Bhargava, S. (2013). A Note on Evolutionary Algorithms and Its Applications. *Adults Learning Mathematics: An International Journal, 8*(1), 31-45

A Note on Evolutionary Algorithms and Its Applications

Shifali Bhargava

Dept. of Mathematics, B.S.A. College,

Mathura (U.P)- India.

<shifalibhargava@gmail.com>

Abstract

This paper introduces evolutionary algorithms with its applications in multi-objective optimization. Here elitist and non-elitist multiobjective evolutionary algorithms are discussed with their advantages and disadvantages. We also discuss constrained multiobjective evolutionary algorithms and their applications in various areas.

Key words: evolutionary algorithms, multi-objective optimization, pareto-optimality, elitist.

Introduction

The term evolutionary algorithm (EA) stands for a class of stochastic optimization methods that simulate the process of natural evolution. The origins of EAs can be traced back to the late 1950s, and since the 1970's several evolutionary methodologies have been proposed, mainly genetic algorithms, evolutionary programming, and evolution strategies. All of these approaches operate on a set of candidate solutions. Using strong simplifications, this set is subsequently modified by the two basic principles of evolution: selection and variation. Selection represents the competition for resources among living beings. Some are better than others and more likely to survive and to reproduce their genetic information. In evolutionary algorithms, natural selection is simulated by a stochastic selection process.

Each solution is given a chance to reproduce a certain number of times, dependent on their quality. Thereby, quality is assessed by evaluating the individuals and assigning them scalar fitness values. The other principle, variation, imitates natural capability of creating "new" living beings by means of recombination and mutation. Although the underlying principles are simple, these algorithms have proven themselves as a general, robust and powerful search mechanism. Moreover, EAs seem to be especially suited to multi-objective optimization because they are able to capture multiple pareto-optimal solutions in a single simulation run and may exploit similarities of solutions by recombination.

The application of evolutionary algorithms (EAs) in multi-objective optimization is currently receiving growing interest from researchers with various backgrounds. Most research in this area has understandably concentrated on the selection stage of EAs, due to the need to integrate vectorial performance measures with the inherently scalar way in which EAs reward individual performance, i.e., number of offspring. The first pioneering studies on evolutionary multiobjective optimization appeared in the mid-1980s (Fourman, 1985; Schaffer, 1984; Schaffer, 1985). After that a few different MOEA implementations were proposed in the years 1991–1994 (Fonseca & Fleming, 1993; Hajela & Lin, 1992; Horn et al., 1994; Srinivas & Deb, 1994; Kursawe, 1990). Later, these approaches (and variations of them) were successfully applied to various multiobjective optimization problems (Cunha et al., 1999; Fonseca & Fleming, 1998; Ishibuchi & Murata, 1997; Parks & Miller, 1998).

The question is which EA implementations are suited to which sort of problem and what are the specific advantages and drawbacks, respectively, of different techniques.

- In contrast to SOPs, there is no single criterion to assess the quality of a trade-off front; quality measures are difficult to define. This might be the reason for the lack of studies in that area. Up to now, there has been no sufficient, commonly accepted definition of quantitative performance metrics for multiobjective optimizers.
- There is no accepted set of well-defined test problems in the community. This makes it difficult to evaluate new algorithms in comparison with existing ones.
- The various MOEAs incorporate different concepts, e.g., elitism and niching that are in principle independent of the fitness assignment method used. However, it is not clear what the benefits of these concepts are. For instance, the question of whether elitism can improve multi-objective search in general is still an open problem.

The above issues sketch the scope of the present work and result in the following research goals:

1. Comparison and investigation of prevailing approaches.
2. Improvement of existing MOEAs, possible development of a new, alternative evolutionary method.
3. Application of the most promising technique to real-world problems in the domain of system design.

The first aspect aims at finding advantages and disadvantages of the different approaches and yielding a better understanding of the effects and the differences of the various methods. This involves the careful definition of quantitative performance measures which ideally allow for different quality criteria. The last goal is important for identifying those problem features which cause the most difficulty for MOEAs to converge to the pareto-optimal front. The comparison also includes the examination of further factors of evolutionary search such as populations size and elitism.

As a result, these investigations may either contribute to the problem of sampling the search space more efficiently by improving existing methods or lead to the development of a new evolutionary approach. Usually, these applications are by far too complex to be handled by exact optimization algorithms.

This paper reviews current evolutionary approaches to multi-objective optimization discussing their similarities and differences. It also tries to identify some of the main issues raised by multi-objective optimization in the context of evolutionary search, and how the methods discussed address them. From the discussion, directions for future work, in multi-objective evolutionary algorithms are identified.

Evolutionary Approaches to Multi-objective Optimization

The family of solutions of a multiobjective optimization problem is composed of all those elements of the search space which are such that the components of the corresponding objective vectors cannot be all simultaneously improved. This is known as the concept of Pareto optimality.

A more formal definition of pareto-optimality is as follows:

Consider without any loss of generality, the minimization of n components $f_k, k = 1,2,...n$ of a vector function f of a vector variable x in a universe U, where $f(x) = (f_1(x), f_2(x),...f_n(x))$.

