AN EXPLANATION OF EVOLUTIONARY ALGORITHMS AND THEIR USES

Konduru Naveen Kumar Raju
Research Scholar
Department of Electronics & Communication Engineering
Sunrise University, Alwar, Rajasthan.
naveen.babji@gmail.com

Dr. Ram Mohan Singh Bhadoria
Research Guide
Department of Electronics & Communication Engineering
Sunrise University, Alwar, Rajasthan.

Abstract

This work presents evolutionary techniques for multi-objective optimization. Elitist and non-elitist multiobjective evolutionary algorithms are compared here. Constrained multiobjective evolutionary algorithms are also discussed.

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

Introduction

EAs are stochastic optimization approaches that imitate natural evolution. EAs date back to the late 1950s, and since the 1970s, evolutionary approaches such genetic algorithms, evolutionary programming, and evolution strategies have been presented. These methods use candidate solutions. Selection and variation modify this collection after substantial simplifications. Selection symbolizes living things competing for resources. Some are better at surviving and reproducing. Stochastic selection simulates natural selection in evolutionary algorithms.

Quality determines how many times any solution may duplicate. Hence, rating people and giving scalar fitness values determines quality. Another concept, variety, mimics nature's capacity to create "new" living things via recombination and mutation. These search algorithms are generic, resilient, and strong despite their basic principles. EAs may capture many pareto-optimal solutions in a single simulation run and exploit commonalities through recombination, making them ideal for multi-objective optimization.

Multi-objective optimization using evolutionary algorithms (EAs) is gaining popularity among scholars. Because of the requirement to reconcile vectorial performance measurements with EAs' fundamentally scalar reward system—number of offspring—most research has focused on EA selection. Evolutionary multiobjective optimization pioneered in the mid-1980s (Fourman, 1985; Schaffer, 1984; Schaffer, 1985). 1991–1994 saw many MOEA implementations suggested (Fonseca & Fleming, 1993; Hajela & Lin, 1992; Horn et al., 1994; Srinivas & Deb, 1994; Kursawe, 1990). Eventually, these methods and modifications were effectively applied to multiobjective optimization problems (Cunha et al., 1999; Fonseca & Fleming, 1998; Ishibuchi & Murata, 1997; Parks & Miller, 1998).

Which EA implementations fit various problems and what are their pros and cons?

• Unlike SOPs, trade-off front quality is harder to quantify. This may explain the scarcity of research in that region. Quantitative performance measures for multiobjective optimizers are yet undefined.

• No community-accepted test issues exist.

This makes comparing new and old algorithms challenging.

•Elitism, niching, and other MOEA ideas are independent of fitness assignment technique. These notions' advantages are unclear. Is elitism good for multi-objective search?

The study aims are based on the aforementioned issues:

• Analyzing current methods.

• Enhancing MOEAs, maybe creating a new evolutionary technique.

• Applying the best system design methodology to real-world challenges.

The first part examines the pros and cons of each strategy to better understand their impact and distinctions. Quantitative performance measurements that accommodate diverse quality requirements must be carefully defined. The fourth aim helps discover troublesome aspects that make MOEAs converge to the pareto-optimal front hardest. Population size and elitism are also compared in evolutionary search.

These studies may improve current approaches or lead to a new evolutionary way to sample the search space. Some applications are too complicated for precision optimization techniques.

This study compares evolutionary multi-objective optimization methods. It also identifies multi-objective optimization challenges in evolutionary search and how the approaches proposed handle them. Discussions suggest multi-objective evolutionary algorithm research areas.

Evolutionary Approaches to Multi-objective Optimization

The family of solutions of a multiobjective optimization problem is all search space elements that cannot enhance all objective vector components at once. Pareto optimality applies.

The pareto-optimal efficient or acceptable set is all pareto-optimal decision vectors. Non-dominated objective vectors correspond. Pareto-optimality is simply a first step in solving a multiobjective issue, which generally requires choosing a compromise solution from the non-dominated set based on preference information.

