International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988, Volume 5 (2013) pp. 704-713

Measuring Performance of Algorithms in Dynamic Environement

Ramoza Ahsan, Waseem Shahzad

[1] National University of Computer and Emerging Sciences,
Computer Science Department, Sector H-11/4, Islamabad, Pakistan
ramoza.ahsan@nu.edu.pk

[2] National University of Computer and Emerging Sciences,
Computer Science Department, Sector H-11/4, Islamabad, Pakistan
waseem.shahzad@nu.edu.pk

Abstract—— **this paper presents the metrics used for measuring performance of optimization algorithms in dynamic environment. Real world is dynamic in nature so problems in it are mostly dynamic where the optimal solution changes over time. Algorithms proposed for solving such problems must be able to adapt change and produce enough diversity to locate the new or changed optimal solution. Evolutionary algorithms are more suitable to find solutions to dynamic problems as compared to Classical Optimization techniques which sequentially search for the solution and are based on differential equations. Evolutionary algorithms evolve with each generation and can find multiple optima in parallel. Different performance metrics have been used to measure the performance of multi modal and multi objective optimization algorithms in static environment. However, performance measurement in dynamic environment is difficult and complex as traditional performance metrics of static environment like mean fitness, best current fitness, and speed of convergence are not relevant in dynamic environment as fitness of the optima may change over time. A number of techniques have been proposed to find solutions to multi-modal and multi objective problems in dynamic environment. The paper explores the common performance metrics and experimental framework used to measure the performance of algorithms in dynamic environment.**

Keywords- ***Dynamic Environment, Evolutionary Algorithms (EA) Multimodal functions, Multi Objective, Performance metrics.***

I. INTRODUCTION

The world human beings are living is constantly changing whether it is weather, environment, business strategies or daily routines of a day. Changes in the real world problem can occur due to any reason that may include change in customer priorities, change in classes in universities, unpredicted natural disasters or phenomenon etc. in any case when an environment changes whether that change is minimal or devastating, the algorithm which has to find the optimal solution must adapt to the change, take necessary measures and then find the new and changed optimum solution.

Algorithms in the beginning were of the era of classical optimization (CO) techniques which were used to find solutions to most mathematical problems. They dealt with finding a solution to problems that were continuous or finding solutions to differentiable functions. These classical optimization techniques use mechanisms of differential calculus to find optimal solution which are analytical in nature. CO techniques can find the solution of linear, quadratic or specialized problems [1] in more efficient manner.

However in most practical cases in real world, objective function might not be continuous or constant. Thus classical optimization techniques are not appropriate for these types of problems and for them evolutionary algorithms (EAs) were introduced. EAs are inspired from the natural evolution process where the solution is evolved in iteration or steps and final optimal solution is achieved. They are probabilistic and stochastic in nature where in each iteration new solutions are evolved from which the fittest solution is carried on to next generations. They are more suitable for problems that are multi-modal i.e. have more than one optimum solution, or discontinuous or have an element of noise in them. Example of evolutionary algorithms are Genetic Algorithms (GAs) which are well suited for combinatorial optimization problems, genetic Programming (GP) which are used to evolve computer programs, Evolutionary programming (EP) which uses the development of behavioral models instead of genetic models like GAs etc.

When the optima changes in dynamic environment CO techniques have to restart where there is no guarantee that it will find the solution. The search process done until the time a change occurs also gets lost. On the other hand EA may not be restarted again and they can adapt easily to changed conditions. Many problems in Artificial Intelligence (AI) focus on search and optimization techniques, where the goal is to search intelligently through many possible solutions. Simple exhaustive searches are hardly ever applicable to actual and genuine world problems and the complexity grows exponentially. This results in a search that is either too slow or unable to find the required solution.

Optimization is the process of finding solution to a problem where purpose is to either maximize or minimize the objective function and also satisfy the constraints of the variables involved in it. Optimization problems can be divided into 3 broad categories [2].

1. **Uni-modal:**
   Uni-modal are the problems which have a single best solution. They have one global optima and no local optimum exists in such functions. It has a single maximum or minimum within a specified range. The task of uni-modal optimization is to find this "the best solution". Both CO and EAs are used to find the solution of uni-modal problems. Uni-modal functions if plotted have single peak in their graphs. Uni-modal functions can be continuous functions as illustrated in figure 1 or can be discontinuous as described in figure 2.