Then a decision vector $x_u \in U$, is said to be pareto-optimal if and only if $\exists$ $x_v \in U$ for which $v = f(x_v) = (v_1, v_2 ... v_n)$ dominates $u = f(x_u) = (u_1, u_2 ... u_n)$

The set of all pareto-optimal decision vectors is called the pareto-optimal efficient or admissible set of the problem. The corresponding set of objective vectors is called the non-dominated set. The notion of pareto-optimality is only a first step towards the practical solution of a multiobjective problem, which usually involves the choice a single compromise solution from the non-dominated set according to some preference information.

Evolutionary algorithms seem particularly suitable to solve multi-objective optimization problems, because they deal simultaneously with a set of possible solutions, the so-called population. This allows to find several members of the pareto optimal set in a single run of the algorithm instead of having to perform a series of separate runs as in the case of traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the pareto front as they deal easily with discontinuous or concave pareto fronts, whereas these two issues are of real concern for mathematical programming techniques. MOEA are very attractive MOP solution techniques because they address both search and multi-objective decision making. Additionally they have the ability to search partially ordered spaces for several alternative trade-offs. They find a wide range of non-dominated solutions close to the true pareto-optimal solutions.

A MOEA defining characteristic is the set of multiple objectives being simultaneously. Otherwise task decomposition clearly shows little structural difference between the MOEA and its single objective EA counterparts.

General EA Tasks

1. Initialize population
2. Fitness evaluation (vector/ fitness transformation)
3. Recombination
4. Mutation
5. Selection

Non- Elitist Multi-objective Evolutionary Algorithms

Non-Elitist Multi-Objective Evolutionary Algorithms (MOEA) are algorithms which do not use any elite-preserving operator. Some important Non-Elitist MOEA includes the following:

1. Vector Evaluated Genetic Algorithm (VEGA)

This is the simplest possible multi-objective GA and is a straight forward extension of single-objective GA for multi-objective optimization. Schaffer implemented this first multi-objective GA to find a set of non-dominated solutions (Schaffer, 1984). This GA evaluated an objective vector, with each element of the vector representing each objective function and emphasizes solutions which are good for individual objective functions. To find intermediate trade-off solutions, Schaffer allowed cross-over between any two solutions in the entire population. Also a VEGA has the same computational complexity as that of single-objective GAs. The main advantage of a VEGA is that it uses a simple idea and is easy to implement. Only minor changes are required to be made in a simple GA to convert it to a multi-objective GA and this does not incur any additional computational complexity. But here, as each solution in a VEGA is evaluated only with one objective function, thus every solution is not tested for other objective functions, all of which are also important in the context of multi-objective optimization.

2. Vector-Optimized Evolution Strategy

In this approach, the basic self-adaptive evolution strategy for single-objective optimization is modified to handle multi-objective optimization problems. This algorithm performs a domination check to retain non-dominated solutions and a niching mechanism to eliminate crowded solutions. The simulation results were shown on a single problem and no further work has been pursued, hence this is not used by current researchers (Kursawe, 1990).

3. Weight Based Genetic Algorithm

The key issue in WBGAs is to maintain diversity in the weight vectors among the population members. In WBGAs the diversity in the weight vectors is maintained in two ways. In the first approach, a niching method is used only on the substring representing the weight vector, while in the second approach, carefully chosen subpopulations are evaluated for different pre-defined weight vectors, an approach similar to that of the VEGA. Since a WBGA uses a single-objective GA, not much change is needed to convert a simple GA implementation into a WBGA one. Moreover, the complexity of the algorithm is smaller than other multi-objective evolutionary algorithms. As the WBGA uses a proportionate selection procedure on the shared fitness values, for mixed type of objective functions (some are to be minimized and some are to be maximized), complications may arise in trying to construct a fitness function. WBGA may also face difficulties in finding pareto-optimal solutions in problems having non-convex pareto-optimal region (Hajela et al., 1993).

4. Multi-objective Genetic Algorithm

Fonseca and Fleming first introduced a multi-objective GA (MOGA) which used the non-dominated classification of a GA population (Fonseca & Fleming, 1993). This explicitly caters to emphasize non-dominated solutions and simultaneously maintains diversity in the non-dominated solutions. The MOGA differs from a standard tripartite GA in the way fitness is assigned to each solution in the population. The rest of the algorithm is the same as in classical GA. Since niching is performed in the objective space, the MOGA can be easily applied to other optimization problems. This algorithm may be sensitive to the shape of the pareto optimal front and to the density of solutions in the search space.

5. Non-Dominated Sorting Genetic Algorithm

In non-dominated sorting GA, the dual objectives in a multi-optimization algorithm are maintained by using a fitness assignment scheme which prefers non-dominated solutions and by using a sharing strategy which preserves diversity among solutions of each non-dominated front. The computational complexity of the fitness assignment procedure is mainly governed by the non-dominated sorting procedure and the sharing function implementation. The main advantage of an NSGA is the assignment of fitness according to non-dominated sets. An NSGA progresses towards the pareto-optimal region frontwise (Srinivas & Deb, 1994).