Due to their population-based approach, evolutionary algorithms are well-suited to multi-objective optimization problems. This enables the method to locate numerous members of the pareto optimum set in a single run instead of several runs like standard mathematical programming approaches. Evolutionary algorithms can handle discontinuous or concave pareto fronts, which mathematical programming methods struggle with. MOEA are interesting MOP solution strategies because they handle search and multi-objective decision making. They may scan partly sorted areas for several trade-offs. They identify several non-dominated solutions near to pareto-optimal.

Many goals concurrently define MOEA. Nevertheless task breakdown demonstrates minimal structural difference between MOEA and single objective EA.

General EA Tasks

- Initialize population
- Fitness evaluation (vector/ fitness transformation)
- Recombination
- Mutation
- Selection

Non- Elitist Multi-objective Evolutionary Algorithms

MOEAs without elite-preserving operators are non-elitist. Non-Elitist MOEA:

Vector Evaluated Genetic Algorithm (VEGA)

The simplest multi-objective GA is a basic modification of single-objective GA for multi-objective optimization. Schaffer's first multi-objective GA found non-dominated solutions (Schaffer, 1984). This GA assessed an objective vector with each element representing an objective function and prioritized excellent solutions for each function. Schaffer enabled all population solutions to cross-over to uncover intermediate trade-off solutions. VEGAs are computationally similar to single-objective GAs. VEGAs are simple and straightforward to implement. From a basic GA to a multi-objective GA requires just modest modifications and does not increase computing complexity. Each solution in a VEGA is assessed only using one objective function, hence it is not examined for additional objective functions, which are critical in multi-objective optimization.

Vector-Optimized Evolution Strategy

This method adapts single-objective self-adaptive evolution to multi-objective optimization. This method retains non-dominated solutions and eliminates crowded solutions using a dominance check and niching technique. Current researchers don't utilize the simulation findings since they were only presented for one issue (Kursawe, 1990).

Weight Based Genetic Algorithm

WBGAs must maintain weight vector variability among population members. WBGAs retain weight vector variety two methods. The first technique uses a niching method exclusively on the substring representing the weight vector, whereas the second evaluates carefully picked subpopulations for distinct pre-defined weight vectors, comparable to the VEGA. Converting a basic GA implementation to a WBGA is easy since WBGAs employ single-objective GAs. The algorithm is simpler than previous multi-objective evolutionary algorithms. For mixed objective functions (some to be reduced and others to be maximized), the WBGA utilizes a proportional selection technique on shared fitness values, making fitness function construction difficult. WBGA may also have trouble solving non-convex pareto-optimal issues (Hajela et al., 1993).

Multi-objective Genetic Algorithm

Fonseca and Fleming developed a multi-objective GA (MOGA) using the non-dominated categorization of a GA population (Fonseca & Fleming, 1993). This promotes non-dominated solutions and preserves their variety. The MOGA assigns fitness to each population solution differently than a tripartite GA. The algorithm continues as classical GA. The MOGA can solve various optimization issues since niching is done in goal space. The pareto optimal front and search space density may affect this technique.

Non-Dominated Sorting Genetic Algorithm

A fitness assignment scheme that prioritizes non-dominated solutions and a sharing approach that retains diversity among non-dominated front solutions sustain the dual goals in a multi-optimization algorithm in non-dominated sorting GA. Non-dominated sorting and sharing function implementation determine the fitness assignment procedure's computing cost. NSGAs assign fitness using non-dominated sets. Frontwise, NSGAs approach the pareto-optimal zone (Srinivas & Deb, 1994).

