The multi-objective genetic algorithm based techniques for intrusion detection

Dadmehr Javadi Meraai
Department of Computer Application, Al maktum University, Lebanon

Summary
The Multi Objective Genetic Algorithms (MO- GAs) are one of the most widely used techniques that have the capability to find the solution to the problem having multiple conflicting objectives like Intrusion De- tection. It is a population based technique capable of producing a set of non-inferior solutions that exhibit the classification trade-offs for the user. This capabil- ity of MOGA can be exploited for generating optimal base classifiers and ensembles thereof for Intrusion De- tection. This paper explores the various MOGAs proposed in the literature along with their pros and cons. The motivation for the use of MOGA and its issues are high- lighted. Finally, the chapter highlights the concluding remarks.
Keywords:
*Genetic algorithm, Intrusions, Intrusion, Detection, Network Security, Security Threats***.***

1. Introduction

In the recent past, multi-objective optimization techniques have been successfully utilized to solve the problems having multiple conflicting objectives in spite of their computational expenses. The availability of fast machines and computational models has boosted the use of these techniques to solve many problems [32]. The most widely used such technique is the Genetic algorithm (GA). GA is a nature inspired search technique whose working is based on Darwin's theory of the survival-of-the-fittest [16]. It can be used to solve the optimization problems having single or multiple conflicting objectives [19, 4]. It is a population based technique that works with a set of solutions rather than a single solution and generate a set of candidate solutions.
Article overview: following this introduction, section 2 highlights motivations for use of MOGA for intrusion detection. Section 3 introduces various MOGA based algorithms. Section 4 presents the important implementation of MOGA for intrusion detection. Section 5 highlights the key issues in using MOGA for intrusion detection. Finally, the paper concludes the current scenario of MOGA based techniques for intrusion detection.

2. Motivation for MOGA

MOGAs have become increasingly popular in a wide variety of application domains like engineering, industrial and scientific [6]. Many researchers employed the GA to solve the problems in the field of multi-objective optimization [4, 14, 15]. They used GA in two different ways. The first way to solve a multi-objective problem is to convert multiple objectives into a single objective [3, 10]. The single objective is further optimized by GA to produce single solution. Generally priori knowledge about the problem, or some heuristics guide the GA to produce a single solution. By changing the parameters of the algorithm and executing the algorithm repeatedly, more solutions can be produced. This approach has several limitations for multi objective optimization problems. The second way to solve multi objective optimization problems by using GA produces a set of non-inferior (non-dominated) solutions. This set of non-inferior solutions represents trade-offs between multiple objectives which is identified as a Pareto optimum front [17, 27]. By incorporating domain knowledge, the user can select a desired solution. Here, GA has produced a set of solutions in Pareto front in a single run without incorporating any domain knowledge or any other heuristic about the problem. Some of the important researches in developing multi-objective Genetic algorithms (MOGAs) are Strength Pareto Evolutionary Algorithm (SPEA2) [36], Pareto-Envelope based Selection Algorithm (PESA-II)[8], Non-dominated Sorting Genetic Algorithm (NSGA-II) [11], Archive based Micro Genetic Algorithm 2 [33] and many more. A comprehensive review of various MOGAs can be further referred in [4], [3] and [10].

3. Algorithms

The important MOGAs proposed in the literature are described in subsequent subsections:

Manuscript received March 5, 2016
Manuscript revised March 20, 2016

3.1 VEGA

The first implementation of MOGA is called Vector Evaluated Genetic Algorithm (VEGA), incorporates a modified selection process to cope with multiple evaluation criteria [29]. [29] proposed a modified selection procedure to handle multiple objectives. He suggested dividing the whole population into groups equal to a number of objectives. Selection procedure in each group is based on a single objective. Mating limits help limited combinations of individual solutions in the same group. Pairwise comparison helps to recognize the dominated solutions. After a few generations, a set of non-dominated solutions are recognized to represent the Pareto front [17]. But, the major difficulty with this algorithm is that it prevents to determine the location of the Pareto front. Another problem of the selection proce- dure, the individual solutions that are better in one objective are given preference over the other individ- ual solutions. This leads the algorithm to converge to individually best solutions only [10].