6. Predator-Prey Evolution Strategy

This strategy does not use a domination check to assign fitness to a solution but uses the concept of predator-prey model. The main advantages of this method are its simplicity and that it does not emphasize non-dominated solutions directly. The disadvantage of this strategy is that no explicit operator is used to maintain a spread of solutions in the obtained non-dominated set. Instead, each predator is assigned the task of eliminating the worst neighboring solution with respect to a different objective. Also there is no special care taken to maintain the intermediate solutions (Laumanns et al., 1998)

7. Distributed Sharing GA

In this approach, the distributed island model is used to maintain diversity among non-dominated solutions. The GA population is divided into a number of subpopulations and independent genetic operations are performed to each island. Subpopulations from all islands are collected together and the non-dominated solutions are recorded (Hiroyasu et al., 1999).

8. Distributed Reinforcement Learning Approach

Maraino and Morales suggested a distributed reinforcement learning approach, where a family of agents is assigned to different objective functions (Mariano & Morales, 2000). Each agent proposes a solution to optimize its objective function. All such solutions are combined and a non-dominated compromised set of solutions are identified. Each non-dominated solution is rewarded. In the context of solving continuous search space problems, an agent considers solutions in a particular search direction from its current location. The solution is evaluated by the agent's corresponding objective function. The rewarding mechanism provides a direction for the algorithm to move towards the pareto-optimal region and the non-domination check maintains a diverse set of solutions, while simultaneous creation of multiple solutions by a directional search method helps to find new solutions in the search space.

9. Nash GA

This GA is motivated by a game theoretic approach in which one player is allowed to get associated with each objective function and it tries to optimize its objective function while keeping other objective functions unchanged. In a periodic sequence of operations, the Nash GA is terminated when no more improvement is recorded. At this steady-state scenario, the resulting solution is a Nash-Equilibrium solution and is a candidate pareto-optimal solution. Although the investigators claim better convergence properties of this GA compared to the NSGA, it is clear that an explicit niche-performing operator must be used to maintain multiple pareto-optimal solutions (Sefrioui & Periaux, 2000).

Elitist Multi-objective Evolutionary Algorithms

These are evolutionary algorithms which use elite preserving operator. Elite preservation or emphasizing currently elite solutions is an important operator in an EA. An elite preserving operator favors the elites of a population by giving them an opportunity to be directly carried over to the next generation. Elitism can be implemented to different degrees in an MOEA. The presence of elitism should improve the performance of a multi-objective EA, but care must be taken to control the effective degree introduced in the progress. Now we present some algorithms that attempt to achieve a controlled elitism in multi-objective evolutionary optimization:

1. Rudolph's Elitist Multi-Objective Evolutionary Algorithms

Rudolph suggested a multi-objective evolutionary algorithm which required the introduction of a diversity preservation mechanism (Rudolph, 2001) This algorithm with a positive variation kernel of its search operators allows convergence to the pareto-optimal front in a finite number of trials in finite search space problems. The positiveness of the variation kernel makes sure that the probability of creating any offspring from an arbitrary set of parent solutions in a finite number of trials is one. Thus in any population, if no pareto-optimal solution exists, the positiveness of the variation kernel of the combined search operators ensures that one such member will be created in a finite number of trials. With the elite preserving strategy of the above algorithm, this member cannot ever be deleted from the population. The main disadvantage of this algorithm is that it does not ensure any diversity among the obtained solutions.

2. Elitist Non-Dominated Sorting Genetic Algorithm

Deb suggested an elitist non-dominated sorting GA (termed NSGA- II) which uses an explicit diversity-preserving mechanism (Deb, 2000). The overall complexity of the NSGA- II is at most $O(MN^2)$. The diversity among non-dominated solutions is introduced by using the crowding comparison procedure which is used with the tournament selection and during the population reduction phase. Since solutions compete with their crowding distances, no extra niching parameter is required here. In the absence of the crowded comparison operator, this algorithm also exhibits a convergence proof to the pareto-optimal solution set similar to that in Rudolph's algorithm, but the population size would grow with the generation counter. The elitism mechanism does not allow an already found pareto-optimal solution to be deleted. This algorithm loses its convergence property when the crowded comparison is used to restrict the population size. In latter generations when more than N members belong to the first non-dominated set in the combined parent-offspring population, some closely packed pareto-optimal solutions may give their places to other non-dominated yet non-pareto-optimal solutions. Although these latter solutions may get dominated by other pareto-optimal solutions in a later generation, the algorithm can resort into this cycle of generating pareto-optimal and non pareto-optimal solutions before finally converging to a well distributed set of pareto-optimal solutions.

3. Strength Pareto Evolutionary Algorithm (SPEA)

Zitzler and Thiele (1998) proposed Strength Pareto Evolutionary Algorithm (SPEA) which introduced elitism by explicitly maintaining an external population $\overline{P}$. This population stores a fixed number of the non-dominated solutions that are found until the beginning of a simulation. At every generation, newly found non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved. The SPEA not only preserves the elites but also uses these elites to participate in the genetic operations along with the current population in the hope of influencing the population to steer towards good regions in the search space. In the SPEA, clustering ensures that a better spread is achieved among the obtained non-dominated solutions. This clustering algorithm is parameter-less, thereby making it attractive to use. The fitness assignment procedure in the SPEA is more or less similar to that of Fonseca and Fleming's (1993) MOGA and is easy to calculate. In SPEA, if a large external population is used, the selection pressure for the elites will be large and the SPEA may not be able to converge to the pareto-optimal front. On the other hand, if a small external population is used, the effect of elitism will be lost. Moreover, in the SPEA fitness assignment, an external solution, which dominates more solutions, get a worse fitness. This assignment is justified when all dominated solutions are concentrated near the dominating solution.