Predator-Prey Evolution Strategy

Predator-prey model fitness is used in this approach instead of dominance check. Its simplicity and lack of non-dominated solution emphasis are its key benefits. This technique lacks an explicit operator to disseminate solutions in the non-dominated set. Each predator must eliminate the poorest surrounding solution for a distinct purpose. Intermediate solutions are not maintained either (Laumanns et al., 1998)

Distributed Sharing GA

The distributed island model maintains non-dominated solution variety in this technique. Each island of the GA population undergoes genetic procedures separately. The non-dominated solutions of all island subpopulations are documented (Hiroyasu et al., 1999).

Distributed Reinforcement Learning Approach

Maraino and Morales proposed distributed reinforcement learning, where a family of agents is allocated to distinct goal functions (Mariano & Morales, 2000). Each agent provides an objective function-optimizing solution. Combining these solutions yields a non-dominated compromised set. Each non-dominated solution wins. In continuous search space issues, an agent searches in a specified direction from its present position. Agent objective functions assess the solution. The rewarding mechanism guides the algorithm toward the pareto-optimal area, the non-domination check preserves a varied group of solutions, and directional search helps identify new solutions in the search space by simultaneously creating many solutions.

Nash GA

This GA is inspired by a game theoretic technique in which one player is permitted to associate with each objective function and maximize its goal function while leaving others unaltered. When improvement stops, the Nash GA is terminated in a periodic sequence. This steady-state solution is a Nash-Equilibrium candidate pareto-optimal solution. This GA has superior convergence qualities than the NSGA, but an explicit niche-performing operator is needed to maintain several pareto-optimal solutions (Sefrioui & Periaux, 2000).

Elitist Multi-objective Evolutionary Algorithms

Elite-preserving operator evolutionary algorithms. EAs must preserve elite solutions or emphasize them. An elite preservation operator supports a population's elites by directly passing them on to the next generation. MOEA elitism might vary. Elitism may boost multi-objective EA performance, but it must be controlled. We describe multi-objective evolutionary optimization techniques that aim controlled elitism:

Rudolph's Elitist Multi-Objective Evolutionary Algorithms

Rudolph proposed a multi-objective evolutionary algorithm with diversity preservation (Rudolph, 2001) This method with a positive variation kernel of its search operators converges to the pareto-optimal front in finite trials in finite search space issues. The variation kernel's positivity guarantees that a limited number of trials will produce any child from any set of parent solutions. Consequently, if no pareto-optimal solution exists in any population, the positive variation kernel of the combined search operators guarantees that one such member will be formed in a limited number of trials. This individual cannot be removed from the population using the elite preservation procedure above. This method lacks solution variety, which is its fundamental drawback.

Elitist Non-Dominated Sorting Genetic Algorithm

Deb proposed an elitist non-dominated sorting GA (NSGA-II) with a diversity-preserving mechanism (Deb, 2000). NSGA-II complexity is at most O(MN 2). Crowding comparison during tournament selection and population reduction promotes non-dominated solution variety. No niching parameter is needed because solutions compete with crowding distances. This method also converges to the pareto-optimal solution set without the crowded comparison operator, but the population size grows with the generation counter. Elitism prevents deleting a pareto-optimal solution. Because crowded comparison restricts population size, this method loses convergence. In later generations, when more than N individuals belong to the first non-dominated set in the combined parent-offspring population, certain densely packed pareto-optimal solutions may give way to other non-dominated but non-pareto-optimal solutions. The method may generate pareto-optimal and non-pareto-optimal answers before settling to a well-distributed collection of pareto-optimal solutions.

Strength Pareto Evolutionary Algorithm (SPEA)

Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler and Thiele (1998) added elitism by explicitly maintaining an external population P. This population holds a set number of non-dominated solutions until the simulation begins. New non-dominated solutions are compared to the external population every generation and stored. The SPEA protects elites and employs them in genetic operations with the population to influence the population to seek in favorable places. Clustering improves SPEA non-dominated solution spread. Parameter-free clustering algorithms are appealing. SPEA fitness assignments are straightforward to compute and resemble Fonseca and Fleming's (1993) MOGA. SPEA may not converge to the pareto-optimal front if a large external population is utilized. Elite selection pressure will be high. Yet, a limited foreign population eliminates elitism. The SPEA fitness assignment penalizes external solutions that dominate more solutions. All dominated solutions close the dominant solution justify this assignment.