3.2 MOGA

In 1993, another implementation of MOGA was proposed which employed the concept of niching and dominance along with the rank based fitness assignment. The non-dominated solutions are categorized into groups. The individual solutions are assigned same ranks in each group. The other groups of the solutions which are dominated by the current group are assigned next ranks. To maintain the diversity among the groups, the author proposed to use dynamically updated sharing. But, the major problem of this method is slow conver- gence that prevents from finding the optimum Pareto front [10, 17].

3.3 NSGA and its variants

NSGA: Similar to MOGA, [31] proposed an algorithm called Non dominated Sorting Genetic Algorithm (NSGA) based on the concept of dominance and sharing. The NSGA is based on several layers of classification of individuals as suggested by [19]. Be- fore selection is performed, the population is ranked on the basis of non-domination: all non dominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the di- versity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored, and another layer of non dominated individuals is con- sidered. The process continues until all individuals in the population are classified. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the popula- tion. The diversity among the individual solutions is maintained by using the sharing concept. How- ever, NSGA does not involve dynamic updating of any niche that makes it faster than MOGA. The al- gorithm of the NSGA is not very efficient, because Pareto ranking has to be repeated again. Evidently, it is possible to achieve the same goal in a more efficient way.

NPGA: [23] proposed a Niched Pareto Genetic Algorithm (NPGA) based on the concept of dominance and sharing. NPGA differs from earlier approaches in the selection of individual solutions. Here, the se- lection is based on a modified tournament selection than the proportional selection (as in NSGA). The basic idea of the algorithm is quite clever: two indi- viduals are randomly chosen and compared against a subset of the entire population (typically, around 10% of the population) [6]. If one of them is dom- inated (by the individuals randomly chosen from the population) and the other is not, then the non- dominated individual wins. All the other situations are considered as a tie (i.e., both competitors are ei- ther dominated or non-dominated). When there is a tie, the result of the tournament is decided through fitness sharing.

NSGA-II: Another fast, elitist algorithm called NSGA- II is proposed by [11] as a version of NSGA proposed by [31]. NSGA-II is a generational algorithm that works upon the concept upon dominance. Instead of sharing, NSGA-II uses the crowding distance to maintain the diversity among the individual solu- tions. Here, the author proposed to use tournament selection strategy for selection of individual solu- tions. In this algorithm, to sort a population of as- signed size according to the level of non-domination, each solution must be compared with every other so- lution in the population to find if it is dominated. Solutions of the first non-dominated front are stored in the first Pareto front, solutions of the second front on the second Pareto front and so on. The new pop- ulation is constituted by solutions on the first Pareto front, if they are less than the initial population size: solutions from the next front are taken according to their ranks. In the NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions to which it dominates [6]. The NSGA-II estimates the density of solutions sur- rounding a particular solution in the population by computing the average distance of two points on ei- ther side of this point along each of the objectives of the problem. This value is the so-called crowd- ing distance. During selection, the NSGA-II uses a crowded-comparison operator which takes into con- sideration both the non-domination rank of an in- dividual in the population and its crowding dis- tance (i.e., non-dominated solutions are preferred over dominated solutions, but between two solu- tions with the same non-domination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not use an external memory as the other MOEAs previously discussed. Instead, the elitist mechanism of the NSGA-II consists of com- bining the best parents with the best offspring ob- tained (i.e. non-dominated solutions are preferred over dominated solutions, but between two solutions with the same non-domination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not use an external memory as the other MOEAs previously discussed. Instead, the eli- tist mechanism of the NSGA-II consists of combin- ing the best parents with the best offspring obtained (i.e. a ($\mu + \lambda$) selection). Due to its clever mech- anisms, the NSGA-II is much more efficient (com- putationally speaking) than its predecessor, and its performance is so good, that it has become very popular in the last few years, becoming a landmark against which other multi-objective evolutionary al- gorithms have to be compared.