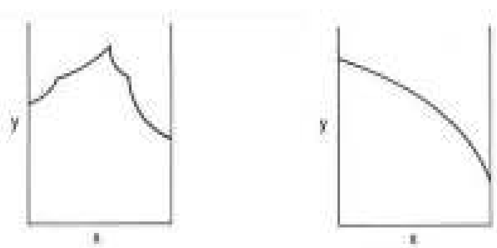

Figure 1: Continuous Uni-modal function

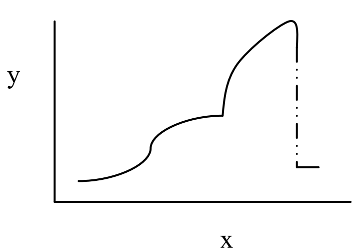

Figure 2; Discontinuous uni-modal function

They can be monotonically increasing or decreasing functions as illustrated in figure 3 and 4 respectively.

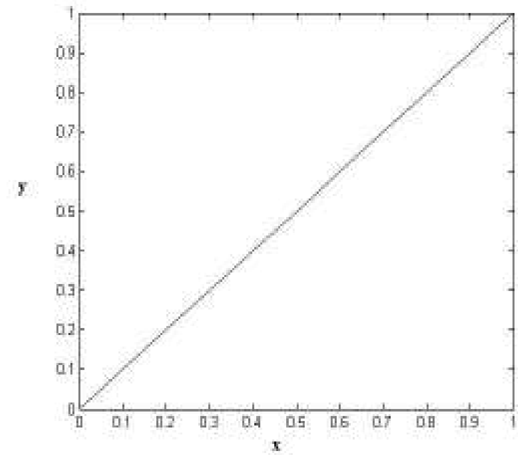

Figure 3: Monotonically increasing uni-modal function (y=f(x))

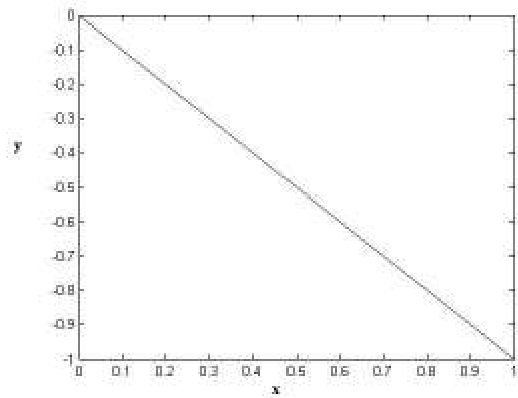

Figure 4: Monotonically decreasing function (y=f(x))

Unimodal optimization search techniques can work in multimodal functions, however they will locate only one optimal solution and not multiple.

2. **Multimodal:**

   In multimodal functions more than one optimal solution exists. Optimal solution can consists of all global good solutions as described in figure 5.

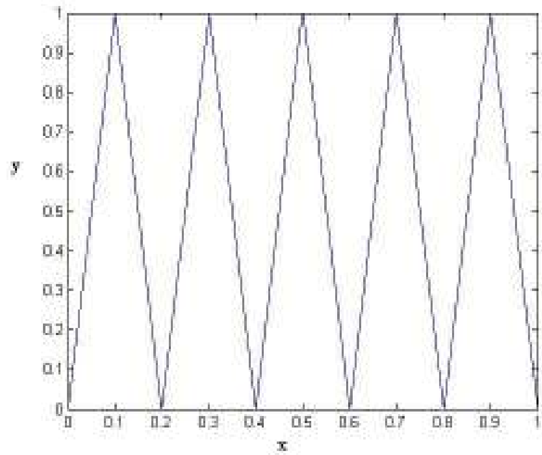

Figure 5: Multimodal function y=sin^6(5πx)

However multimodal function solution can be combination of both global good solution and local optimum as illustrated in following figure.

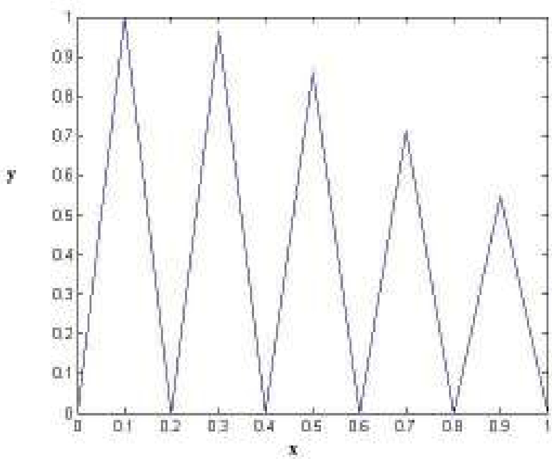

Figure 6: Multimodal function $y = \exp\left(-2\log(2).\left(\frac{x-0.1}{0.8}\right)^2\right) sin^6(5\pi x)$

In multimodal optimization, algorithm must first locate all the global optima before locating the local ones so that it cannot get stuck in local ones.