Pareto-Archived Evolution Strategy (PAES)

PAES's evolution approach determines a winner among various goals. The PAES controls pareto-optimal solution diversity. In search spaces with non-uniformly dense solutions, the PAES outperforms other approaches. PAES's depth parameter rapidly increases the number of hypercubes, making solution dispersion difficult to regulate. Multi-membered ES gives PAES a global viewpoint. This strategy does not ensure that the best non-dominated offspring solutions are stressed adequately as offspring are simply compared to the archive (Knowles, & Corne, 2000).

Multi-objective Messy Genetic Algorithm (MOMGA)

This clumsy GA extension suggests using m template strings per period (Veldhuizen, 1999). Level-1 MOMGA fills incomplete strings with m randomly selected template strings before the period begins. Several goal functions assess each full string. Selection operators employ this process's goal vector. At the conclusion of each era, the best answer for each goal function is chosen as the template string. Parallel MOMGA apps with separate beginning random templates suggested a concurrent MOMGA. After all MOMGAs, the external sets of non-dominated solutions are pooled and the best set is reported as the CMOMGA's non-dominated set. This concurrent technique may be useful in actual problem solving, even if its results lack the stability of an isolated MOMGA run. A probabilistically full MOMGA population initialization reduces computing load, improving earlier results (Zydallis et al., 2001).

Non-dominated sorting in Annealing GA (NSAGA)

NSAGA employs the Metropolis criteria and simulated annealing-like temperature decrease. The first-stage probability computation is the transfer probability from parent to offspring. The Metropolis criteria employs an energy function to calculate the second probability. An offspring population is approved elitistally if the Metropolis criteria with an updated temperature notion has a sufficient likelihood of forming it. This study modifies NSGA with a simulated annealing-like acceptance criteria to prove convergence.

Multi-objective Micro-GA

Multi-objective Micro-GA supports two populations. The elite population stores GA non-dominated solutions, whereas the GA population operates like the single-objective micro-GA. The elite archive receives innovative solutions like the PAES. The search space is grid-celled. The archive accepts or rejects new non-dominated solutions based on grid crowding (Coello & Toscano, 2000).

Elitist MOEA with Coevolutionary Sharing (ERMOCS)

Goldberg and Wang's coevolutionary sharing idea underpins this multi-objective GA (ERMOCS) (Goldberg & Wang, 1998; Neef et al., 1999). Coevolutionary shared niching (CSN) keeps non-dominated solutions diverse. A pre-selection strategy where a better offspring replaces a poorer parent solution in recombination preserves elites. In the coevolutionary model, customers and businessmen interact as in the CSN model, but an imprint operator emphasizes non-dominated solutions. After updating customer and businessman populations, each businessman is compared to a random sample of customers. Any consumer that dominates the rival businessman and is at least a crucial distance from other businesses replaces it. Hence, businessmen discover non-dominated consumer solutions. Over many generations, ERMOCS finds well-distributed consumer and businessman populations on a scheduling issue.

Constrained Multi-objective Evolutionary Algorithms

Constrained search and optimization challenges are typical. Several nonlinear limitations exist. We now consider different multi-objective evolutionary algorithms designed for restrictions.

Penalty Function Approach

Penalty functions add the constraint violation in an infeasible solution to each objective function. Optimizing penalized objective function values follows. This approach compares infeasible solutions by constraint violations for relatively high penalty terms compared to objective function values. For the same reason, a realistic option will almost always win. These traits enable population members to become viable from infeasible solutions and converge closer to actual pareto-optimal solutions.