3.4 SPEA and its variants

SPEA: Several researches were carried to improve the performance of VEGA, NPGA and NSGA. One such algorithm was proposed by [36] called the Strength Pareto Evolutionary Algorithm (SPEA). SPEA is an elitist MOGA where elitism helps to improve its convergence properties. Here, they proposed to maintain an archive of non-dominated solutions from the beginning of the algorithm. The archive is ex- ternal to the main population, and it takes part in fitness computation. With the use of the archive, its size may increase very large so its pruning may be done to keep it in limits. Limited archive size helps in the selection of individual solutions.
SPEA uses an archive containing non-dominated solutions previously found (the so-called external non-dominated set) [6]. At each generation, non- dominated individuals are copied to the external non-dominated set. For each individual in this exter- nal set, a strength value is computed. This strength is similar to the ranking value of MOGA [18], since it is proportional to the number of solutions to which a certain individual dominates. In SPEA, the fitness of each member of the current population is com- puted according to the strengths of all external non- dominated solutions that dominate it. The fitness assignment process of SPEA considers both close- ness to the true Pareto front and even distribution of solutions at the same time. Thus, instead of using Niches based on distance, Pareto dominance is used to ensure that the solutions are properly distributed along the Pareto front. Although this approach does not require a niche radius, its effectiveness relies on the size of the external non-dominated set. Since, the external non-dominated set participates in the selection process of SPEA, if its size grows too large, it might reduce the selection pressure, thus slowing down the search. Because of this, the authors de- cided to adopt a technique that prunes the contents of the external non-dominated set so that its size remains below a certain threshold.

SPEA2: In SPEA2 proposed by [37], an improved method for pruning the size of the archive that retain the boundary solutions in the archive. SPEA2 involves
a fine grained fitness function based number of in- dividual solutions that dominate a current solution and how many it dominates. Fitness function also includes density information based on a k-NN al- gorithm. SPEA2 is found to be superior in perfor- mance than NSGA-II, especially in high dimensional search spaces [37].

SPEA2 has three main differences with respect to its predecessor [37]: (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals by which it is dominated; (2) it uses a nearest neighbor den- sity estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

3.5 PAES

Pareto Archived Evolutionary Strategy (PAES) algo- rithm was proposed by [24]. It is a simple multi-objective evolutionary algorithm using a single parent- single child strategy. In this strategy, binary strings and bitwise mutation are used to create children in replacement of real parameters. PAES algorithm consists of a (1 + 1) evolution strategy (i.e., a single parent that gener- ates a single offspring) in combination with a historical archive that records the non-dominated solutions pre- viously found [6]. This archive is used as a reference set against which each mutated individual is being com- pared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides the objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its coordinates or geographical location). A map of such a grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space).

3.6 PESA and its variants

PESA: Pareto Envelope based Selection Algorithm (PESA) was proposed by [9] which uses a small internal pop- ulation and a large external population. The di- versity is maintained by borrowing the concept of the hyper grid division of the phenotype space from PAES. But, the selection process is per- formed by the crowding distance method. In PESA, the exter- nal population plays an important role as it is con- sidered to determine selection as well as to maintain diversity among the solutions.

PESA-II: The PESA is further revised to a new version called PESA-II [8]. The PESA-II uses region based selection. In the region based selection, the unit of selection is hyper box rather than an individual. Further, an individual is selected randomly from the hyperbox. The main objective of a PESA set of al- gorithms is to reduce the computational overhead associated with other representative methods.

3.7 AMGA and its variants

In Micro Genetic Algorithm (MGA) was originally pro- posed by [5]. It is a GA with a small population and re-initialization process. The working of the MGA in- volves the generation of random population, which gets loaded into memory in two different portions named replaceable and non-replaceable portion. The contents of replaceable portion get changed after each cycle of MGA whereas the contents of non-replaceable portion never changes during execution of the algorithm. The population of MGA is randomly taken as a mixture of individuals from both the portions. During each iter- ation, the algorithm experiences genetic operators. At the end of each iteration, two non-dominated individu- als from final population are selected to compare with contents of the external memory. In this way, all domi- nated solutions from the external memory are removed. The MGA uses three forms of elitism:

1. It retains non-dominated solutions found within the internal iteration of the algorithm.
2. It uses a replaceable portion of the memory whose contents are partly restored at certain intervals.
3. It exchanges the population of the algorithm by nominal solutions created.

