
Article:

Kumar, R. and Rockett, P. (2002) Improved sampling of the pareto-front in multiobjective genetic optimizations by steady-state evolution: a Pareto converging genetic algorithm. Evolutionary Computation, 10 (3). pp. 283-314. ISSN 1063-6560

https://doi.org/10.1162/106365602760234117

Improved Sampling of the Pareto-Front in Multiobjective Genetic Optimizations by Steady-State Evolution: A Pareto Converging Genetic Algorithm

Rajeev Kumar rkumar@cse.iitkgp.ernet.in
Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721 302, India

Peter Rockett p.rockett@sheffield.ac.uk
Department of Electronic and Electrical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

Abstract

Previous work on multiobjective genetic algorithms has been focused on preventing genetic drift and the issue of convergence has been given little attention. In this paper, we present a simple steady-state strategy, *Pareto Converging Genetic Algorithm* (PCGA), which naturally samples the solution space and ensures population advancement towards the Pareto-front. PCGA eliminates the need for sharing/niching and thus minimizes heuristically chosen parameters and procedures. A systematic approach based on histograms of rank is introduced for assessing convergence to the Pareto-front, which, by definition, is unknown in most real search problems. We argue that there is always a certain inheritance of genetic material belonging to a population, and there is unlikely to be any significant gain beyond some point; a stopping criterion where terminating the computation is suggested. For further encouraging diversity and competition, a nonmigrating island model may optionally be used; this approach is particularly suited to many difficult (real-world) problems, which have a tendency to get stuck at (unknown) local minima. Results on three benchmark problems are presented and compared with those of earlier approaches. PCGA is found to produce diverse sampling of the Pareto-front without niching and with significantly less computational effort.

Keywords

Genetic algorithms, multiobjective optimization, steady-state, Pareto converging, rank histogram, island model.

1 Introduction

Genetic algorithms (GAs) are increasingly used in problem domains that can be (re-) defined in terms of search procedures and subsequent optimization of objective function(s). Searching for (near-) optimal solutions within some real-world space generally requires optimization of multiple and often conflicting objectives that rarely yield a unique solution. In a typical multiobjective optimization problem, there exists a family of equivalent solutions that are superior to the rest of the solutions and are considered equal from the perspective of simultaneous optimization of multiple (and possibly competing) objective functions. Such solutions are called noninferior, nondominated, or Pareto-optimal solutions (Hwang and Masud, 1980; Pareto, 1896; Steuer, 1986). In this paper, we use all three terms interchangeably.

In simple treatments, multiple objectives are combined in an *ad hoc* manner to yield a scalar objective, typically by linear combination of the vector elements, and constraints are incorporated with associated thresholds and penalty functions. Though the process results in a simple optimization algorithm, the solution obtained is very sensitive to small changes in weight vector and penalty function.

A wide-ranging review of evolutionary approaches to multiobjective optimization has been given by Fonseca and Fleming (1995a) who discuss their similarities, differences, sensitivity to objective scaling and possible concavities in the trade-off surfaces, and cost landscapes. Since then, many review articles appeared (for a current list of references, see Coello (2001)). Moreover, two recent publications are worth mentioning. First is an overview of evolutionary algorithms by Whitley (2001), which covers practical issues and common pitfalls. The second is a detailed coverage of multiobjective optimization by Deb (2001). A brief review of multiobjective genetic optimization highlighting the distinctive features of our approach is given in the next section of this paper.

A study identifying typical features of multiobjective test problems such as multimodality and deception has also been carried out by Deb (1999). Solutions of many multiobjective (mostly dual objective) analytical functions have been reported by numerous researchers, although in most such problem domains the solution space was known *a priori*. All of these authors used sharing/mating restrictions in one form or another to achieve genetic diversity on the known Pareto-fronts.

Simultaneously, many real-world problems including NP-complete problems have been attempted through multiobjective genetic algorithms. For example: a gas turbine engine controller (Chipperfield and Fleming, 1996), 3-D system packaging (Larcombe, 1996), pressurized water reactor reload design (Parks and Miller, 1998), microprocessor chip design (Stanley and Mudge, 1995), and 0/1 knapsack problem (Zitzler and Thiele, 1999). Such real-world problems can be classified into two groups: the first whose solution is known a priori or can be approximated by some means, and second, those for which the solution space is unknown. For the first class of problems, tolerance limits or achievable percentages of defined goals can give some indication of solutions moving towards goal-convergence, and thus solutions obtained by genetic optimization could be compared. Many metrics are available in the literature (Zitzler and Thiele, 1999; Tan et al., 2001) that measure the diversity of the obtained Pareto-front and the distance between the obtained front and the desired one. Thus, the efficacy of the genetic implementation may be measured and the results obtained by genetic optimization verified. However, for real-world problems where we have neither prior knowledge nor any approximation of the solution space, the issue of convergence is no simple matter.