Jimenez-Verdegay-Gomez-Skarmeta's Method

This study advised careful assessment of viable and infeasible solutions and niching to retain variation in pareto-optimal solutions. Binary tournament selection underpins this approach. Here, viable and infeasible alternatives are thoroughly examined to ensure that no infeasible option is fitter than any feasible one (Jimenez et al., 1999). The technique can handle any inequality constraint, however their analysis only considers lesser-than-equal-to restrictions. This technique may slow progress toward the viable zone by explicitly retaining variety among infeasible solutions. A user must specify two more settings. Large comparison checks reduce stochasticity in the non-domination check. The program also doesn't verify tournament dominance.

Constrained Tournament Method

This redefines dominance. Checking feasibility before comparing two dominance solutions. The practical solution beats the unfeasible one. Infeasible solutions with lower normalized constraint violations win. If both alternatives are practical, the standard dominating principle applies. This technique just requires constraint violation calculations, saving computational time. The constraint dominance concept works with any MOEA. No constraint management approach is required since it always dominates an infeasible solution.

Ray-Tai-Seow's Method

Ray, Tai, and Seow (2001) proposed a more complex constraint management method that does a non-domination check on all constraint violations instead of adding them. Three non-dominated sorting methods are utilized. The new population requires a non-dominated sorting of the goal functions, two of the constraint violation values, and a combined set of both. This algorithm handles infeasible solutions more carefully than previous limited handling methods and maintains population variety. Yet, the algorithm stagnates when all population members are viable and belong to a sub-optimal non-dominated front in a later generation. Three offspring are produced during crossover. Uniform crossover with equal likelihood of picking one variable value from each parent creates the first one. Blend crossovers apply a uniform probability distribution across a threshold parameter range to construct the other two options. Choosing operator parameter values is hard. Each crossover operation accepts five solutions, creating another issue. This will reduce population diversity. Three non-dominated ranking and head-count calculations make the approach more computationally demanding than others.

Salient issues of Multi-objective Evolutionary Algorithms

Several innovative methods have emerged from MOEAs' success in many problem areas. A good way to evaluate a novel algorithm is needed. An MOEA algorithm must be evaluated for its ability to converge close to the real pareto-optimal front and preserve a diversified group of non-dominated solutions. Unfortunately, one performance measure cannot fully assess both challenges.

Test a novel algorithm with issues with known search space complexity and pareto-optimal sets. The specific locations of pareto-optimal solutions aid algorithm search. Many MOEAs have been developed in recent years, prompting comprehensive comparison studies. Elite preservation, non-dominated solutions, and non-dominated solution variety were all successful MOEAs. Elite preservation helps converge and preserve a broad variety of non-dominated solutions, according to various research.

Diversifying non-dominated solutions requires space. The diversity preserving operator must consider choice variable space solution closeness. If variety is more essential, assess closeness in the objective space. Remember that closeness in one place does not guarantee proximity in the other. Especially in nonlinear and difficult issues.

Multi-objective optimization may tackle other optimization issues efficiently. A restricted single-objective optimization problem that optimizes the objective function and minimizes constraint violations is a multi-objective optimization problem. Goal programming is another issue that may benefit from numerous optimum solutions. Goal programming techniques employ relative weights of goals and discover one solution per weight vector since there is no optimization algorithm that can find several optimum solutions concurrently. When the number of goals increases, a considerable fraction of a randomly selected population becomes non-dominated, which is fascinating. This complicates elitism. While many population members are candidate elite solutions, few answers are adopted in any generation. There are several techniques to pick a population size that includes a suitable fraction of dominated fronts to first introduce diversity in the population.

MOEAs need convergence to the pareto-optimal front and excellent distribution. Despite theoretical convergence to the genuine pareto-optimal front, MOEAs do not ensure solution spread. Further study is needed to build MOEAs with convergence and solution spread features.

