Supplementary Information
Classification of Two and Multi-Class Cancer Data Reliably Using Multi-Objective Evolutionary Algorithms

Kalyanmoy Deb and A. Raji Reddy

Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
{deb,arreddy}@iitk.ac.in
http://www.iitk.ac.in/kangal

1 INTRODUCTION TO EVOLUTIONARY PRINCIPLES

1.1 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive and robust computational procedures modeled on the mechanics of natural genetic systems. GAs act as a biological metaphor and try to emulate some of the processes observed in natural evolution. They are viewed as randomized, yet structured, search and optimization techniques. They express their ability by efficiently exploiting the historical information to speculate on new offspring with expected improved performance. Specifically, they are intended to mimic some of the principles observed in natural evolution in the following ways:

1. Evolution is the process that operates on encoding of biological entities known as *genotypes*, rather than on the living beings themselves known as *phenotypes*. Similarly, GAs operate on encoding of possible solutions such as chromosomes of the problems, by manipulating strings of characters of an alphabet.
2. Natural selection is the link between a chromosome and its performance measured on its decoded version. Nature obeys the principle of Darwinian "survival of the fittest"; the chromosomes with high fitness values will, on the average, reproduce more often than those with low fitness values. In GAs, selection of chromosomes also tries to mimic this procedure. Highly fitted chromosomes will reproduce more often at the cost of lower fitted ones.
3. Nature acts as the environment and the biological entities are modified to adapt in their environment. Whatever nature knows about the evolution of good chromosomes is contained in the set of chromosomes possessed by the current individuals and in the chromosome decoding procedure. Likewise, GAs also operate on chromosomes blindly. They use only the objective function information which acts as the environment. Based on this information, each chromosome is evaluated and during the selection process more importance is given on choosing chromosomes having high fitness values.

1.2 Principles of Genetic Algorithms

The working principle of GAs is very different from that of most classical optimization techniques [6, 10]. In brief, the GA-based evolution starts from a set of individuals or a set of solutions for the function to be optimized and proceeds from generation to generation through genetic operations. Replacement of old population with a new population is known as generation. Let us briefly describe the steps to be followed to obtain a new population from an old population:

- **Population -** To solve an optimization problem, GAs start with the chromosome or structural representation of a parameter set $[x_1,x_2,...,x_p]$. The parameter set is to be coded as a finite length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's, and each chromosome refers to a coded possible solution. A set of such chromosomes in a generation is called a population.
- **Encoding/decoding procedure -** It is the mechanism to convert the parameter values of a possible solution into binary strings resulting into a chromosome representation. If the solution of a problem depends on p parameters and if we want encode each parameter with a binary string of length q, then the length of each chromosome will be $p \times q$. Decoding is just the reverse of encoding i.e., converting a binary string into a parameter set.
- **Fitness evaluation -** Assigning fitness to a solution depends upon the problem and it is done in such a way that highly fitted strings have high fitness values. Since, it is the only index to select a chromosome to reproduce for the next generation.
- **Reproduction/selection -** The primary objective of the selection operator is to make duplicates of good solutions into a mating pool by eliminating bad solutions in a population, while keeping the population size constant. The number of copies reproduced for the next generation by an individual is expected to be directly proportional to its fitness value, thereby mimicking the natural selection procedure to some extent. Most frequently used methods for selection are *tournament selection, roulette wheel selection and stochastic remainder roulette-wheel selection.*
- **Crossover -** Crossover is similar to the sexual reproduction where a child genotype is produced by mixing the genes of the parent genotypes. Similarly, in GAs, It takes two chromosomes from the mating pool at random and some portion of the chromosomes are exchanged to create new chromosomes with a probability of p_c.
- **Mutation -** The purpose of mutation is to maintain genetic diversity into the population. Likewise, in GAs the mutation operator is also expected to do same. A random position of a random string is selected and negates the bit value (i.e., a bit value of a 1 to a 0, and vice versa) with a probability of p_m. The parameter value of p_m plays a vital role in GAs.