The important implementation of the micro genetic al- gorithm are AMGA [34] and its enhancement AMGA2 [33].

AMGA: The Archive based Micro Genetic Algorithm (AMGA) is a constrained multi-objective evolution- ary optimization algorithm [34]. It is a generational genetic algorithm since during a particular iteration (generation), only solutions created before that it- eration takes part in the selection process. AMGA uses genetic variation operators such as crossover and mutation to create new solutions. For the pur- pose of selection, AMGA uses a two tier fitness as- signment mechanism; the primary fitness is the rank which is based on the domination level and the sec- ondary fitness is based on the diversity of the solu- tions in the entire population. This is in contrast to NSGA-II, where diversity is computed only among the solutions belonging to the same rank. The AMGA generates a very small number of new solutions at each iteration, and can be classified as a micro- GA. Generation of a very small number of solu- tions at every iteration helps in reducing the num- ber of function evaluations by minimizing explo- ration of less promising search regions and direc- tions. The AMGA maintains an external archive of good solutions obtained. Use of the external archive helps AMGA in reporting a large number of non- dominated solutions at the end of the simulation. It also provides information about its search history which is exploited by the algorithm during the se- lection operation. At each iteration, the parent pop- ulation is created from the archive and the binary tournament selection is performed on the parent population to create the mating population. The off- spring population is created from the mating pool, and is used to update the archive. The size of the archive determines the computational complexity of the AMGA. The design of the algorithm is indepen- dent of the encoding of the variables and thus the proposed algorithm can work with almost any kind of encoding (so long as suitable genetic variation operators are provided to the algorithm). The algo- rithm uses the concept of Pareto ranking borrowed from NSGA-II and includes improved diversity com- putation and preservation techniques. The diversity measure is based on efficient nearest neighbor search and modified crowding distance formulation [11].

AMGA2: An improved Archive-based Micro Genetic Algorithm (referred to as AMGA2) for constrained multi-objective optimization is proposed in [33]. AMGA2 is designed to obtain fast and reliable convergence on a wide variety of optimization problems. AMGA2 benefits from the existing literature in that it bor- rows and improves upon several concepts from the existing multi- objective optimization algorithms. Im- provements and modifications to the existing diver- sity assessment techniques and genetic variation op- erators are also proposed. AMGA2 employs a new kind of selection strategy that attempts to reduce the probability of exploring un-desirable search re- gions. The proposed

AMGA2 is a steady-state ge- netic algorithm that maintains an external archive of the best and diverse solutions and a very small working population. AMGA2 has been designed to facilitate the decoupling of the working population, the external archive, and the number of solutions desired as the outcome of the optimization process.
Comprehensive benchmarks and comparison of AMGA2 with the current state-of-the-art multi-objective op- timization algorithms demonstrate its improved search capability.

4. Key points/issues in designing an efficient MOGA

The literature review of various multi-objective genetic algorithms (MOGAs) indicates that following key points are used for designing them.

– Population Approach: It is observed from recent re- searches that following concepts are used for popu- lation in MOGAs [33]:

1. Dynamic population size
2. Small population size
3. External archive

The use of dynamic and small population size as- sists in reducing the number of function evaluations. The reduced function evaluations help to converge it faster [7]. The use of external archive helps to store a large number of non-dominated solutions that approximate optimum Pareto front accurately [37]. These concepts can be adapted in a single al- gorithm to enhance its performance.
– Selection approach: The most widely used approaches for selection mechanism are non-dominated sorting [12] and the improved strength Pareto approach [37]. The latter approach involves the number of solu- tions an individual solution dominates and number of individual solutions it is dominated by for selec- tion mechanism. For efficient classification of the solutions, multi level fitness is employed based on domination level and diversity.