Applications of Multi-objective Evolutionary Algorithms

We briefly explore real-world multi-objective evolutionary algorithm applications. These examples solve real-world challenges. Computational finance, economics, engineering design, encryption, and codebreaking are major applications. Each example illustrates distinct MOEA design, implementation, and use features.

Financial Time Series

Niched Pareto Genetic Algorithm finds patterns in financial time series to forecast stock behavior (Horn, Nafploitis & Goldberg, 1994). Significant technical analysis patterns in financial time series have been identified using the approach (Ruspini & Zwir, 1999). Fitness quality and extent are examined. Fitness examines whether time series values match a financial uptrend, downturn, or head and shoulders interval.

Forecasting Stock Prices

Heuristics may be used for short-term stock market predictions, but not long-term. Genetic programming can emulate ANNs' symbolic regression tasks, making it prominent in this field.

Stock Ranking

This challenge classifies companies as strong or weak based on technical indicators and uses this knowledge to pick equities for investment and client recommendations. This application contains several MOEAs. Mullei and Beling (1998) selected classifier system rules based on profitability using a GA with linear weights.

Risk Return Analysis

Risk-return trade-up in investment portfolios differs. Bank credit portfolios are not modeled using Markowitz because they follow distinct criteria. Schlottmann and Seese (2002) solved real-world banking portfolio selection issues using an NSGA-II-like technique (Deb, Pratap, Agrawal & Meyarivan, 2002). The authors examined a bank's supervisory capital budget. Investments in a portfolio comprising a subset of assets (e.g., loans to bank customers) that are exposed to default risk have an upper limit (capital risk). Like with the original Markowitz problem, each asset has an anticipated rate of return, expected default probability, and net exposure within a given risk horizon. The discrete limited search space contains multiple local optima and two competing goal functions. The authors used an external repository of non-dominated solutions uncovered throughout the search, unlike the original NSGA-II. Credit-Metrics Technical Paper data was used to validate the technique.

Economic Modelling

Mardle optimizes a fisheries bio-economic model using a weighted goal programming GA (Mardle et al., 2000). Some fisheries employ bio-economic models to estimate the best resource utilization and evaluate management strategies.

Model Discovery

This intriguing field of econometrics assumes non-parametric models and uses an evolutionary approach to create a model for a specific situation (e.g., forecasting nonlinear time series). Several academics have employed evolutionary algorithms to determine the best artificial neural network (ANN) to simulate the issue.

Data Mining

Data mining for complicated pattern learning in economics and finance seems promising. Financial time-series mining to uncover trading decision models is an interesting field (Chen, 2002).

Investment Portfolio Optimization

Investment portfolio optimization has great potential. It ranges from small personal holdings to massive professional investment portfolios. Stocks, bank investments, real estate, bonds, treasury bills, etc. Its goal is to locate the best set of asset investments. Multi-objective optimum selection and weighting maximizes investment reward and minimizes risk. Problem types have varied limitations. Weights have lower, upper, and other limitations. Optimization methods provide this best investment portfolio. Markowitz portfolio selection is used to study this topic (Markowitz, 1952)

Risk Managem

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)

Coevolution

Co-evolutionary techniques for economics and finance challenges like artificial foreign exchange markets are fascinating and warrant research. Co-evolutionary MOEAs are rare, although their financial applications may pique researchers' curiosity. Consumer behavior, credit rating, economic development, and auction games are further possibilities.

Air Operations Mission Planning

As air assets expand in quantity, diversity, and interaction, mission planning becomes increasingly complicated. Mission planners provide near-optimal air asset taskings for missions with tight deadlines. Decision-support systems can automate this search function. Multi-objective evolutionary algorithms have been used to find acceptable air operations plans, including dynamic targeting for air attack assets, ISR asset mission planning, and UAS planning (Rosenberg, Richards, Langton, Tenenbaum & Stouch , 2008).