3. **Multi-objective:**

When an optimization problem involves only one objective function it is called single objective optimization. For example given height and weight objective function is minimization of Heart Attack. However, problems where two or more conflicting objective functions need to be minimized or maximized simultaneously is called Multi-objective optimization. Example of such problems is given a cylindrical with height and diameter, maximizes volume while at the same time minimizing the cost of it. Or in a structural design, it is desired that the structure should be both light and rigid which conflict with each other. Multi objective Optimization algorithm task is to find Pareto optima solutions which are set of solutions that cannot be improved upon one criterion without affecting the criteria of other. Thus the goal of such algorithms is to find the solutions as close as possible to Pareto Optimal front and the solutions should be as diverse as possible. Example of it is illustrated in following figure where multiple solutions are compared with Pareto optima front and the solutions near the front curve and are non-dominating are maintained in an archive.

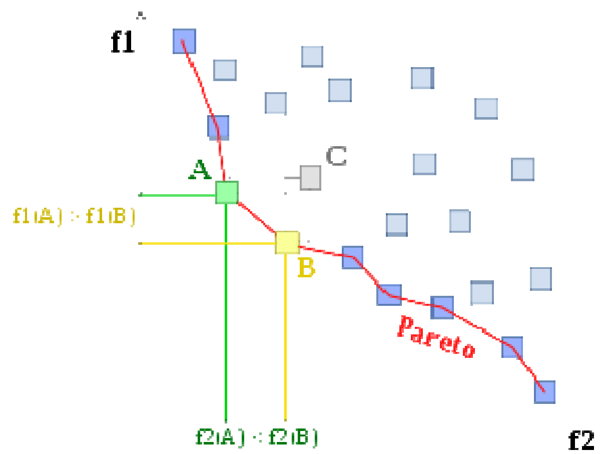

Figure 7: Multi Objective example. Pareto Optima Front

Pareto based techniques use the dominance concept to find the optimal solution. A solution Y dominates solution Z if it is better than solution Z in at least one objective where other objectives may be equal. Pareto optimal front represents all the non dominated solutions set and task of multi-objective optimization algorithm is to find solutions as close to Pareto front as possible.

The paper is organized as follows. Section II describes related work of algorithms for uni-modal, multimodal and multi-objective problems in dynamic environment. Section III presents types of dynamic environment. Section IV describes the performance metrics used for all the optimization problems. Section V presents the experimental frameworks used for measuring the performance and section VI draws conclusion.

II. RELATED WORK

A number of techniques have been proposed for dynamic problems optimization, such as (but not limited to) speciation used by Daniel Parrott and Xiaodong [4] to track multiple optima is both static and dynamic environment using PSO model. Particle is the population is sorted in descending order and specie seeds are identified. Each particle is assigned to particular specie where they form a neighborhood and follow specie seed as their neighborhood best. Each specie locates different optima in parallel.

Rauf Baig and M. Rashid [5] present an algorithm which is based on the food foraging behavior of honey bees. There are several swarms of bees searching for optimal solution and at the hive a blackboard is maintained. Bees update this blackboard whenever useful information is available and use this information to relocate themselves in next iterations. One bee represents one complete solution. h bees are randomly initialized in the search space and are sorted according to their fitness value. Sub-swarms are created around m best bees. Overlapping swarms are removed and remaining bees are randomly placed to search in the space. n bees search for a peak in a collective manner in a swarm while f bees (scout bees) search randomly without any social component. After several iterations best fitness from the swarm and best fitness value of scout bees is sorted and used in next iterations.

Changhe Li, Shengxiang Yang [6] presents algorithm to track multi optima by using fast particle swam optimizer. It prevents overcrowding at the peaks and employs local search method to find local optima in promising regions. One parent swarm explores the search space, maintains diversity and search for peaks by suing fast evolutionary programming (FEP) algorithm while at the same time it creates multiple child swarms that searches locally for peaks using fast PSO algorithm. One parent swarm detect the potential areas in the search space in the case of environment change and child swarms exploit the promising areas to find the local optimum in their own local search spaces.

Shengxiang Yang and Changhe Li [7] presents clustering approach in PSO to find multiple optima in dynamic environment. The proposed CPSO (Clustering based PSO) starts from an initial swarm, named the cradle swarm. Then, sub swarms are created by a hierarchical clustering method. When sub swarms are created, local search is launched on them in order to exploit potential peaks covered by these sub swarms respectively. Finally, overlapping, convergence, and overcrowding checks are performed on the sub swarms before the next iteration starts. If an environmental change is detected, a new cradle swarm will be randomly re-generated with the reservation of the positions located by all survived sub swarms in the previous environment

Tim Blackwell and Jürgen Branke [8] presents an algorithm based on standard PSO and Charged Particle Swarm Optimizations where multi-swarms interact with each other. Whenever local optimum is identified a charged PSA swarm is created on it while the neutral swarm continues to search for other peaks. Initial positions of the particles are randomly initialized; test for change is conducted to check if the environment has changed. If change has occurred function values of each particle attractor are re-evaluated. Swarm attractor is updated and each particle's position and attractor is updated.