1.3 Multi-Objective Genetic Algorithms

When an optimization problem modeling a physical system involves only one objective function, the task of finding the optimal solution is called single-objective optimization. Whereas, modeling a physical system involves more than one objective function, the task of finding one or more optimum solutions is known as multi-objective optimization. Most real-world search and optimization problems deal with simultaneous optimization of multiple objectives. In solving an multi-objective optimization problem (MOOP), there exist a number of fundamental differences between the working principles of single and multi-objective optimization algorithms [1, 6, 12], such as search and decision making. The first aspect refers to the optimization process in which a set of feasible solutions are sampled for Pareto-optimal solutions. The second aspect addresses the problem of selecting a suitable compromise solution from the Pareto-optimal set of solutions may produce trade-offs among different objectives. So, a human decision maker is necessary to make the trade-offs among conflicting objectives. Let us illustrate this trade-offs between two conflicting objectives with an example buying an automobile car [6]. Figure 1 shows two extreme hypothetical cases, i.e., one costing about ten thousand dollars (solution A) and another costing about a hundred thousand dollars (solution D) and other solutions represents with varying costs in between these two. If cost is the only objective of this decision-making process, the optimal choice is solution A. But, in most cases, barring some exceptions, this is not the only objective and it is expected that an inexpensive car is likely to be less comfortable. To rich buyers for whom comfort is the only objective of this decision-making process, the optimal choice is solution D. Between these two extreme solutions, there exist many other solutions (B, and C), where a trade-off between cost and comfort exists. Thus, between any two such solutions, one is better in terms of one objective, but this betterment comes only from a sacrifice of other objective.

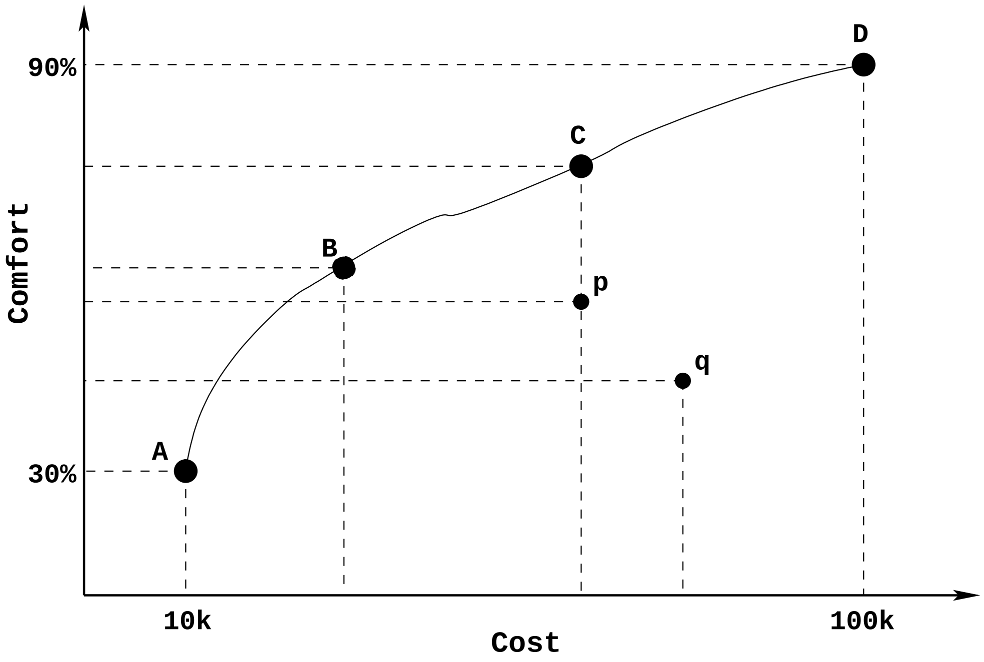

Fig. 1. Illustrates the concept of multi-objective optimization.

The concept of optimality in multi-objective optimization deals with a set of solutions instead of one solution. Based on this aspect, we first define the concept of domination for a solution to become dominated with respect to another solution and then discuss the concept of Pareto-optimality for a set of solutions to become a Pareto-optimal set.