In addressing an unsolved real-world problem in hydro-systems with two conflicting objective attributes, Horn et al. (1994) observed that they were in no position to appreciate whether the obtained solution set was really a true optimal solution, since they could only visualize the population diversity across a simple two-dimensional Pareto-front of objectives. Similar observations were made by many other researchers who tried to solve unknown problems with multiobjective genetic algorithms, e.g., Kumar and Rockett (1998). Clearly in previous work there has been insufficient consideration given to the convergence of the solution space; not only do we seek a set of uniformly distributed solutions but also the optimal set of solutions. Ignorance of convergence of a Pareto-based multiobjective optimization is one of the issues we are attempting to address in the present work.

In a single-criterion optimization, the notion of convergence (as distinct from optimality) scarcely needs explanation; we simply seek the best value of the objective, cost, or fitness. Nonetheless, considerable work has been done on understanding the convergence rates of various selection schemes for single objective optimization (Bäck, 1995; Bäck and Schwefel, 1993; Eshelman and Schaffer, 1991; Goldberg and Deb, 1991; Miller and Goldberg, 1995; Neri and Saitta, 1995; Thierens and Goldberg, 1994). Work on understanding the statistics of the dynamics has been done by Shapiro et al. (1994). In an elitist genetic algorithm, Bhandari et al. (1996) have shown that an optimal solution can be obtained as the number of generations goes to infinity. Aytug and Koehler (1996, 2000) established tighter theoretical bounds on the number of GA iterations at which the search should be terminated. However, Aytug and Koehler's theoretical bounds are only applicable to single-objective problems and do not address how to maintain diversity in a finite population using optimum and tractable computing resources. Rudolph (1998) and Rudolph and Agapie (2000) extended the single-objective theoretical framework to multiobjective optimization and showed that a true Pareto-front can be obtained in a finite number of functions evaluations in a finite but continuous search space. Rudolph's bound ensures convergence but ignores the equally important issue of maintaining diversity. Here we recall that the *Vector Evaluated Genetic Algorithm* (VEGA) (Schaffer, 1985) does converge to some optimal solutions but does not achieve diversity.

In most real search problems, the location of the actual Pareto-front is, by definition, unknown, and the identification of a "best value" of some criterion does not necessarily mean global convergence. In problem domains of low objective dimensionality, the Pareto-front can be examined for genetic diversity but not for convergence; high-dimensional objective spaces cannot generally be visualized for either diversity or convergence. (Some performance metrics, e.g., volume of space covered, distribution (Tan et al., 2001), can provide information about diversity alone.) Knowledge of the propagation of the solution front through successive generations of the population, however, can serve as a cue for convergence. With this background, we reformulate the notion of convergence in a multiobjective optimization environment in terms of:

1. Preventing a premature convergence and ensuring a uniform/diverse sampling of the Pareto-front,
2. Monitoring of the evolved population: how much fitter/superior is a given generation than the previous one, and the rate of propagation towards the Pareto-front,
3. Setting some stopping criterion; where to terminate the evolution of further generations.

Although the sharing/mating restrictions employed by various authors partly solve the problem of premature convergence, they do not necessarily guarantee overall convergence. Viewed as a statistical sampling problem over the objective space, just because a given solution point dominates all others in the (finite) sample does not imply that it is drawn from the Pareto-optimal set — the given nondominated point could itself be dominated by another yet undiscovered solution that, in turn, need not necessarily be drawn from the Pareto-optimal set. In the past, a simple upper bound on the number of generations/iterations has frequently been used as a stopping point, while others have employed the production of some percentage of nondominated individuals in the total population as a stopping criterion. The first of these is unsatisfactory because a large amount of processor time could be wasted producing further generations for an optimization that has already converged; alternatively, there is no way of knowing that a particularly stubborn problem is still far from convergence. The second option is ill-conceived because solutions are nondominated relative to the population sample not the universe of optimal solutions; just because a solution dominates all others in the current population does not imply that it lies on the desired Pareto-front. In the course of a multiobjective optimization, it is completely normal for solutions that are nondominated at some stage in the computation to become dominated by a superior solution at a later stage.