Lam T. Bui, Jurgan Branke, Hussein A. Abbass [9], use evolutionary multi-objective optimization methods where non dominated solutions are sorted and with each step ranks are assigned to each solution. Diversity within a same rank is determined by supporting individuals that have higher distance in all dimensions from neighbor's solutions. Artificial second objective is created by defining the timestamp and random values are assigned to each individual. Algorithm uses binary tournament selection, single point cross over and mutation operators in each generation.

Nhu Binh Ho and Joc Cing Tay [15] presents a solution for solving multiple objective job shop problems by using evolutionary algorithm and guided local search (GLD). This technique increases the convergence speed of the algorithm in finding solutions close to the Pareto Optimal front and at the same time maintains diversity in the population which discovers unexplored regions. All non-dominated solutions are maintained in a elitism memory which updates the current population to improve the solution's quality.

Gary G. yen and Haiming [17] Lu presents an algorithm based on evolutionary algorithms that uses cell based ranking and density estimation information to find non-dominated solutions and maintain diversity in the population. It converts the original multi objective problem into bi-objective domain. Rank and density values are calculated each time population is updated. Cell based ranking is used to reduce the computation when population increases or decreases dynamically.

III. DYNAMIC OPTIMIZATION PROBLEM

Optimization is the process of finding solution to a problem where the objective is to either minimize or maximize a real function variables while at the same time satisfying certain constraints. Optimization problems can be of two types.

- **Static Optimization Problems**: In these problems, throughout the execution of the algorithm environment or the landscape environment does not change which means that the location of the optimum remains the same throughout the process

[2].

- **Dynamic Optimization**:
  Landscape changes in dynamic problems and old solution may not remain to be optimum solution.

In dynamic environment when the optima changes, it is necessary that the algorithm tracks the new optima and population of the particles must be distributed uniformly and diverse enough in the whole search space in order to find the new and changed optima and also in environment where the change is drastic algorithm should locate the new optima.
Some of the examples of dynamic optimization problems are:

- Increase or decrease in cost of raw material in manufacturing business. Labor cost can increase or customer preference can change which can result in scheduling plan of the product.
- Flight scheduling can change due to some weather conditions or due to repair of some parts.
- In a job shop scheduling problem, where a number of jobs has to be assigned to a setup of machines guaranteeing a high through-put and reacting flexibly on newly arriving jobs.
- In universities, when the timing of the course changes or clashes of students occur, time table has to change.

In order to deal with such types of dynamic problems, algorithm adapts either of the two strategies [2]:

1. Whenever a change occurs restart the algorithm, memory is re-initialized and search is started from scratch. However search done until now will be lost with this strategy and algorithm will take more time to find the optimum solution.
2. Use the information from the previous steps and with each iteration adapt to the change to find new solution.

Evolutionary algorithms which are based on natural evolution process are more suitable to dynamic environment problems as compared to classical optimization techniques. However, main difficulty with standard evolutionary algorithms is the loss of diversity when they are near convergence to optimal solution. Algorithm either favors exploitation or exploration. Old information will be useful in the scenario if the changed optimum is near location of the old one however if the landscape has changed drastically then saving of too much old information will result in loss of enough diversity and algorithm may fail in reaching the new optimum. On the other hand if old information is discarded by re-initializing the memory algorithm will take long time to find the new optimum solution.

DYNAMIC ENVIRONMENT TYPES

There are three different scenarios in which a change may occur depending upon whether change is temporal or spatial. Dynamic environments can be divided into 3 categories.

1. **Change in Height of Optima Fitness**
   Many real world problems can be simulated by allowing the heights of the optima to modify in subsequent steps. Examples in real life can be increase or decrease in cost of raw material in manufacturing business, increase in labor cost or customer preference can change etc. which will result in change in the fitness of global optima. Global optima can transform into local optima and local optima can change into global one.

2. **Change in Fitness of Optima Shapes**
   change in environment can result in change in morphology or basic shape of the landscape. For example shape can change from tall to short or from wide to narrow etc. Evolutionary algorithm's performance is not much affected if only the shape of the landscape changes as they will be able to locate the optima but they can incur much difficulty and complexity when the shape changes along with other simultaneous changes.

3. **Change in Optima locations**
   Change in fitness location of the optima poses more difficulty for the existing Evolutionary algorithms. Those EAs that lack the capability of tracking changed optima fail in these scenarios. Tracking the optima becomes more difficult if the peaks move independently from others or the change does not follow any fixed or periodic pattern.