1.4 Multi-Modal Optimization

As the name suggests, multi-modal functions will have multiple optimum solutions, of which many are local optimal solutions. However, one of the difficulties with the classical optimization methods is that the methods works with a single solution and finally allows an algorithm to converge to only one optima. Hence, they fail to find multiple optimal solutions. Whereas the objective in multi-modal function optimization is to identify multiple optimal solutions having either equal or unequal objective function values. Since EAs work with a population solutions instead of one solution, some amendments in the basic EA [5–7], enable to find multiple optimal solutions simultaneously in one single simulation run. To date, there exist a number of studies dealing with multi-modality in single objective optimization and have been successfully employed to many practical problems. But, there exist not even a single study dealing with the concept of multi-modality in MOEAs. In this study, we encounter a problem where there exist multi-modal solutions. Intuitively, the concept of multi-modality in multi-objective optimization can be thought of as *multiple distinct solutions per an objective vector* and this scenario is illustrated in Figure 2. Here, different distinct solutions (say $x^{'}$, $x^{''}$, and $x^{'''}$) in decision variable space are mapped to only one objective vector(f) in objective space.

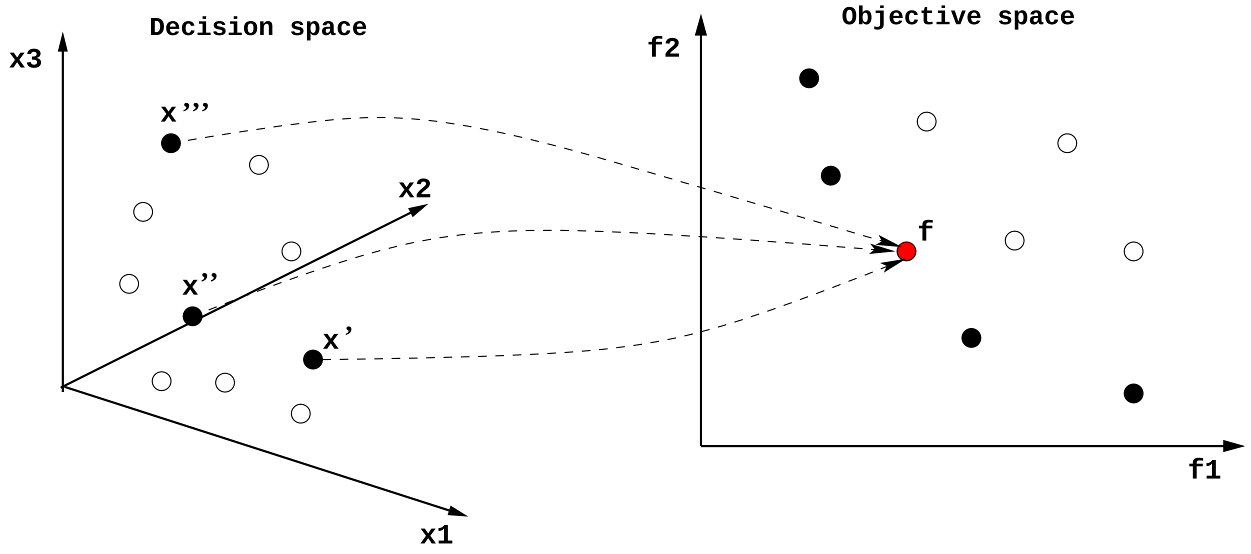

Fig. 2. Representation of the multiple distinct solutions in decision space and the corresponding objective vector in objective space.

1.5 Concept of Domination

Most multi-objective optimization algorithms are intended to use the concept of domination [6] to identify non-dominated set (i.e., Pareto-optimal set) of solutions for the given set of solutions. For this purpose, two solutions are compared on the basis of whether one dominates the other or not. The conditions for a solution to become non-dominated with respect to the other solutions are given below.

Definition 1 *If there are 'M' objective functions, a solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$, if both the conditions (1) and (2) are true:*

1. *The solution $x^{(1)}$ is no worse than $x^{(2)}$ in all objectives, or $f_j(x^{(1)}) \ntriangleright f_j(x^{(2)})$ for all $j = 1, 2, .., M$.*
2. *The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_{\bar{j}}(x^{(1)}) \triangleleft f_{\bar{j}}(x^{(2)})$ for at least one $\bar{j} \in \{1, 2, .., M\}$.*

If any of the above condition is violated, the solution $x^{(1)}$ does not dominate the solution $x^{(2)}$. If the solution $x^{(1)}$ dominates the solution $x^{(2)}$ (sometimes referred as, $x^{(2)}$ is dominated by $x^{(1)}$ or $x^{(1)}$ is non-dominated by $x^{(2)}$), this condition is mathematically represented as $x^{(1)} \preceq x^{(2)}$.