4. Pareto-Archived Evolution Strategy (PAES)

PAES uses an evolution strategy and its main crux lies in the way that a winner is chosen in the midst of multiple objectives. The PAES has a direct control on the diversity that can be achieved in the pareto-optimal solutions. The PAES performs better when compared to other methods in handling problems having a search space with non-uniformly dense solutions. The disadvantage of PAES is that change of the depth parameter changes the number of hypercube exponentially, thereby making it difficult to arbitrarily control the spread of solutions. In order to give the PAES a global perspective the concept of multi-membered ES is introduced. Since offspring are not compared against each other and only compared with the archive, this method does not guarantee that the best non-dominated solutions among the offspring are emphasized enough (Knowles, & Corne, 2000).

5. Multi-objective Messy Genetic Algorithm (MOMGA)

This is the extension of the original messy GA in which the use of m different template strings in an era is suggested (Veldhuizen, 1999). In the level-1, MOMGA each partial string is filled from m template strings chosen randomly before the era has begun. Each filled string is evaluated with a different objective function. The objective vector obtained in this process is used in the selection operator. At the end of an era, the best solution corresponding to each objective function is identified and is assigned as the template string corresponding to that objective function. A concurrent MOMGA was also proposed by suggesting parallel applications of the MOMGA with different initial random templates. At the completion of all MOMGAs, the obtained external sets of non-dominated solutions are all combined together and the best non-dominated set is reported as the obtained non-dominated set of solutions of the CMOMGA. Although the solutions obtained by this procedure do not indicate the robustness associated with an independent run of an MOMGA, this parallel approach may be desirable in practical problem solving. A study using a probabilistically complete initialization of MOMGA population to reduce the computational burden is an improvement over the past studies (Zydallis et al., 2001).

6. Non-dominated sorting in Annealing GA (NSAGA)

This non-dominated sorting in annealing GA (NSAGA) uses a simulated annealing-like temperature reduction concept along with the Metropolis criterion. The first-stage probability calculation is along the lines of finding the transition probability of creating the offspring population from the parent population. The second probability calculation is based on the Metropolis criterion, which uses an energy function related to the number of non-dominated solutions in a population. In an elitist sense, an offspring population is accepted only when the probability of creating such a population and accepting it with the Metropolis criterion with an updated temperature concept is adequate. Clearly the goal of this work is to modify the NSGA procedure with a simulated annealing-like acceptance criterion, so that a proof of convergence can be achieved.

7. Multi-objective Micro-GA

This Multi-objective Micro-GA maintains two populations. The GA population is operated in a similar way to that of the single-objective micro-GA, whereas the elite population stores the non-dominated solutions obtained by the GA. The elite archive is updated with new solutions in a similar way to that achieved in the PAES. The search space is divided into a number of grid cells. Depending on the crowding in each grid with non-dominated solutions, a new solution is accepted or rejected in the archive (Coello & Toscano, 2000).

8. Elitist MOEA with Coevolutionary Sharing (ERMOCS)

This multi-objective GA (ERMOCS) is based on Goldberg and Wang's coevolutionary sharing concept (Goldberg & Wang, 1998; Neef et al., 1999). For maintaining diversity among non-dominated solutions, the coevolutionary shared niching (CSN) method is used. The elite preservation is introduced by using a pre-selection scheme where a better offspring replaces a worse parent solution in the recombination procedure. In the coevolutionary model, the customer and businessman populations interact in the same way as in the CSN model, except an additional imprint operator is used for emphasizing non-dominated solutions. After both customer and businessman populations are updated, each businessman is compared with a random set of customers. If any customer dominates the competing businessman and the latter is at least a critical distance away from other businessmen, it replaces the competing businessman. In this way non-dominated solutions from the customer population get filtered and find their place in the businessman population. On a scheduling problem, ERMOCS is able to find well-distributed customer as well as businessman populations after a few generations.

Constrained Multi-objective Evolutionary Algorithms

In most practical search and optimization problems, constraints are evident. Often the constraints are many in numbers and are nonlinear. Now we deal with several multi-objective evolutionary algorithms which have been particularly suggested for handling constraints.

1. Penalty Function Approach

In the penalty function approach, the constraint violation in an infeasible solution is added to each objective function. Thereafter, the penalized objective function values are optimized. For relatively large penalty terms (compared to objective function values), this method practically compares infeasible solutions based on their constraint violations. Again for the same reason, a feasible solution will practically dominate an infeasible solution. Both of these characteristics together allow the population members to become feasible from infeasible solutions and, thereafter, allow solutions to converge closer to the true pareto-optimal solutions.

2. Jimenez-Verdegay-Gomez-Skarmeta's Method

This work suggested a careful consideration of feasible and infeasible solutions and the use of niching to maintain diversity in the obtained pareto-optimal solutions. This algorithm uses the binary tournament selection in its core. Here feasible and infeasible solutions are carefully evaluated by ensuring that no infeasible solution gets a better fitness than any feasible solution (Jimenez et al., 1999). Only inequality constraints of the lesser-than-equal-to type are considered in their study, whereas any other constraints can also be handled by using the procedure. The disadvantage of this algorithm is that by preserving diversity among infeasible solutions explicitly, the progress towards the feasible region may be sacrificed. Also there exist a couple of additional parameters which a user must set right. In order to make the non-domination check less stochastic, a large comparison check is needed. Furthermore, the algorithm does not explicitly check the domination of participating solutions in a tournament.