IV. PERFORMANCE METRICS

Much of the work has been done on measuring the performance of multi objective functions [10],[11], [12], 13] and multi modal [18], [19], [20] in static environment while research questions still remain on measuring performance of algorithms in a dynamic environment. Performance metric used for uni-modal problems is accuracy, where it measures the closeness of the maximum or minimum found by the algorithm to the known optimal solution. Challenging task is to determine the performance of algorithms for multi modal and multi-objective in dynamic environment. Different performance measures have been proposed in literature however more commonly used metrics for multi modal and multi objective optimization problems are explained further.

PERFORMACE METRICS FOR MULTI MODAL FUNCTIONS

In static environment performance is usually measured in terms of accuracy, convergence speed and success rate where each term is defined as follows:

1) ***Accuracy****:* The algorithm is simulated for fixed number of iterations and the *accuracy* is calculated. Accuracy refers to the measure of nearness of the known optima with the optima found by the algorithm. *Accuracy* is measured by calculating the average of the fitness differences between all known global optima to their corresponding near species seeds.

2) ***Convergence Speed****:* In order to measure the convergence speed of the algorithm, fixed accuracy level is specified and the algorithm is run until the required accuracy level is achieved. The number of function evaluation required in order to achieve the required accuracy measures the convergence speed of the algorithm. Lower the number of function evaluation higher will be the convergence speed of the algorithm.

3) Success rate refers to the number of runs (in percentage) in which the algorithm successfully finds all the known global optima.

Measuring the performance of the algorithm in dynamic environment is much more difficult and complex as compared to static environment. Conventional performance measures like mean fitness, best current fitness, and speed of convergence are not relevant in dynamic environment as fitness of the optima may change over time. Hence, in order to measure accuracy of the algorithm to locate the optima, following performance measures are more commonly used

Global Error:

In each iteration of the algorithm difference of the fittest solution found so far and known global optimum is calculated by (1)

$$e_i = 1 - \frac{f_i}{h_i}$$

Where f denotes the best fitness for the fittest solution found so far at iteration *i* and h_i is the fitness of the known global optimum at iteration *i*; e_i' (global error) is calculated by picking the minimum e_i for all iterations which represent the best e_i since the last change in the environment. [14]

$$e'_i = \min\{e_\tau, e_{\tau+1}, \dots, e_i\} \quad (2)$$

where τ is the last iteration at environment had changed.

Global error measures the accuracy of the optima found compared to the global optima. However, in order to measure the algorithm performance in locating multiple peaks along with global optimum local error metric is used.

Local Error:

Error is calculated for all the optima that are visible at the current iteration as some of the optima may disappear over time. In order to calculate best average local error since the iteration where change occurred following equation is used [14]

$$e_{avg_i} = \frac{1}{N} \sum_{i=1}^{N} e'_i \quad (3)$$

Where N is the number of optima visible at the current iteration and e'_i is calculated from (1) and (2) using the best fitness f_i of particle at the current iteration on the peak which has the height h_i, from this the best local error is calculated by taking the minimum of the all the average local errors by using the following equation

$$e'_{avg_i} = \min\{ e_{avg_\tau}, e_{avg_{\tau+1}}, \dots, e_{avg_i} \} \quad (4)$$

Using this best local average error and global error offline errors $e'_{g\ (offline)}$ and $e'_{avg(offline)}$ are calculated as described by Branke in [22]

$$e'_{g\ (offline)} = \frac{1}{I} \sum_{i=1}^{I} e'_{g_i} \quad (5)$$

$$e'_{avg\ (offline)} = \frac{1}{I} \sum_{i=1}^{I} e'_{avg_i} \quad (6)$$

Where I is the number of iterations algorithm has completed until now.

Average local error is used to measure the performance of algorithm in locating multiple optima along with the global optimum.

In order to determine fitness of the fittest solution f_i when calculating e_{avg} using (1) in (3), it is considered to be on the optimum or close to it if the optimum is visible at location of it. If no solution found by the algorithm is on the location of the optimum then error is calculated by taking the difference between the optimum's heights on a particular peak and fittest of the closest solution found by the algorithm.

PERFORMACE METRICS FOR MULTI OBJECTIVE FUNCTIONS

When measuring the performance of multi-objective problem two important criteria needs to be addresses

- Computational cost and complexity
- Quality of the solutions

Multi-objective algorithm goal is to minimize the distance of non-dominating solutions found by algorithm to Pareto Optima Front, maintaining good diversity in the sought solutions and wide variety or solutions be searched in every objective domain. Most commonly used metrics for measuring performance of multi objective algorithms are:

Generational Distance (GD):

GD finds the average distance between the non-dominating solutions sought and Pareto optimal front. The goal of the algorithm to maintain minimum GD. Euclidean distance between each solution and its closest point on optimal front is calculated and averaged over all the objectives present in the problem.