In this paper, we present a straightforward way of examining convergence of a rank-based multiobjective GA that makes principled comparison of the population states during the computation. In doing so, we attempt to minimize the dependency on heuristically chosen parameters and procedures, both for implementation and measuring the performance. We implement a simple, steady-state evolutionary algorithm in a multiobjective environment that produces diverse sampling of the Pareto-front without any diversity-preserving mechanisms, e.g., niching, sharing/mating restrictions and/or crowding in a parameter or objective space.

The remainder of this paper is organized as follows. In Section 2, we briefly review multiobjective genetic algorithms and build the problem definition. In Section 3, we present our simple, steady-state strategy for multiobjective genetic algorithms that naturally prevents genetic drift and ensures population advancement towards the Pareto-front. We eliminate the need for choosing a domain for sharing, i.e., phenotypic/genotypic and obviously sharing parameter(s). We introduce an optional isolated island approach for encouraging diversity and competition across islands. We test this genetic strategy, which we call the *Pareto Converging Genetic Algorithm* (PCGA), on three benchmark analytical functions as a proof of concept for solving benchmark problems and present the results in Section 4. Along with discussion of the results, a systematic and statistical (albeit heuristic) approach for examining convergence to the Pareto-front is discussed in Section 5. We draw conclusions in Section 6.

2 Related Work

Work on multiobjective genetic optimization dates back to 1985 with the introduction of VEGA, but seminal results on Pareto rank-based selection began with the work of Fonseca and Fleming (1993).

Mathematically, a general multiobjective optimization problem containing a number of objectives to be maximized/minimized along with (optional) constraints for satisfaction of achievable goal vectors can be written as:

$$\begin{gathered} Minimize/\ Maximize\ Objective\ f_m\left(\mathbf{X}\right), \quad m = 1, 2, ..., M \\ subject\ to\ Constraint\ g_k\left(\mathbf{X}\right) \le c_k, \quad k = 1, 2, ..., K \end{gathered}$$

$$\begin{gathered} where\ \mathbf{X} = \{x_n : n = 1, 2, ..., N\}\ is\ an\ N - tuple\ vector\ of\ variables \\ and\ \mathbf{F} = \{f_m : m = 1, 2, ..., M\}\ is\ an\ M - tuple\ vector\ of\ objectives \end{gathered}$$

In rest of this section, we critically examine some of the representative algorithms and their implementations. Then we analyze their distinctive features and build the problem definition of the work presented in this paper.

2.1 Previous Approaches to Multiobjective Genetic Optimization

2.1.1 Non-Pareto Approaches

The first practical genetic algorithm for finding multiple solutions to a multiobjective problem was Schaffer's VEGA. In VEGA, the converged population is biased towards the individual optimum objective regions — a phenomenon called speciation. A similar problem was addressed by Fourman (1985) who assigned different priorities to multiple objectives in lexicographic ordering (Ben-Tal, 1980). Initially, the objective priorities were assigned by the user thus biasing the solutions towards the user's subjective preferences; in another version of the algorithm, priorities were randomly picked. Kursawe (1991) also implemented a multiobjective selection process with similarities to both Schaffer and Fourman in an attempt to reduce the bias against extreme regions of an individual objective, but the deletion of individuals by random objective selection creates a nonstationary population; individuals in the extremes of the objective spaces are liable to be eliminated if an objective at which they happen to perform poorly is selected as the basis for deletion. The population under such a situation adapts to changes rather than converging. Hajela and Lin's (1992) algorithm used a variable weighted sum method for fitness assignment; weights were genotype encoded and contributed to the diversity. Laughlin and Ranjithan (1997) proposed a neighborhood constraint GA that uses a population approach coupled with the classical ε-constraint method, where all but one of the objectives are converted into constraints one-by-one, and only one objective is optimized at a time. This algorithm is very sensitive to the many parameter settings.

2.1.2 Pareto Rank-Based Methods

Goldberg (1989) introduced the notion of Pareto-optimality for assigning equal probabilities of regeneration to all the nondominated individuals in a population and stated the condition for Pareto-optimality as : In a minimization problem, an individual objective vector $\mathbf{F}_i$ is partially less than another individual objective vector $\mathbf{F}_j$ (symbolically $\mathbf{F}_i \prec \mathbf{F}_j$) *iff*

$$(\mathbf{F}_i \prec \mathbf{F}_j) \;\triangleq\; (\forall_m)\,(f_{mi} \leq f_{mj}) \wedge (\exists_m)\,(f_{mi} < f_{mj})$$

Then $\mathbf{F}_i$ is said to dominate $\mathbf{F}_j$. If an individual is not dominated by any other in the population, it is said to be nondominated.