– Diversity determination: Several methods are sug gested to determine the diversity of the solutions. The most widely used methods involve fitness-sharing [20], crowding distance [12, 13], K-mean clustering [21], -domination, cell-based (hyper-grid) meth- ods [9, 21, 35], and fast pruning of crowded solutions using efficient nearest neighbor search [30, 25]. De- pending on the requirement of the problem, quantitative measure (like crowing distance) or non-quantitative measure (like the pruning of crowding solutions) can be employed.
Variation Operators: The most important proper- ties of variation operators are as follows:

1. Parent-centric property
2. Self-Adaptivity
3. Invariance to affine transformations of the search space
4. Disrupts to impart random behavior and resilience to pre-mature convergence

Some of the cross over operators like uni-modal normal distribution crossover [22], simulated binary crossover, and parent-centric crossover [12] are proposed to deal real variables and have some sort of parent- centric property. But, these operators have the limi tation of self-adaptive. Whereas, the differential evo- lution (DE) [26] operator exhibits the self-adaptive property.
In addition to these points, [6] listed many chal- lenges in the current set of MOGA (a class of MOEA- Multi Objective Evolutionary Algorithm). He highlighted the important challenges as follows:
1. Parameter control is a topic that has been only hardly explored in MOEAs. Is it possible to design an MOEA that self-adapts its parameters such that the user doesn't have to fine-tune them by hand?
2. What is the minimum number of fitness function evaluations that are actually required to achieve a minimum desirable performance with an MOEA? Recently, some researchers have proposed the use of black-box optimization techniques normally adopted in engineering to perform an incredibly low number of fitness function evaluations while still producing reasonably good solutions. However, this sort of ap- proach is inherently limited to the problems of low dimensionality. So, the question is: are there any other ways of reducing the number of evaluations without sacrificing dimensionality?
3. More research is required for development & imple- mentation of MOEAs that are independent of the platform and programming language in which they were developed. This is an important step towards a common platform that can be used to validate the new algorithms.
4. How to deal with the problems having multiple ob- jectives? Some recent studies have shown that tra- ditional Pareto ranking schemes do not behave well in the presence of many objectives (where ”many” is normally a number above 3 or 4).
5. There are plenty of fundamental questions that re- main unanswered. For the example: what are the sources of difficulty of a multi-objective optimiza- tion problem for a MOEA? What are the dimen- sional limitations of the current MOEAs? Can we use alternative mechanisms into an evolutionary al- gorithm to generate non-

dominated solutions with- out relying on Pareto ranking (e.g., adopting con- cepts from game theory).

5. MOGA based techniques for intrusion detection

Several researchers employed MOGA for finding a set of non-dominated solutions for the problem of intrusion detection. Such initiative was carried by [28] by suggesting an evaluation function which was later known as Parrot Function. He proposed to use accuracy of each target class as a separate objective in their evaluation function for MOGA. Here, accuracy of each class refers to correctly classified instances of that class. The Parrot function was further adopted by [1, 2] to generate an ensemble of base classifiers. The generation of the ensemble was completed in two stages using modified NSGA-II [12]. In the first stage, a set of base classifiers was generated. Second stage optimized the combina- tion of base classifiers using a fixed combining method. Both of these methods differ in their function evalua- tion. The former study of [1] proposed to optimize the classifiers by minimizing the aggregated error of each class and maximize diversity among them. Since, the error on each class is not treated as separate objectives, this is similar to a general error measure such as MSE, which will have the same issues as the implementation of [28], being biased towards the major class (es). In the second phase of the approach proposed by [1, 2], the objectives are to minimize the size of the ensemble and maximize the accuracy. Consequently, the drawback of their approach is to create a single best solution based on general performance metrics. The same concept was further extended by [17] by conducting similar exper- iments with different evaluation functions for creating an ensemble of ANNs as base classifiers in the presence of imbalanced datasets using NSGA-II. He used 3-class classification by using ANNs and MOGA. He proved that MOGA based approach is an effective way to train the ANN which works well for minority attack classes in imbalanced datasets. He proposed two phase process for intrusion detection. In the first phase, he generated a set of base classifiers of ANNs by optimizing their weights assuming a fixed number of hidden layers and the number of neurons per hidden layer in ANN. The second phase generates improved non-dominated front of ensemble solutions based upon base ANN solutions optimized in phase 1.