Let us describe these conditions with the above example, In Figure 1, if solution A is compared with solution B, condition 1 is violated (since, A is better in first objective but worst in second objective compared to B), so, A and B are non-dominated. If solution p is compared with solution q, conditions 1 & 2 are satisfied (since, p is better than q in both the objectives), thus, q is dominated by p. In case of solution C and p, C is better than p in second objective and no worse than p in first objective (perhaps, they are equal), thus, solution p is dominated by solution C.

1.6 Concept of Pareto-Optimality

A Pareto-front is the locus formed by a set of solutions which are equally good as compared to any other solutions of the set i.e., any solution on the front cannot be selected when compared with other, without being biased towards other. If the set of solutions lying on one particular front are non-dominated by the solutions lying on the other fronts is known as the Pareto-optimal front and the set of solutions lying on this front are known as Pareto-optimal solutions. In Figure 1, solutions A, B, C, and D forms a Pareto-optimal set. In the following subsections, we describe different operators used in NSGA-II [9].

1.7 Crowded Tournament Selection Operator

Diversity preservation along the front is expected to be the hub of most multi-objective optimization algorithms and it is expected to be achieved by using a crowded comparison operator [6]. This operator works on the basis of the crowding distance (d). The crowding distance is measured as the distance of the biggest cuboid containing the two neighboring solutions of the same Pareto-optimal front in the objective space. This parameter gives an estimation of the density of solutions surrounding a particular solution 'i' in the population of same rank (Figure 3). The solutions at the ends of a front are assigned a large value of crowding distance. This is done to keep the extremities of the front alive. Next, we define two conditions for a solution 'i' to become the winner against solution 'j' in the population [9].

Definition 2 *In a minimization type multi-objective problem, a solution 'i' wins a tournament with another solution 'j' if any of the following conditions are true:*

1. *If solution 'i' has a better rank, i.e. $r_i < r_j$.*
2. *If they have the same rank but solution i has a better crowding distance than solution j, i.e. $r_i = r_j$ and $d_i > d_j$.*

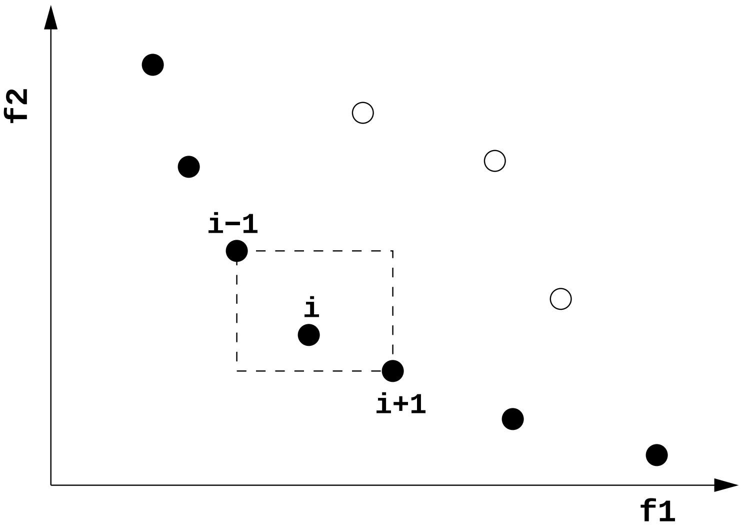

Fig. 3. Crowding distance calculation (taken from [6]).

Here, the first condition ensures that the chosen solution (winner) lies on a better non-dominated front. The second condition ensures the diversity in a particular front. If the solutions are in the same front, then the solution in a lesser crowded area (with a larger crowding distance d_i) is selected.

1.8 Elitism

Although the crossover and mutation are the core search operators of GA, there exists a probability of destruction of the good solutions developed so far due to these operators. Using elitism, it is possible to pass the best solutions of a particular generation to the next generation unaffected [6]. This helps the evolutionary algorithm to converge fast. To implement elitism in EAs, the old and the new population (after selection, crossover, and mutation) are merged together. The next generation is formed from this combined population by selecting the best individuals. Then as many individuals are selected in order to make the size of the new population is same as the old one. This ensures the preservation of good individuals of old and new population.