Survival Analysis

A multi-objective evolutionary method for survival analysis model extraction was designed and tested. Multi-objective evolutionary algorithms improve survival analyses. They can handle feature interactions, noisy data, and multi-objective optimization. This method can simulate issues that break conventional assumptions (Setzkorn, Taktak & Damato, 2006).

Engineering Design

GAs are increasingly used to improve structural and operational design of buildings, industries, machinery, etc. by optimizing material use. They are optimized for heat exchangers, robot grabbing arms, satellite booms, building trusses, flywheels, turbines, and other computer-assisted engineering design applications. There is effort to integrate GAs optimizing several parts of engineering challenges to address design difficulties and predict future weaknesses and point failures.

Trip Traffic and Shipment Routing

Travel planners, traffic routers, and freight businesses may now utilize the "Traveling Salesman Problem" GA to design the most effective routes and timing. The GA provides the quickest routes, time to avoid traffic jams, most effective shipment methods, and pickups and deliveries along the way. When humans work, the software models all this and boosts productivity.

Encryption and Code Breaking

GAs may encrypt and decrypt sensitive data. Computers have always been about encrypting data, safeguarding copyrights, and cracking rivals' codes, therefore rivalry is fierce. When encryption techniques get more sophisticated, someone else develops a Way to crack them. Quantum computers that produce indecipherable codes are expected shortly.

Optimizing Chemical Kinetic Analysis

GAs optimize transportation, aircraft, and electricity generation designs. By predicting fuel chemical kinetics and engine performance, industry and consumers can get better blends and designs faster. Certain computer modeling systems in this field mimic the efficacy of lubricants and can determine ideal operating vectors, which might lead to considerably higher efficiency before conventional fuels run out.

Reservoir System Optimization

This paper introduces Differential Evolution to solve multiobjective reservoir system optimization challenges (DE). Multi-objective Differential Evolution (MODE) uses pareto dominance criteria for nondomination selection, crowded distance comparison operator for solution diversity, and elitism to increase algorithm performance. Optimizing flood risk, hydropower production, and irrigation shortages while considering other restrictions is the goal. By selecting reservoir releases and storage policies, the MODE produced multiple Pareto optimum solutions in one run. MODE maximizes decision variable interdependence. MODE outperforms NSGA-II for reservoir system optimization. Hence, the MODE technique seems resilient and converges to the genuine Pareto optimum front with excellent solution spread and coverage (Reddy & Kumar, 2007).

Summary

This paper briefly discusses multi-objective evolutionary algorithms. This study discusses multi-objective evolutionary algorithms, their benefits, and their drawbacks. This study also examines multi-objective evolutionary algorithms in finance, engineering, economics, chemistry, transportation, etc.

References

1. 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,
2. Veracruz, Mexico.
3. 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.
4. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering,186(2-4), 311-338.
5. Deb K. (2002). Multi-objective optimization using evolutionary algorithms. New York: John Wiley & Sons. Ltd.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. Hajela, P. & Lin, C.Y. (1992). Genetic search strategies in multi-criterion design. Structural Optimization, 4(2), 99-107.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.

21. *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.*

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

23. *Mullei, S. & Beling, P. (1998). Hybrid Evolutionary Algorithms for a Multiobjective Financial Problem.*

24. *Proceedings IEEE International Conference on Systems, Man, and Cybernetics,4, 3925–3930.*

25. *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.*

26. *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.*

27. *Ray, T., Tai, K. & Seow, K.C. (2001). An evolutionary algorithm for multi-objective optimization.*

28. *Engineering Optimization, 33(3), 399-424.*

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

30. *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.*

31. *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.*

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

33. *Schaffer, J.D. (1984). Some experiments in machine learning using vector evaluated genetic algorithms.*

34. *Ph. D. Thesis, Nashville, TN: Vanderbilt University.*

35. *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.*

36. *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.*

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

38. *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.*

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

40. *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.*

41. *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*

42. *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).*

43. *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.