6. Conclusions

The MOGA has been successfully employed to solve the problems of many domains having multiple conflict- ing objectives. This chapter introduced the motivation for use of MOGAs, various algorithms proposed in the literature and issues of designing an efficient MOGA. The use of MOGA for creating diverse base classifiers and their ensembles is described. This is followed by the proposed MOGA based ensemble technique for in- trusion detection. The proposed multi-objective genetic algorithm based technique can learns successfully from a benchmark dataset. It offers the user a pool of so- lutions that exhibit different trade-offs in their perfor- mance. The proposed technique has three phases. In phase 1, the proposed technique presents an implicit mechanism for generating a diverse pool of classifiers that can formulate the base classifiers for the ensembles. In phase 2, ensembles are created by approximating an improved Pareto front of ensemble solutions over that of phase 1. The proposed technique successfully takes the advantage of multi-objective techniques that a good ap- proximation of the true Pareto front of non-dominated solutions is obtained in a single run. In phase 3, pre- dictions of selected base classifiers are fused together to compute the final prediction of the ensemble using the majority voting method. The Pareto analysis done in this work presents a novel perspective on the genera- tion and selection of the base classifiers and ensembles thereof. Instead of combining all base classifiers or re- duced set of base classifiers, only those base classifiers are combined which perform better in their training for the final ensemble. The proposed technique is a gener- alized classification technique that is applicable to the problem of any field having multiple conflicting objec- tives and a dataset can be represented in the form of labeled instances in terms of its features.

References

[1] Ahmadian, K., Golestani, A., Analoui, M., Jahed, M.: Evolving ensemble of classifiers in low-dimensional spaces using multi-objective evolutionary approach. In: Proc. of 6th IEEE/ACIS International Conference on Com- puter and Information Science (ICIS), pp. 217–222. IEEE (2007)

[2] Ahmadian, K., Golestani, A., Mozayani, N., Kabiri, P.: A new multi-objective evolutionary approach for creating ensemble of classifiers. In: Proc. of IEEE International Conference on Systems, Man and Cybernetics (ISIC), pp. 1031–1036. IEEE (2007)

[3] Coello, C.: An updated survey of ga-based multiobjec- tive optimization techniques. ACM Computing Surveys (CSUR) 32(2), 109–143 (2000)

[4] Coello, C., et al.: A comprehensive survey of evolutionary-based multiobjective optimization tech-

niques. Knowledge and Information systems 1(3), 129–156 (1999)

[5] Coello, C.C., Toscano, P.G.: A micro-genetic algorithm for multiobjective optimization. In: Proc. of Evolution- ary Multi-Criterion Optimization, pp. 126–140. Springer (2001)

[6] Coello Coello, C.: Evolutionary multi-objective optimization: a historical view of the field. Computational Intel- ligence Magazine, IEEE 1(1), 28–36 (2006)

[7] Coello Coello, C., Pulido, G., Montes, E.: Current and future research trends in evolutionary multiobjective optimization. Information Processing with Evolutionary Algorithms pp. 213–231 (2005)

[8] Corne, D., Jerram, N., Knowles, J., Oates, M., et al. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO2001). Citeseer (2001)

[9] Corne, D., Knowles, J., Oates, M.: The pareto envelopebased selection algorithm for multiobjective optimiza- tion. In: Proc. of Parallel Problem Solving from Nature PPSN VI, pp. 839–848. Springer (2000)

[10] Deb, K.: Multi-objective optimization. Multi-objective optimization using evolutionary algorithms pp. 13–46 (2001)

[11] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. Lecture notes in computer science 1917, 849–858 (2000)

[12] Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evolutionary computation 10(4), 371–395 (2002)

[13] Deb, K., Tiwari, S.: Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. European Journal of Operational Research 185(3), 1062–1087 (2008)