1.9 Modified Non-Dominated Sorting GA

Among the different multi-objective optimization algorithms, it has been observed that an elitist non-dominated sorting genetic algorithm (NSGA-II) has good features like, an explicit diversity preserving mechanism i.e., crowded tournament selection operator, and elite expressing operator. It also emphasizes the non-dominated solutions [5, 9]. The working principle of a multi-objective GA (NSGA-II) is described and used for reliable gene subset identification task.

In NSGA-II, the offspring population Q_t (of size N) is first created by using the parent population P_t (of size N) and the usual genetic operators such as crowded tournament selection, single-point crossover and bit-wise mutation operators (described in section 1.1). Thereafter, the two populations are combined together to form R_t of size $2N$. Then, a non-dominated sorting procedure is carried using the concept of domination to classify the entire population R_t. Once the non-dominated sorting is over, the new parent population P_{t+1} is created by choosing solutions of different non-dominated fronts, one at a time. The filling starts with strings from the best non-dominated front and continues with strings of

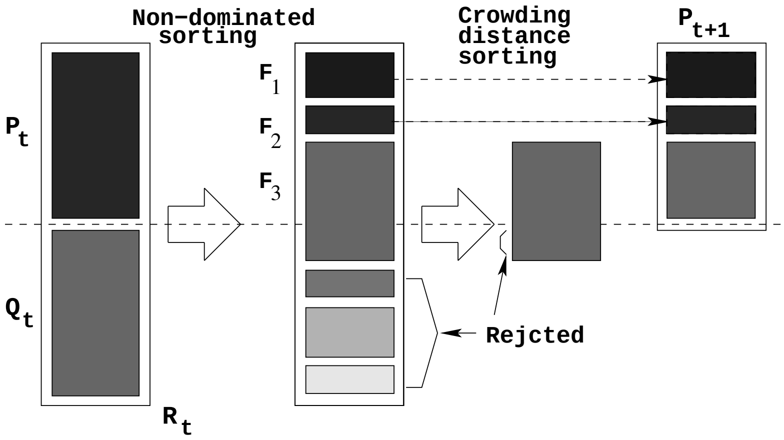

Fig. 4. Schematic of the NSGA-II procedure.

second non-dominated front, followed by the third non-dominated front, and so on. Since the overall population size of R_t is $2N$, not all fronts can be accommodated in the new parent population. The fronts which could not be accommodated at all are simply discarded. However, while the last allowed front is being considered, there may exist more strings in it than the remaining population slots in the new population. This scenario is illustrated in Figure 4. Instead of arbitrarily choosing some strings from this last front, the strings which will make the *diversity* of the selected strings the maximum are chosen. For each string, we calculate a computationally simple *crowding distance* measuring the Euclidean distance among the neighboring strings in the objective space. Thereafter, those strings having largest crowding distance values are chosen to fill up the new parent population. This procedure is continued for a maximum of user-defined T iterations. It has been proved in evolutionary computation that due to the emphasis of the non-dominated solutions, preserving diversity among population members and an elitist approach, NSGA-II has been successful in converging quickly close to the true Pareto-optimal front with a well-diversed set of solutions in the objective space.

1.10 Multi-Modal MOEAs for Multiple Solutions

However, the next and most important observation we made in the gene subset identification task is that the identification task with multiple objectives may involve multi-modal solutions, meaning that for a point in the objective space, there may exist more than one solutions in the decision variable space (on the Hamming string space). When this happens, it becomes important and useful for a biologist to know which all gene combinations may provide an identical classification. Moreover, it is expected to significantly aid in understanding and constructing metabolic pathways. For the first attempt in multi-objective EAs, in the current study, we have proposed a modified NSGA-II procedure for identifying such multi-modal solutions. Here, we define two multi-modal solutions in the context of multi-objective optimization as follows:

Definition 3 (Multi-modal solutions) *If for two different solutions* $\mathbf{x}^{(1)}$ *and* $\mathbf{x}^{(2)}$ *satisfying* $\mathbf{x}^{(1)} \neq \mathbf{x}^{(2)}$*, all objective values are the same, or* $f_i(\mathbf{x}^{(1)}) = f_i(\mathbf{x}^{(2)})$ *for all* $i = 1, 2 \ldots, M$*, then solutions* $\mathbf{x}^{(1)}$ *and* $\mathbf{x}^{(2)}$ *are multi-modal solutions.*

In such problems, more than one solutions in the decision variable space maps to one point in the objective space. We are not aware of any MOEA which has been attempted for this task or for any other similar problems. Fortunately, the gene subset identification problem is an ideal problem in which multiple gene subsets (that is, different combinations of genes) can produce an identical outcome in terms of their mismatches in the training and test samples.