3. Constrained Tournament Method

Here the definition of domination is modified. Before comparing two solutions for domination, they are checked for their feasibility. If one solution is feasible and the other is not, the feasible solution dominates the other. If two solutions are infeasible, the solution with the smaller normalized constraint violation dominates the other. On the other hand, if both solutions are feasible, the usual domination principle is applied. The advantage of this method is that in addition to the constraint violation computations, this strategy does not require any extra computational burden. The constraint domination principle is generic and can be used with any other MOEAs. Since it forces an infeasible solution to be always dominated by a feasible solution, no other constraint handling strategy is needed.

4. Ray-Tai-Seow's Method

Ray, Tai, and Seow (2001) suggested a more elaborate constraint handling technique, where the constraint violations of all constraints are not simply added together; instead, a non-domination check of the constraint violations is made. Here, three different non-dominated sorting procedures are used. In addition to a non-dominated sorting of the objective functions, a couple of non-dominated sortings using the constraint violation values and a combined set of objective function and constraint violation values are needed to construct the new population. This algorithm handles infeasible solutions with more care than any other of the constrained handling techniques and diversity is maintained in the population. But the disadvantage is that in a later generation, when all population members are feasible and belong to a sub-optimal non-dominated front, the algorithm stagnates. Also during the crossover operation, three offspring are created. The first one is created by using a uniform crossover with an equal probability of choosing one variable value from each parent. The other two solutions are created by using a blend crossover, which uses a uniform probability distribution over a range that depends on a number of threshold parameter values. The difficulty arises in choosing parameter values related to each of these operators. Another difficulty arises because five solutions are accepted after each crossover operation. This process will cause the population to soon lose its diversity. Three non-dominated ranking and head-count computations make the algorithm more computationally expensive than the other algorithms.

Salient issues of Multi-objective Evolutionary Algorithms

With the success of MOEAs in different problem domains, many new techniques have been suggested. This demands a proper method of assessing the performance of a newly suggested algorithm. Since an MOEA is supposed to perform the tasks of converging close to the true pareto-optimal front and maintaining a diverse set of non-dominated solutions, an algorithm must be assessed with respect to both of these tasks. Ironically, it is difficult to have one performance metric to evaluate both of the above issues adequately.

For evaluating a new algorithm, there is also a need to test it with problems possessing known complexities of the search space and with a known pareto-optimal set. Knowledge of the exact locations of the pareto-optimal solutions is helpful in investigating the search abilities of an algorithm. With the development of a number of MOEAs over the past few years, there have been some studies in comparing them systematically. Those MOEAs which properly implemented elite preservation, emphasized non-dominated solutions, and maintained diversity among non-dominated solutions, all performed well. In several studies it was clear that elite preservation is an important operation in converging as well as sustaining a good diverse set of non-dominated solutions.

An important aspect of maintaining diversity among non-dominated solutions is the space in which the diversity is required. The diversity preserving operator must treat the proximity of the solutions in the decision variable space. On the other hand, if the diversity in the objective space is more important, the proximity must be measured in the objective space. It is important to keep in mind that the proximity in one space may not mean that a proximity in the other space would be automatically obtained. This has been found to be particularly true in certain nonlinear and complex problems.

The concept of multi-objective optimization can also be used to solve other kinds of optimization problems in an efficient way. For example, a constrained single objective optimization problem can be considered as being a multi-objective optimization problem of optimizing the objective function and minimizing all constraint violations. The principle of finding multiple optimal solutions can also be extended to other similar problems, such as goal programming. Because of the lack of an optimization algorithm which can find multiple optimal solutions simultaneously, goal programming approaches traditionally use relative weights of objectives and resort to finding one solution corresponding to one weight vector at a time. Also an interesting aspect is that as the number of objectives increase, a large proportion of a randomly chosen population becomes non-dominated. When this happens, introduction of elitism becomes tricky. This is because a large number of population members are candidate elite solutions, therefore not allowing many solutions to be accepted in any generation. Moreover, there are ways to choose an adequate population size such that a reasonable proportion of the population members belongs to the dominated fronts for initially introducing variability in the population.

In MOEAs, convergence to the pareto-optimal front and simultaneous maintenance of a good distribution are both important. Although there exist a number of MOEAs with theoretical convergence properties to the true pareto-optimal front, they do not guarantee maintaining any spread of solutions. More studies to develop MOEAs with properties of convergence as well as spread of solutions remain as an imminent challenge to the researchers of MOEAs.

Applications of Multi-objective Evolutionary Algorithms

Here we discuss concisely some real life application examples of multi-objective evolutionary algorithms. These application examples are conducted in the context of real-life problems. Some important applications are in computational finance, economics, engineering design, encryption and code breaking etc. Each example shows different particularities of the MOEA design, implementation and usage.

1. Financial Time Series

Niched Pareto Genetic Algorithm has been used to find patterns in financial time series such that predictions can be made regarding the behavior of a certain stock (Horn, Nafploitis & Goldberg, 1994). The methodology has also been used for the identification of significant technical analysis patterns in financial time series (Ruspini & Zwir, 1999). Two objectives are considered i.e. quality of fitness and its extent. Fitness measures the extent to which the time series values correspond to a financial uptrend, downtrend or head and shoulders interval.

