preference for the particle to continue moving in the same direction as in the previous iteration.

- C_1R_1 and C_2R_2 are called control parameters. These two **control parameters** determine the type of trajectory the particle travels. If R_1 and R_2 are 0.0, it is obvious that v = v + 0 and x = x + v (for w = 1). It means the particles move linearly. If they are set to very small values, the trajectory of x rises and falls slowly over time.

- $P^{i,g}_t$is the position of the global best particle in the population, which guides the particles to move towards the optimum, The important part in MOPSO is to determine the best global particle $P^{i,g}_t$ for each particle i of the population. In single-objective PSO, the global best particle is determined easily by selecting the particle that has the best position. But, in MOPSO $P^{i,g}_t$, must be selected from the updated set of non-dominated solutions stored in the archive A_{t+1}. Selecting the best local guide is achieved in the function *FindBestGlobal* (A_{t+1}, X^i_t) for each particle i.

- P^i_t is the best position that particle i could find so far. This is like a memory for the particle i and keeps the non-dominated (best) position of the particle by comparing the new position X^i_{t+1} in the objective space with P^i_t (P^i_t is the last non-dominated (best) position of the particle i).

These best positions attract the particle, biasing its movement towards the search space regions they lie, with p_i representing the inherent knowledge accumulated by the particle during its search, while p_g is the socially communicated information of its neighborhood, determined through the adopted neighbourhood topology.

In multi-objective problems, we can distinguish two fundamental approaches for designing PSO algorithms. The first approach consists of algorithms that consider each objective function separately. In these approaches, each particle is evaluated only for one objective function at a time, and the determination of the best positions is performed similarly to the single objective optimization case. The main challenge in such cases is the proper manipulation of the information coming from each objective function in order to guide the particles towards Pareto optimal solutions. The second approach consists of algorithms that evaluate all objective functions for each particle, and, based on the concept of Pareto optimality, they produce non-dominated best positions that are used to guide the particles. In these approaches, the determination of leaders is not straight forward, since there can be many non-dominated solutions in the neighborhood of a particle, but only one is usually selected to participate in the velocity update.

In the aforementioned approaches, the problem of maintaining the detected Pareto optimal solutions must be addressed. The most trivial solution would be to store non-dominated solutions as the particle's best positions. However, this choice is not always valid, since the desirable size of the Pareto front may exceed the swarm size. Moreover, two non-dominated solutions are equally good, arising questions regarding the selection of the one that will be used as the best position of a particle.

The size problem can be addressed by using an additional set, called the external archive, for storing the non-dominated solutions discovered during search, while the problem of selection of the most proper archive member depends on the approach. Nevertheless, an external archive has also bounded size, thereby making unavoidable the imposition of rules regarding the replacement of existing solutions with new ones.

The general multi-objective PSO scheme can be described with the following pseudo code:

```
Begin
            Initialize swarm, velocities and best positions
            Initialize external archive (initially empty)
            While (stopping criterion not satisfied) do
               For each particle
                 Select a member of external archive
                 Update velocity and position
                 Evaluate new position
                 Update best position and external archive
               End for
            End While
  End
```

It is clear that selection of a member of the external archive, as well as the update of archive and best positions, constitute key concepts in the development of multi-objective PSO approaches, albeit not the only ones. Diversity also affects significantly the performance of the algorithm, since its loss can result in convergence of the swarm to a single solution.

The problem of selecting members from the external archive has been addressed through the determination of measures that assess the quality of each archive member, based on density estimators. Using such measures, archive members that promote diversity can be selected. The most commonly used density estimators are the Nearest Neighbor Density Estimator [18] and the Kernel Density Estimator [19].

The problem of updating the archive is more complex. A new solution is included in the archive if it is non-dominated by the new solution, then they are usually deleted from the archive. The necessity for a bounded archive size originates from its tendency to increase significantly within a small number of algorithm iterations, rendering domination check computationally expensive.

Also, the user must decide for the action taken in the case of a candidate new solution that is non-dominated by all members of a full archive. Obviously, this solution must compete all members of the archive in order to replace an existing member. Diversity is again the fundamental criterion, that is, the decision between an existing and a new solution is taken such us; the archive retains the maximum possible diversity.

For this purpose, different clustering techniques have been proposed [20]. A similar approach uses the concept of ϵ-dominance to separate the Pareto front in boxes and retain one solution for each box. This approach has been shown to be more efficient than simple clustering techniques.

The update of each particle's own best position is more straightforward. Thus, in approaches based on distinct evaluation of each objective function, it is performed as in standard PSO single-objective optimization. On the other hand, in Pareto-based approaches, the best position of a particle is replaced only by a new one that dominates it. If the candidate and the existing best position are non dominated, then the old one is usually replaced in order to promote swarm diversity. At this point, we must also mention the effect of the employed neighborhood topology and PSO variant, on the performance of the algorithm. However, there are no extensive investigations to support the superiority of specific variants and topologies in multi-objective cases.

The MOPSO algorithms is coded in JAVA and run on an Intel(R)P(R) 1.7 GHz PC with 512MB memory. There are fewer parameters used for the MOPSO algorithms and they are as follows: The size of the population (swarm) is the number of locations are set as (20, 50,100) the social and cognitive probabilities, c1 and c2, are set as c1=c2=2 for CPSO. Inertia weight (w) is taken as 0,9, the maximum of velocity (v) is taken as 100 and dimension of space as 10 for MOPSO. Each function choice run is conducted for 10 replications and 1000 iterations. The Best fitness and Variance are provided in table II.