1.11 Proposed NSGA-II for Multi-Modal Solutions

Recall that in an NSGA-II iteration, the parent P_t and offspring Q_t populations are combined to form an intermediate population R_t of size $2N$. Thereafter, R_t is sorted according to a decreasing order of non-domination level $(\mathcal{F}_1, \mathcal{F}_2, \ldots)$. In the original NSGA-II, solutions from each non-dominated level (starting from the best set) are accepted till a complete set cannot be included without increasing the designated population size. A *crowding operator* was then used to choose the remaining population members. In the proposed approach, we follow an identical procedure till the non-dominated sorting is done. Thereafter, we use a slightly different procedure. But before we describe the procedure, let us present two definitions:

Definition 4 (Duplicate solutions) *Two solutions* $\mathbf{x}^{(1)}$ *and* $\mathbf{x}^{(2)}$ *are duplicates to each other if* $\mathbf{x}^{(1)} = \mathbf{x}^{(2)}$.

This definition follows the principle - duplicate solutions will have identical objective values.

Definition 5 (Distinct objective solutions) *Two solutions* $\mathbf{x}^{(1)}$ *and* $\mathbf{x}^{(2)}$ *are distinct objective solutions if* $f_i(\mathbf{x}^{(1)}) \neq f_i(\mathbf{x}^{(2)})$ *for at least one* i *(where* $i \in 1, 2 \ldots, M$*).*

First, we delete the duplicate solutions from each non-domination set in R_t. Thereafter, each set is accepted as usual till the last front $\mathcal{F}_l$ which can be accommodated. Let us say that solutions remaining to be filled before this last front is considered is N' and the number of non-duplicate solutions in the last front is N_l $(> N')$. We also compute the number of distinct objective solutions in the set $\mathcal{F}_l$ and let us say it is n_l (obviously, $n_l \leq N_l$). This procedure is illustrated in Figure 5. If $n_l \geq N'$ (the top case shown in the figure), we use the usual crowding distance procedure to choose N' most dispersed and distinct solutions from n_l solutions. In this case, even if there exist multi-modal solutions to any n_l solutions, they are ignored due to lack of space in the population. The major modification to NSGA-II is made when $n_l < N'$ (bottom case in the figure). This means that although there are fewer distinct solutions than the population slots, the distinct solutions are multi-modal. However, the total number of multi-modal solutions of all distinct solutions (N_l) is more than the remaining population slots. Thus, we need to make a decision of choosing a few solutions. The purpose here is to have at least one copy of

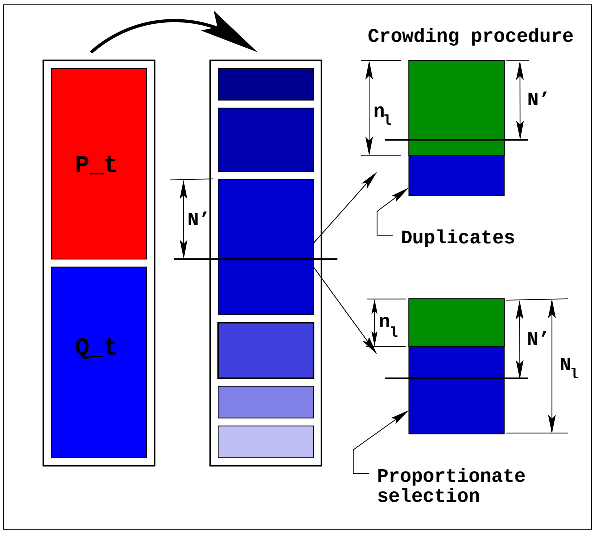

Fig. 5. Schematic of the multi-modal NSGA-II procedure is shown.