2. Forecasting Stock Prices

Although long term forecasting is not possible for the stock market, it is normally possible to perform short term forecasting with heuristics. The use of genetic programming in this area has become increasingly popular, since GP can be used for symbolic regression, emulating the tasks traditionally performed by ANNs.

3. Stock Ranking

The aim of this problem is to classify stocks as strong or weak performers based on technical indicators and then use this information to select stocks for investment and for making recommendations to customers. Many MOEAs has been reported in this application area. Mullei and Beling (1998) used a GA with a linear combination of weights to select rules for a classifier system based on profitability.

4. Risk Return Analysis

It is slightly different from risk-return trade up which is made in investment portfolio. Credit portfolios handled by banks operate under different rules and therefore they are not modeled using the original Markowitz approach. Schlottmann and Seese (2002) used an approach similar to the NSGA-II for solving portfolio selection problems relevant to real-world banking (Deb, Pratap, Agrawal & Meyarivan, 2002). In the problem studied by the authors, a bank has a fixed supervisory capital budget. There is an upper limit for investments into a portfolio consisting of a subset of assets (e.g., loans to be given to different customers of the bank), each of which is subject to the risk of the default (capital risk). So, in this case, besides having an expected rate of return (as in the original Markowitz problem), each asset also has an expected default probability and a net exposure within a fixed risk horizon. The resulting problem has a discrete constrained search space with many local optima and two conflicting objective functions. Unlike the original NSGA-II, the authors adopted an external archive containing the non-dominated solutions found along the search. For validating the approach, the authors adopted data from the Credit- Metrics Technical Document.

5. Economic Modelling

Mardle uses a GA with a weighted goal programming approach to optimize a fishery bio-economic model (Mardle et al., 2000). Bio-economic models have been developed for a number of fisheries as a means of estimating the optimal level of exploitation of the resource and for assessing the effectiveness of the different management plans available.

6. Model Discovery

This is an interesting area in econometrics in which non-parametric models are assumed and one tries to use an evolutionary algorithm to derive a model for a certain type of problem (e.g., forecasting nonlinear time series). Normally, artificial neural networks (ANNs) have been used for the model itself, but several researchers have used evolutionary algorithms to find the most appropriate ANN that models the problem of interest.

7. Data Mining

The use of data mining techniques for learning complex patterns is a very promising research area in economics and finance. For example, the mining of financial time-series for finding patterns that can provide trading decision models is a very promising topic (Chen, 2002).

8. Investment Portfolio Optimization

One of the most promising fields of application is investment portfolio optimization. It can vary from simple portfolios held by individuals to huge portfolios managed by professional investors. The portfolio contains stocks, bank investments, real estate holdings, bonds, treasury bills etc. The motto of it is to find an optimal set to invest on, as well as the optimal investment for each asset. This optimal selection and weighting is a multi-objective problem where total profit of investment has to be maximized and total risk is to be minimized. There are also different constraints, depending on the type of problem to be solved. For example, the weights normally have lower bounds, upper bounds and many other constraints. This is the so-called optimal investment portfolio that one wishes to obtain by using optimization techniques. This problem is traditionally studied using the Markowitz portfolio selection model (Markowitz, 1952)

9. Risk Management

The study of risk and the reaction of an agent is a very interesting research area. Some researchers have studied, the formation process of risk preferences in financial problems (Chen, 2002)

10. Coevolution

The use of co-evolutionary approaches for certain problems in economics and finance (e.g. for studying artificial foreign exchange markets) is a very interesting topic that certainly deserves attention. Co-evolutionary MOEAs are still not too common, but their potential use in financial areas may boost the interest of researchers in paying more attention to them. Many other possible areas include, the study of consumers patterns, credit scoring, economic growth and auction games.

11. Air Operations Mission Planning

Air operations mission planning is a complex task, growing ever more complex as the number, variety, and interactivity of air assets increases. Mission planners are responsible for generating as close to optimal taskings of air assets to missions under severe time constraints. This function can be aided by decision-support tools to help to ease the search process through automation. Several applications of multi-objective evolutionary algorithms for discovering suitable plans in the air operations domain, including dynamic targeting for air strike assets, intelligence, surveillance, and reconnaissance (ISR) asset mission planning, and unmanned aerial systems (UAS) planning have been presented (Rosenberg, Richards, Langton, Tenenbaum & Stouch , 2008).

12. Survival Analysis

A multi-objective evolutionary algorithm for the extraction of models for survival analysis has been proposed and evaluated. To use multi-objective evolutionary algorithms for survival analysis has several advantages. They can cope with feature interactions, noisy data, and are capable of optimizing several objectives. This approach is capable of producing accurate models, even for problems that violate some of the assumptions made by classical approaches (Setzkorn, Taktak & Damato, 2006).

13. Engineering Design

Getting the most out of a range of materials to optimize the structural and operational design of buildings, factories, machines, etc. is a rapidly expanding application of GAs. These are being created for such uses as optimizing the design of heat exchangers, robot gripping arms, satellite booms, building trusses, flywheels, turbines, and just about any other computer-assisted engineering design application. There is work to combine GAs optimizing particular aspects of engineering problems to work together, and some of these can not only solve design problems, but also project them forward to analyze weaknesses and possible point failures in the future so these can be avoided.