TABLE II
APPLIED MOPSO FOR FEW FUNCTIONS

Name of function	Best fitness	Variance
Sphere	0.64516896	1.4250265E-16
	2.291355E-10	4.439057E-23
	7.2057063E-32	5.6994934E-38
Rosenbrock	549.37256	1.0641416E-12
	5.681832	8.019663E-12
	0.0062124003	2.3051954E-13
Rastrigrin	55.96876	1.339896E-13
	16.91429	1.48025E-15
	31.83862	2.306803E-15
Griewank	1.0001535	2.4825341E-17
	1.0	7.993606E-17
	1.0	4.5936946E-17
DejongF4	0.0022106753	2.070291E-14
	2.6852278E-14	2.6434333E-29
	1.4503539E-29	4.2898922E-38
Ackely	2.5801415	4.2826412E-16
	2.3168497	4.5514267E-16
	2.0133152	7.3262606E-16

We use the notion of MOPSO for a few methods, and we find in most of the results more the nunmber of the swarm increases more the accuracy of object is achieved with greater accuracy. Sometimes, if we reach our goals, it should not increase the number of swarms and run again for not losing the accuracy of solutions obtained previously.

VII. Conclusion

This paper gives an overview of different aspects multi-objective optimization. Issues related to the operation of PSO in multi-objective environments have been pointed and a plethora of approaches with various characteristics have been exposed. Since multi-objective optimization is intimately related to the real life applications, efficiency must be the key issue in the development of new multi-objective PSO approaches. The experimental results demonstrate how the performance of the method has improved over the development steps. Finally, the development of multi-objective PSO approaches is currently and will remain a very active and exciting research field.

Acknowledgment

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

We would like to extend our thanks to Ms. A. Ayadi from the Engineering School of Sfax-Tunisia, for her help in English language.

REFERENCES

[1] C. Coello, “Evolutionary multiobjective optimization: A historical view of the field,” IEEE Computational Intelligence Magazine,1(1), pp. 28–36, 2006.

[2] J. Horn, “Multicriteria decision making,” In T. Back, D. B. Gogel and Z. Michalewicz (Eds.)Handbook of evolutionary computation. Institute of Physics Publishing, 1997.

[3] T. Back, “Evolutionary algorithms in theory and practice,” New York: Oxford University Press, 1996.

[4] Z. Michalewicz, “Genetic algorithms + data structures = evolution programs,” (3rd ed) London: Springer-Verlag, 1996.

[5] C. Coello, “EMOO repository,” Retrieved February 5, 2008 from, http://delta.cs.cinvestav.mx/ ccoello/EMOO/, 2006.

[6] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” In Proceedings of IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948, 1995.

[7] M. Dorigo and T. Stutzle, “Ant colony optimization,” MIT Press, 2004.

[8] K. Price, R. Storn and J. Lampinen, “Differential evolution - A practical approach to global optimization,” Berlin, Germany: Springer, 2005.

[9] T. Senjyu, K. Shimabukuro, K. Uezato, and T. Funabashi. “A fast technique for unit commitment problem by extended priority list,” IEEE Trans. Power Syst, 18(2): pp. 882–888, 2003.

[10] M. Burd, “Ecological consequences of traffic organisation in ant societies,” Physica A: Statistical and Theoretical Physics, 372(1): pp. 124–131, 2006.

[11] C. Detrain, and J. L. Deneubourg. “Self-organized structures in a superorganism: do ants ”behave” like molecules,” Physics of Life Reviews, 3(3): pp. 162–187, 2006.

[12] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. “Selforganization in biological systems,” Princeton: Princeton University Press, 2001.

[13] C. Coello, G. T.Pulido and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” IEEE Transactions one Evolutionary Computation, 8(3), pp. 256–279, 2004.

[14] E. O.Wilson and his contributions to ant systematics, “Advances in ant systematics (Hymenoptera: Formicidae): homage to E. O,” Wilson - 50 years of contributions. Memoirs of the American Entomological Institute, 80, pp. 3–7,2007.

[15] E. Bonabeau, and G. Theraulaz, “L’intelligence en essaim,” Pour la Science, 271: pp. 66–73, 2000.

[16] W. Elloumi, and A. M. Alimi, “Combinatory Optimization of ACO and PSO,” Proc. Int. Conf. on Metaheuristics and Nature Inspired Computing META, pp. 8, 2008.

[17] W. Elloumi, N. Rokbani and A. M. Alimi, “Ant Supervised By PSO,” Porc. 4th International Symposium on Computational Intelligence and Intelligent Informatics ISCIII, pp. 161–166, 2009.

[18] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II”, IEEE trans. Evol. Comput., 6(2), pp. 182–197, 2002.

[19] K. Deb and D. E. Goldberg, “An investigation of niche and species formation in genetic function optimization”, In proceedings of the 3rd International Conference on Genetic Algorithms pp. 42–50, 1989.

[20] J. D. Knowels and D. W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy”, Evolutionary computation, 8(2), pp. 149–172, 2000.

[21] S. Mostaghim and J. Teich, “The role of ϵ-dominance in multi objective particle swarm optimization methods”, In proceedings of the IEEE 2003 on Congress on Evolutionary Computation pp. 1764–1771, 2003.