Collective Mean Error

In order to measure the performance of the algorithm on entire run, collective mean error is used. It is calculated by taking the average of GD values over the entire simulation by using following equation. [32]

$$CME = \frac{1}{I}\sum_{t=1}^{I} GD_t \qquad (7)$$

where I is the total number of iterations in simulation.

Spacing:

Spacing measures distribution of the non dominating solutions sought by the algorithm. Spacing metric calculates the distribution of individual points over non-dominating area. It measures the uniformity of solutions sought and distance between the solutions in its neighborhood.

Diversification:

Diversification metric is used to measure the diversity of the set of solutions. Goal of multi-objective optimization is to find solutions as diverse as possible while at the same time are close to Pareto optimal front

V. EXPERIMENTAL FRAMEWORK

Creating an experimental framework to measure the performance of the algorithms in dynamic environment is difficult and complex. Since the real world environment is subjected to ad-hoc and unpredictable changes, it is difficult to capture all the dimensions for creation of simulated environment. In simulations it is very difficult to implement the real world dynamic problems Therefore nearly all the algorithms proposed for dynamic environment make some assumptions about the nature of the problem e.g. assuming that dimension of the problem will not change throughout the execution of the algorithm. Change in environment can be any of the three types mentioned in section III and algorithm should have the capability of finding the optimal solution in all these dynamic environment types.

Experimental frameworks used for multi modal and multi objective functions are explained forewords.

EXPERIMENTAL FRAMEWORK FOR MULTI MODAL FUNCTIONS

To determine the performance of multi modal optimization algorithms two common frameworks used are :

1. Dynamic Test Function Generators
2. Moving Peaks Benchmark (MPB)

DYNAMIC TEST FUNCTION GENERATORS

Different dynamic test function generators are proposed in literature [23],[24], [25].

In Morrison and De Jong's dynamic test function generator [23] can be used to create an environment that changes over time. This test problem generator is capable of generating multiple and mixture of different dynamic environments. In the simplest dynamic environment, the shape or overall structure of the landscape remains the same but it only glides along any dimension. In this case a static function is used to describe the basic morphology of the landscape and then over time add some parameters to it to change its shape.

Consider we have a static two dimensional landscape

$$f(x, y)$$

Then change in this static function can be done by

$$\Delta(x, y, t) = f(x_{t-1} + \Delta x_t, y_{t-1} + \Delta y_t) \qquad (8)$$

Algorithm's motion will be specified by Δxt and Δyt. Also associated with it is to control how rapid or slow will be the change and whether the change is periodic or un-periodic. The function generator used can be applied to any number of dimensions. For 2 dimensional problems following function can be used

$$f(X, Y) = \max_{i=1}^{N} [H_i - R_i.\sqrt{(X - X_i)^2 + (Y - Y_i)^2}] \quad (9)$$

where N specifies the number of optima in the landscape and each optima is individually specified by its position (X_i, Y_i) where i^{th} optima x position is specified by X_i and y position is specified by Y_i. Each optimum has height H_i, and slope R_i. Max function is used to inter-mingle the specified optima together. On each call to this function generator will result in a randomly generated landscape where range of the values are as follows:

Height and range of the landscape are generated by following equations where H_{base} and H_{range} are specified by user.

$$H_i \epsilon\, [H_{base}, H_{base} + H_{range}] \qquad (10)$$

$$R_i \epsilon [R_{base}, R_{base} + R_{range}] \quad (11)$$

Wide variety of landscapes can be generated by specifying the number of optima i.e. N, minimum value of the slope variable i.e. $R_{base,}$ maximum range of the slope i.e. R_{range}, minimum value of the height optima can take i.e. H_{base} and range of values height of the optima can take i.e. H_{range}

MOVING PEAKS BENCHMARK (MPB)

A number of MBP have been proposed in literature of which MPB proposed by Branke [26] has been widely used. In this MPB optima can vary by changing the location, slope or height of the peaks. The function generates a landscape with specified number of peaks in given dimensions whose location and shape of the landscape changes over time. MPB is defined by a function [26]

$$F(\vec{x}, t) = \max_{i=1,\ldots,p} \frac{H_i(t)}{1 + W_i(t)\sum_{j=1}^{D}(x_j(t) - X_{ij}(t))^2} \quad (12)$$

Where D is the dimension of the problem, $X_{ij}(t)$ is the location of the peak i and time t. $h_i(t)$ and $W_i(t)$ is the height and width of the ith peak at time t and p is the total number of optima in the landscape. Position of each optimum changes its location and moves in random direction where height and width severity parameters are added to height and width respectively and position is altered by adding a vector of linear combination and correlation parameters at each change in environment

TRADEOFFS IN PERFORMANCE METRICS

Changing environment requires adaptation of existing algorithms with additional cost. This cost has important role when adapting strategy in the algorithm needs to be implemented. When cost consideration is important then tradeoff is algorithm. Similarly in multi modal and multi objective optimization algorithms different tradeoffs exists between performance metrics.