14. Trip Traffic and Shipment Routing

New applications of a GA known as the "Traveling Salesman Problem" can be used to plan the most efficient routes and scheduling for travel planners, traffic routers and even shipping companies. The GA gives shortest routes for traveling, timing to avoid traffic tie-ups and rush hours, most efficient use of transport for shipping and including pickup loads and deliveries along the way. The program model all this in the background and improve productivity, while the human agents do other things.

15. Encryption and Code Breaking

On the security front, GAs can be used both to create encryption for sensitive data as well as to break those codes. Encrypting data, protecting copyrights and breaking competitors codes have been important in the computer world ever since there have been computers, so the competition is intense. Every time someone adds more complexity to their encryption algorithms, someone else comes up with a GA that can break the code. It is hoped that one day soon we will have quantum computers that will be able to generate completely indecipherable codes.

16. Optimizing Chemical Kinetic Analysis

GAs are proving very useful toward optimizing designs in transportation, aerospace propulsion and electrical generation. By being able to predict ahead of time the chemical kinetics of fuels and the efficiency of engines, more optimal mixtures and designs can be made available quicker to industry and the public. Some computer modeling applications in this area also simulate the effectiveness of lubricants and can pinpoint optimized operational vectors, and may lead to greatly increased efficiency all around well before traditional fuels run out.

17. Reservoir System Optimization

This study presents a novel approach for solving multiobjective reservoir system optimization problems using Differential Evolution (DE). The proposed methodology for Multi-objective Differential Evolution (MODE) combines pareto dominance criteria with DE for nondomination selection and crowded distance comparison operator for promoting solution diversity, and incorporates elitism in its evolution to improve the performance of the algorithm. The optimization involves minimization of flood risk, maximization of hydropower production, and minimization of irrigation deficits while properly evaluating other constraints. The MODE resulted in many Pareto optimal solutions in a single run, by specifying the reservoir releases and storage policy for each solution. The interdependence among the decision variables is better exploited using MODE. It is also found that the performance of MODE is better than NSGA-II for the reservoir system optimization problem. Thus, the obtained results suggest that the MODE approach is robust, and converging to the true Pareto optimal front with a good solution spread and coverage (Reddy & Kumar, 2007).

Summary

This paper gives a brief overview of multi-objective evolutionary algorithms. This paper describes need of multi-objective evolutionary algorithms, development of non-elitist and elitist multi-objective evolutionary algorithms, constrained multi-objective evolutionary algorithms with their salient issues. This paper also discusses the application of multi-objective evolutionary algorithms in several areas such as finance, engineering, economics, chemistry, transportation etc.

References

Chen, S.H. Ed. (2002). *Evolutionary computation in Economics and Finance*. Heidelberg: Physica-Verlag.

Coello, C.A.C & Toscano, G. (2000). *A micro-genetic algorithm for multi-objective optimization.* Technical Report Lania-RI-2000-06, Laboratoria Nacional de Informatica Avanzada, Xalapa, Veracruz, Mexico.

Cunha, A.G., Oleveira, P. & Covas, J.A. (1999). Genetic algorithms in multi-Objective Optimization Problems: An Application to Polymer Extrusion. *Proceedings of the Workshop on Multi-Criterion Optimization Using Evolutionary Methods held at Genetic and Evolutionary Computation Conference (GECCO-1999)*, USA, 129-130.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering,186*(2-4), 311-338.

Deb K. (2002). *Multi-objective optimization using evolutionary algorithms.* New York: John Wiley & Sons. Ltd.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Fonseca, C.M. & Fleming, P.J. (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, USA, 416-423.

Fonseca, C.M. & Fleming, P.J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms. Part-II: Application example. *IEEE transactions on Systems, Man and Cybernetics, 28*(1)*,* 38-47.

Fourman, M.P. (1985). Compaction of symbolic layout using genetic algorithms. In John J. Grefenstette (Ed.), *Proceedings of the first International Conference on Genetic Algorithms and Their Applications* (pp. 141-153). Mahwah, NJ.: Lawrence Erlbaum Associates.

Goldberg, D.E. & Wang, L. (1998). Adapting niching via coevolutionary sharing. In D. Quagliarella, J. Periaux, C. Ploni, and G. Winter (eds), *Genetic Algorithms and Evolution Strategies In Engineering and Computer Science* (pp. 21-38). Chichester, UK: Wiley.

Hajela, P. & Lin, C.Y. (1992). Genetic search strategies in multi-criterion design. *Structural Optimization, 4*(2)*,* 99-107.

Hajela, P., Lee, E. & Lin, C.Y. (1993). Genetic algorithms in structural topology optimization. In M.P. Bendsoe and C.A. Mota-Soares (eds.), *Proceedings of the NATO Advanced Research Workshop on Topology Design of Structures* (pp. 117-133). Drodrecht: Kluwer Academic Publishers.

Hiroyasu, T., Miki, M., & Watanabe S. (1999). Distributed genetic algorithms with a new sharing approach in multi-objective optimization problems. *Proceedings of the Congress on Evolutionary Computation (CEC-1999),*USA, 69-76.