each distinct objective solution and as many multi-modal copies of them so as to fill up the population. Here, we choose a strategy in which every distinct objective solution is allowed to have a proportionate number of multi-modal solutions as they appear in $\mathcal{F}_l$. To avoid loosing any distinct objective solutions, we first allocate one copy of each distinct objective solution, thereby allocating n_l copies. Thereafter, the proportionate rule is applied to the remaining solutions $(N_l - n_l)$ to find the accepted number of solutions for the i-th distinct objective solution as follows:

$$\alpha_i = \frac{N' - n_l}{N_l - n_l}(m_i - 1), \tag{1}$$

where m_i is the number of multi-modal solutions of the i-th distinct objective solution in $\mathcal{F}_l$, such that $\sum_{i=1}^{n_l} m_i = N_l$. It is true that $\sum_{i=1}^{n_l} \alpha_i = N' - n_l$. The final task is to choose $(\alpha_i + 1)$ multi-modal solutions from m_i copies for the i-th distinct objective solution. Although a sharing strategy [2] can be used to choose the maximally different multi-modal solutions, here we simply choose them randomly. Along with the duplicate-deletion strategy, the random acceptance of a specified number multi-modal solutions to each distinct objective solution ensures a good spread of solutions in both objective and decision variable space. In the rare occasions of having less than N non-duplicate solutions in R_t, new random solutions are used to fill up the population. For a problem having many multi-modal solutions, the latter case will occur often and the above systematic preservation of distinct objective solutions and then their multi-modal solutions will maintain a rich collection of multi-modal Pareto-optimal solutions.

2 EVOLUTIONARY GENE SELECTION PROCEDURE

2.1 Resulting Optimization Problem

In the current study, we formulate the gene subset identification task as a multi-objective optimization problem with three different conflicting objectives. As we have discussed earlier that, one of the objectives of the above identification task is to minimize the number of genes used in classification while maintaining acceptable classification accuracy. Although not obvious, when a too small gene subset is used, the classification procedure becomes erroneous. But, to achieve the best possible performance with a particular classifier on a particular domain, the gene subset identification task should consider how the classifier and the training data interact. So, minimization of class prediction mismatches in the training samples and the test samples are also important objectives. Hence, the gene subset identification task can be formulated as a multi-objective optimization problem with three different conflicting objectives. The multi-objective optimization problem formulation is as follows:

1. The first objective function f_1, in gene subset identification task is to minimize the number genes in a subset or to minimize the gene subset size for a classifier. This reflects the number of 1's in a $l-$bit binary string (where l is the number of genes in a data set after filtering).
2. The second objective function f_2, is to minimize the number of class prediction mismatches in the training samples, these are calculated using the LOOCV (section **??** procedure).
3. The third objective function f_3, in identification task is to minimize the number of class prediction mismatches in the test samples, these are calculated using the classifier constructed based on all samples in the training set.

In the current study, we have used an evolutionary based technique to identify most discriminative gene subsets, a weighted voting approach is employed to predict the class of a sample based on such informative gene subsets and a set of samples with known class labels, and the LOOCV procedure is used to determine the number of mismatches in the training samples. Thereafter, the classifier is constructed using such informative gene subsets and all the training samples. Then, the performance of the classifier is estimated using the remaining samples in a test set. A multi-objective GA (NSGA-II) is considered for handling above three different conflicting objectives.

2.2 Solution Procedure Using Evolutionary Algorithms

It is a common practice in most of the machine learning algorithms for feature subset selection problems [3, 4] that, each feature in a data set is represented as a *binary bit* (i.e., a bit value of a 1 or a 0). Thus, the gene subset identification task can be thought as same as the feature subset selection in learning algorithms and each individual or solution for an EA is represented as a l-bit *binary string*, where l is the number of filtered genes in a data set and each bit is associated with a gene in that data set. For a particular individual, the positions marked with a 1 are included in the classification process, and the positions marked with a 0 are not allowed to participate in the classification process. For example, the Figure 6 depicts a 10-bit binary string (representing a total of 10 genes in a data set), first, third, and sixth genes are considered for classification purpose.