In multi modal functions increasing the population size will result in higher accuracy but at the same time will result in increased computational cost.

In multi objective solutions are sought which fall on ideal Pareto Optima Front while at the same time are distributed uniformly throughout the search space which compete with each other. Incorporating diversity may decrease the quality of the resulting Pareto front [17]. To increase the quality and accuracy of solutions, grid of enhanced size is required but it will also increase the computational cost and complexity of the algorithm.

EAs can include cost factor in the fitness evaluation function. By doing this the algorithm will favor the solution that is in the near neighborhood of the old solution and It will avoid the one that diverges from it.

EAs goal is to find solutions that are accurate, robust and are of higher quality while at the same time less computationally expensive. If cost of adaptation is too high or the changes in environment occur so frequently that the EA algorithm fails to find optimal solution, solutions that are near the optimal ones in all feasible scenarios are sought.

VI. CONCLUSION

This paper has presented the metrics used for measuring the performance of algorithms in dynamic environment. Traditionally performance measures used in static environment cannot be used in dynamic set up as location, shape and fitness may change over time. EAs which are based on natural evolution process are more appropriate in finding solutions to multi-modal and multi objective problems as compared to classical optimization techniques. In order to calculate the performance different experimental frameworks are required. When comparing among different algorithms available adaptation cost and accuracy play an important role where tradeoffs exists between different performance metrics like increasing the quality will increase the computational cost of the algorithm. In future more research needs to be carried out to formulate experimental frameworks that closely simulate the dynamism of the real world problems.

ACKNOWLEDGMENT

The authors of the paper would like to acknowledge the support provided by NUCES faculty, CS lab supervisors and librarian for providing the necessary equipment and tools to conduct out this research. We would also like to thank the anonymous readers and reviewers for comments that greatly improved this paper. Special thanks to our family and friends for their assistance and patience to extricate time for this study.

REFERENCES

[1] Andries P. Engelbrecht, 2002, “Computational Intelligence An Introduction”.

[2] Ramoza Ahsan, Waseem Shehzad, Ruaf Baig, “Trends and Challenges in Dynamic Optimization Problems”, Hybrid Intelligent systems , IEEE conference 2011.

[3] Jurgen Branke, “Evolutionary Optimization in dynamic Environments”, Springer 2002.

[4] Daniel Parrot and Xiaodong Li, “Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation”, IEEE Transactions on Evolutionary Computation, Vol. 10, No. 4, August 2006.

[5] Rauf Baig and M. Rashid, “Honey Bee Foraging Algorithm for Multimodal & Dynamic Optimization Problems”, GECCO’07, 2007.

[6] Changhe Li, Optimization Problems”, Fourth International Conference on Natural Computation Shengxiang Yang, “Fast Multi-swarm Optimization for Dynamic, IEEE 2008.

[7] Shengxiang Yang and Changhe Li, “A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments”, IEEE Transactions on Evolutionary Computation, Vol. 14, No. 6, December 2010.

[8] Tim Blackwell and Jürgen Branke, “Multi-swarm Optimization in Dynamic Environments”, Springer 2004.

[9] Lam T. Bui, Jurgan Branke, Hussein A. Abbass, “Multiobjective optimization for dynamic environments”, Evolutionary Computation 2005.

[10] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, UK (2001).

[11] Knowles, J.D. and Corne, D.W.: On metrics for comparing nondominated sets. In Proc. of the 2002 Congress on Evolutionary Computation Conference (CEC’02). IEEE Press (2002) 711-716.

[12] Hanson, M.P. and Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Technical University of Demark, March 1998.

[13] Zitzler, E., Thiele, L. Laumanns, M., Fonseca, C.M. and Fonseca, V.G.d.: Performance assessment of multiobjective optimizers: an analysis and review. In IEEE Trans. on Evolutionary Computation 7(2), (2003) 117-132.

[14] Daniel Parrott and Xiaodong Li,” Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation”, IEEE Transactions on Evolutionary Computation, Vol. 10, No.4, August 2006.

[15] Nhu Binh Ho, Joc Cing Tay, “Solving Multiple Objective Flexible Job Shop Problems by Evolution and Local Search”, IEEE Transactions on System, Man, and Cybernetics, Part C: Applications And Reviews, Vol. 38, No.5, September 2008.

[16] Robin C. Purshouse, Peter J. Fleming, “On the Evolutionary Optimization of Many Conflicting Objectives”, IEEE Transactions on Evolutionary Computation, Vol. 11, no. 6, December 2007.