Horn, J., Nafploitis, N. & Goldberg, D. (1994). A niched pareto genetic algorithm for multi-objective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation,* USA, 82-87.

Ishibuchi, H. & Murata, T. (1997). Minimizing the fuzzy rule and maximizing its performance by a multi-objective genetic algorithm. *Proceedings of the Sixth International Conference on Fuzzy Systems,* China, 259-264.

Jimenez, F., Verdegay, J.L. & Gomez-Skarmeta, A.F. (1999). Evolutionary techniques for constrained multi-objective optimization problems. *Proceedings of the Workshop on Multi-criterion Optimization Using Evolutionary Methods held at Genetic and Evolutionary Computation Conference (GECCO-1999),* USA, 115-116.

Knowles, J. D. & Corne, D.W. (2000). Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary Computation Journal, 8*(2), 149-172.

Kursawe, F. (1990). A variant of evolution strategies for vector optimization. *Proceedings of the Parallel Problem Solving from Nature 1st Workshop (PPSN-I),* USA, 193-197.

Laumanns, M. Rudolph, G., & Schwefel, H.P. (1998). A spatial predator-prey approach to multi-objective optimization: A preliminary study. *Proceedings of the Parallel Problem Solving from Nature 5th Workshop (PPSN-V),* Belgium, 241-249.

Mardle, S., Pascoe, S., & Tamiz, M. (2000). An investigation of genetic algorithms for the optimisation of multi-objective fisheries bioeconomic models. *International Transactions of Operations Research, 7*(1)*,* 33–49.

Mariano, C.E. & Morales, E.F. (2000). Distributed reinforcement learning for multiple objective optimization problems. *Proceedings of the Congress on Evolutionary Computation (CEC-2000),* USA, 188-195.

Markowitz, H.M. (1952). Portfolio Selection. *Journal of Finance, 1*(7)*,* 77-91.

Mullei, S. & Beling, P. (1998). Hybrid Evolutionary Algorithms for a Multiobjective Financial Problem. *Proceedings IEEE International Conference on Systems, Man, and Cybernetics,4,* 3925–3930.

Neef, M., Thierens, D. & Arciszewski, H. (1999). A case study of a multi-objective recombinative genetic algorithm with coevolutionary sharing. *Proceedings of the Congress on Evolutionary Computation (CEC-1999),* USA, 796-803.

Parks, G.T & Miller, I. (1998). Selective breeding in a multi-objective genetic algorithm. *Proceedings of the Parallel Problem Solving from Nature 5th Workshop (PPSN-V),* Belgium, 250-259.

Ray, T., Tai, K. & Seow, K.C. (2001). An evolutionary algorithm for multi-objective optimization. *Engineering Optimization, 33*(3)*,* 399-424.

Reddy, M.J & Kumar, D.N. (2007). Multiobjective differential evolution with application to reservoir system optimization. *Journal of Computing in Civil Engineering,* 136-146.

Rosenberg, B., Richards, M., Langton, J.T., Tenenbaum, S., & Stouch D.W. (2008). Applications of multi-objective evolutionary algorithms to air operations mission planning. *Proceedings of the GECCO conference companion on Genetic and evolutionary computation,* USA, 1879-1886.

Rudolph G. (2001). Evolutionary search under partially ordered fitness sets. *Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ISI 2001),* Dubai, 818-822.

Ruspini E.H. & Zwir I.S. (1999). Automated qualitative description of measurements. *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference,* Venice, Italy.

Schaffer, J.D. (1984). *Some experiments in machine learning using vector evaluated genetic algorithms.* Ph. D. Thesis, Nashville, TN: Vanderbilt University.

Schaffer, J.D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In John J. Grefenstette (Ed.), *Proceedings of the first International Conference on Genetic Algorithms and Their Applications* (pp. 93-100). Mahwah, NJ.: Lawrence Erlbaum Associates.

Schlottmann, F. & Seese D. (2002). Hybrid multi-objective evolutionary computation of constrained downside risk-return efficient sets for credit portfolio. *Proceedings of the 8th International Conference of the Society for Computational Economics. France.*

Sefrioui, M. & Periaux, J. (2000). Nash genetic algorithms: Examples and applications. *Proceedings of the Congress on Evolutionary Computation (CEC-2000),* USA, 509-516.

Setzkorn, C., Taktak, A.F.G. & Damato, D.E.(2006). On the use of multi-objective evolutionary algorithms for survival analysis, *Biosystems , 87*(1)*,* 31-48.

Srinivas, N. & Deb, K.(1994). Multiobjective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal, 2*(3)*,* 221-248.

Tapia, M.G.C. & Coello, C.A.C. (2007). Applications of multiobjective evolutionary algorithms in economics and Finance: A survey. *Proceedings of the Congress on Evolutionary Computation,* Mexico, 532-539.

Veldhuizen, D.V.(1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Ph.D. Thesis, Dayton, OH: Air Force Institute of Technology. Technical Report No. AFIT/DS/ENG/99-01

Zitzler, E. & Thiele, L. (1998). *An evolutionary algorithm for multi-objective optimization: The strength Pareto approach,* Technical report 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology. (ETH).

Zydallis, J.B., Veldhuizen, D.V. & Lamont, G.B. (2001). A statistical comparison of multi-objective evolutionary algorithms including the MOMGA-II. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001),* Brazil, 226-240.