[14] Deb, K., et al.: Evolutionary algorithms for multicriterion optimization in engineering design. Evolution- ary Algorithms in Engineering and Computer Science pp.135–161 (1999)

[15] Dimopoulos, C., Zalzala, A.: Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. IEEE Transactions on Evolutionary Computation, 4(2), 93–113 (2000)

[16] Eldredge, N., Eldridge, N.: Macroevolutionary dynamics: species, niches, and adaptive peaks. McGraw-Hill New York (1989)

[17] Engen, V.: Machine learning for network based intrusion detection: an investigation into discrepancies in findings with the kdd cup'99 data set and multi-objective evolution of neural network classifier ensembles from imbalanced data. Ph.D. thesis, Bournemouth University (2010)

[18] Fonseca, C., Fleming, P., et al.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Proc. of the fifth international conference on genetic algorithms, vol. 1, p. 416. San Mateo, California (1993)

[19] Goldberg, D., Holland, J.: Genetic algorithms and machine learning. Machine Learning 3(2), 95–99 (1988)

[20] Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proc. of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, pp. 41–49. L. Erlbaum Associates Inc. (1987)

[21] Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome research 9(11), 1106–1115 (1999)

[22] Higuchi, T., Tsutsui, S., Yamamura, M.: Theoretical analysis of simplex crossover for real-coded genetic algorithms. In: Parallel Problem Solving from Nature PPSN VI, pp. 365–374. Springer (2000)

[23] Horn, J., Nafpliotis, N., Goldberg, D.: A niched pareto genetic algorithm for multiobjective optimization. In: Proc. of the First IEEE Conference on Evolutionary Computation, pp. 82–87. IEEE (1994)

[24] Knowles, J., Corne, D.: The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: Proc. of the 1999 Congress on Evolutionary Computation (CEC), vol. 1. IEEE (1999)

[25] Kukkonen, S., Deb, K.: A fast and effective method for pruning of non-dominated solutions in many-objective problems. Lecture Notes in Computer Science 4193, 553–562 (2006)

[26] Kukkonen, S., Lampinen, J.: Gde3: The third evolution step of generalized differential evolution. In: Proc of the 2005 IEEE Congress on Evolutionary Computation, vol. 1, pp. 443–450 (2005)

[27] Kumar, G., Kumar, K.: The use of multi-objective genetic algorithm based approach to create ensemble of ann for intrusion detection. International Journal of Intelligence Science 2(24), 115–127 (2012). DOI 10.4236/ijis.2012.224016

[28] Parrott, D., Li, X., Ciesielski, V.: Multi-objective techniques in genetic programming for evolving classifiers. In: Proc. of IEEE Congress on Evolutionary Computation, vol. 2, pp. 1141–1148. IEEE (2005)

[29] Schaffer, J.: Multiple objective optimization with vector evaluated genetic algorithms. In: Proc. of the 1st international Conference on Genetic Algorithms, pp. 93–100. L. Erlbaum Associates Inc. (1985)

[30] Soleymani, M., Morgera, S.: An efficient nearest neighbor search method. IEEE Transactions on Communications 35(6), 677–679 (1987)

[31] Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolution- ary computation 2(3), 221–248 (1994)

[32] Tiwari, S.: Development and integration of geometric and optimization algorithms for packing and layout design. Ph.D. thesis, Clemson University (2009)

[33] Tiwari, S., Fadel, G., Deb, K.: Amga2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. Engineering Optimization 43(4), 377–401 (2011)

[34] Tiwari, S., Koch, P., Fadel, G., Deb, K.: Amga: an archive-based micro genetic algorithm for multi-objective optimization. In: Proc. of Genetic and Evolutionary Computation conference (GECCO-2008), Atlanta, USA, pp. 729–736 (2008)

[35] Yen, G., Lu, H.: Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. IEEE Transactions on Evolutionary Computation 7(3), 253–274 (2003)
[36] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 8(2), 173–195 (2000)
[37] Ziztler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. Evolutionary Methods for Design, Optimization, and Control pp. 95–100 (2002).