[17] Gary G.Yen, Haiming Lu, “Dynamic Multiobjectiv Evolutionary Algorithm:Adaptive Cell-Based Rank and Density Estimation”, IEEE Transactions on Evolutionary Computation, VOL.7,NO.3, June 2003.

[18] J. J. Liang, A. K. Qin, Ponnuthurai Nagaratnam Suganthan, “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions”, IEEE Transactions on Evolutionary Computation, Vol. 10, no. 3, June 2006.

[19] Konstantinos E. Parsopoulos, Michael N. Vrahatis, “On the Computation of All global Minimizers Through Particle Swarm Optimization”, IEEE Transactions on Evolutionary Computation, Vol. 8, no. 3, June 2004

[20] Muhammad Rashid, Abdul Rauf Baig, “Speciation with GP Based Hybrid PSO”, 2009 International Symposium on Computing, Communication and Control, Singapore, Proc. Of CSIT vol. 1, 2011.

[21] Li Liu, S Ranji Ranjithan, ”An adaptive optimization technique for dynamic environments “, Elsevier, Engineering Applications of Artificial Intelligence (2010).

[22] J. Branke, Evolutionary Optimization in Dynamic Environments. Norwell, MA: Kluwer , 2002.

[23] K. A. De Jong and R. W. Morrison, “A test problem generator for non-stationary environments,” in Proceedings Congr. Evolutionary Computation 1999, pp. 2047–2053.

[24] Julio Barrera and Carlos A. Coello Coello, “Test Function Generators for Assessing the Performance of PSO Algorithms in Multimodal Optimization”, Handbook of Swarm Intelligence, 2010 – Springer.

[25] Jani Ronkkonen, Xiaodong Li, Ville Kyrki, Jouni Lampinen, “A Generator for Multimodal Test Functions with Multiple Global Optima”, SEAL 08 Proceedings of the 7th International Conference on Simulated Evolution and Learning Springer, 2008.

[26] J. Branke, “Memory enhanced evolutionary algorithms for changing optimization problems,” in Proc. Congr. Evol. Comput., vol. 3. 1999, pp. 1875–1882.

[27] Deb, K., Bhaskara Rao, U. and Karthik, S.” Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-Thermal Power Scheduling Bi-Objective Optimization Problems”. KanGAL Report No. 2006008 (2006).

[28] Yamasaki, K.” Dynamic Pareto optimum GA against the changing environments”. In Proc. Genetic Evolutionary Computation Conf. Workshop, San Francisco, CA (2001) 47-50.

[29] Bingul, Z., Sekmen, A. and Zein-Sabatto, S.”In alife-inspired evolutionary algorithm applied to dynamic multi-objective problems”. In Proc. Artificial Neural Networks Engineering Conf., C.H. Dagli, et al. (eds.), New York, (2000) 273-278.

[30] Farina, M., Deb, K. and Amato, P.” Dynamic multiobjective optimization problems: test cases, approximation and applications”. In IEEE Trans. on Evolutionary Computation 8(5), (2004) 425-442.

[31] Jin, Y. and Sendhoff, B.: Constructing dynamic test problems using the multi-objective optimization concept. In: Applications of Evolutionary Computing. LNCS 3005, Raidl, G.R. et al. (eds.) Springer (2004) 525-536.

[32] Xiaodong Li, Jurgen Branke, Michael Kirly, “On Performance Metrics and Partcile Swarm Methods for Dynamic Multiobjective Optimization Problems”, Evolutionary Computation, 2007. CEC 2007. IEEE Congress .

[33] Shengxiang Yang, Changhe Li, “A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments”. IEEE Transactions

on Evolutionary Computation, Vol. 14, No. 6, December 2010.

[34] Yu Liu, Xiaoxi Ling, Zhewen Shi, Mingwei LV, Jing Fang, Liang Zhang, "A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization", *Journal of Software*, Vol 6, No 12 (2011), 2449-2455, Dec 2011

Ramoza Ahsan received her MS degree in Computer Science from National University of Computer & Emerging Sciences (NUCES), Islamabad, Pakistan in 2011. Currently she is working as a Lecturer in National University of Computer & Emerging Sciences Islamabad, Pakistan. She has research interests in the field of Artificial Intelligence, Computational Intelligence, swarm intelligence and Machine Learning.

Dr. Waseem Shahzad received his MS degree in Computer Science from the department of Computer Science, National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan in 2007 and later did his Ph. D in Computer Science from (NUCES) in 2010.

Currently he is working as Assistant Professor in National University of Computer and Emerging Sciences Islamabad, Pakistan. He has research interests in the fields of Data Mining, Computational Intelligence, Machine Learning, Theory of Computation and Soft Computing. He has several research publications in these areas.