The random initialization of the population is a common practice in most of the evolutionary algorithms and it has also been observed that the random initialization of the binary strings with a fixed length will have on an average equal number of 1's in all the individuals (approximately half of the length of the strings). But, one of the main difficulties in solving the gene subset identification problem

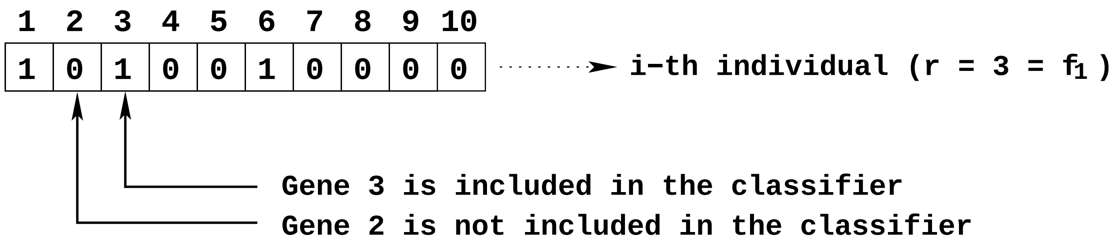

Fig. 6. A 10-bit binary string, representing an i^{th} individual of the population.

is the availability of large number of genes in a data set as compared to the number of tissue samples and which in turn yields the exorbitantly large search space of solutions. In the past studies, it has been observed that the number of genes responsible for causing a particular disease are very less as compared to the total number of genes present in the data set. For example, in the Leukemia data set, there are total 7,129 gene expression measurements in 72 samples are available but Golub et al. [11] have used only 50 genes to achieve a reasonable classification accuracy. Although an EA has the ability to search all possible subsets theoretically, it would take long time to obtain the global optimum solution. Hence, the random initialization of the population for solving such complex problems is meaningless moreover, which makes an EA to loose diversity in the population. Since the first objective in gene subset identification problem is to minimize the gene subset size. So, we can introduce some kind of problem information to make the EAs to be more effective for the gene subset identification problem and in the past [8], it was suggested that the EAs can be used effectively with less computational complexity by incorporating problem information into the EAs.

Thus, the gene subset identification problem can be solved efficiently by initializing the population with a semi-random initialization method. In this approach, we first generate a random number(r_i) between 1 and $0.1 \times l$ for i^{th} individual in the population, which ensure that there is at least one gene in a subset and at most 10% of the total number of genes in the data set. Thereafter, generate r_i random numbers between 1 to l in order to place 1's in a binary string and assign remaining positions in a binary string with 0's. So, the positions marked with 1's represents the gene number and allows to participate in the classification process. This procedure is repeated for all the individuals in a population (i.e., $i = 1, 2, ..., N$). This biasing against 1 in a string allows an EA to start with good population members and which also preserves diversity of the initial population in each objective.

References

1. Petrovski A. and McCall J. Multi-objective optimization of cancer chemotherapy using evolutionary algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pages 531–545, 2001.
2. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 41–49, 1987.
3. Liu J. and Iba H. Selecting informative genes using a multiobjective evolutionary algorithm. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2002)*, pages 297–302, 2002.
4. Yang J. and Honavar V. Feature subset selection using a genetic algorithm. In *In Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 380–386, 1997.

5. Deb K. Genetic Algorithms in Multi-modal Function Optimization. Master's thesis, Tuscaloosa, AL: University of Alabama, 1989.
6. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.
7. Deb K., Horn J., and Goldberg D. E. Multi-modal deceptive functions. *Complex Systems*, 7(2):131–153, 1993.
8. Deb K., Reddy A. R., and Singh G. Optimal Scheduling of Casting Sequence Using Genetic Algorithms. Technical Report (KanGAL Report No. 2002002, Kanpur, India: Department of Mechanical Engineering, Indian Institute of Technology Kanpur, March, 2002. http://www.iitk.ac.in/kangal/pub.htm.
9. Deb K., Agrawal S., Pratap A., and Meyarivan T. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
10. Pal S. K. and Wang P. P. *Genetic Algorithms for Pattern Recognition*. Boca Raton: CRC Press, 1996.
11. Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H., Loh M. L., Downing J. R., Caligiuri M. A., Bloomfield C. D., and Lander E. S. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
12. Li X., Jiang T., and Evans D. J. Medical image reconstruction using a multiobjective evolutionary algorithm. *International Journal of Computer Mathematics*, 75(3):301–314, 2000.