Goldberg suggested the use of Pareto-ranking and selection to move a population towards the Pareto-front and also proposed the use of additional safeguards in the form of niche formation and mating restrictions — originally used in multimodal optimization (Deb and Goldberg, 1989) to prevent convergence to a single point.

The major achievement of the Pareto rank-based research is that a multiobjective vector is reduced to a scalar fitness without combining the objectives in any way. Further, the use of fitness based on Pareto-ranking permits nondominated individuals to be sampled at the same rate, thus according the equal preference to all nondominated solutions in evolving the next generation. The mapping from ranks to fitness values however is an influential factor in selecting mates for reproduction.

Fonseca and Fleming (1993, 1998) implemented their Pareto-ranking version of the *Multi-Objective Genetic Algorithm* (MOGA) in a slightly different way to that of Goldberg (1989). In MOGA, the rank of an individual is equal to the number of individuals by which it is dominated, and all nondominated individuals are given the same rank. The fitness values are assigned by a linear, exponential, or some other functional interpolation of the ranks from the best to worst, and the selection procedure uses those fitness values in a *Stochastic Universal Sampling* (SUS) algorithm (Baker, 1987). MOGA also uses niche formation to distribute the population over the Pareto-optimal region. Srinivas and Deb (1994) implemented similar sorting and fitness assignment procedures, which they call the *Non-dominated Sorting Genetic Algorithm* (NSGA); the minor differences with MOGA are that (i) the Pareto-ranking is exactly that of Goldberg, and fitness values are dummy numbers in decreasing order of nondominated sorting, and (ii) instead of performing sharing in the objective domain (Fonseca and Fleming, 1993), they have used sharing on parameter values/decision variables (Deb and Goldberg, 1989). Ritzel et al. (1994) used tournament selection with Pareto-ranking, and the individual ranks decided the winner of tournaments.

Zitzler and Thiele (1999) proposed a *Strength Pareto Evolutionary Algorithm* (SPEA) that externally stores the nondominated solutions and performs clustering to reduce the size of the external storage; SPEA uses tournament selection for reproduction and fitness sharing. In this algorithm, niches are not defined in terms of distance but Pareto-dominance. Valenzela-Rendón and Uresti-Charre (1997) developed a nongenerational GA that uses Pareto-ranking for measuring the fitness and a sharing function for diversity.

2.1.3 Pareto but Non-Ranking Approach

Horn et al. (1994) introduced the *Niched-Pareto Genetic Algorithm* (NPGA) based on Pareto-domination tournaments and sharing instead of sorting and ranking of obtained solutions. In this method, a comparison set of a specific number of individuals is chosen at random from the population at the beginning of each selection process. Two individuals are randomly picked from the population and compared with the members of the comparison set for domination with respect to the objective functions. If one of them is nondominated and the other is dominated, the nondominated one is selected. In the case of a tie, the winner is decided by sharing-cum-niche count sampling — a metric that combines both the objective and decision variable domains. Horn et al. found the performance of NPGA to be very sensitive to the settings of several parameters. The behavior is most affected by the size of the comparison set — a well-known problem of tournament-based selection approaches (Goldberg et al., 1989). If a proper size is not chosen, true nondominated points may not be found; a small comparison set may result in a few nondominated points in the population, while premature convergence to a small portion of the Pareto-front may result with a large comparison set.

Knowles and Corne (2000) proposed a *Pareto Archived Evolutionary Strategy* (PAES) in two modes: (1+1)-ES and (μ, λ)-ES. (1+1)-PAES is basically a local search strategy based on a mutation operator and the Pareto-dominance. Nondominated solutions are kept in an archive. Diversity is achieved by a crowding scheme that divides the whole search space into n-dimensional hypercubes for n objectives, and fitness is assigned in accordance with a population count — the lower the count, the higher the fitness. (μ, λ)-PAES uses tournament selection, and the hypercube size is a critical/sensitive parameter to achieve diversity. A variation of (1+1)-PAES is a multiobjective micro GA proposed by Coello and Toscano (2001) that is similar to PAES but has two population sets.

Almost all the multiobjective genetic algorithms/implementations have ignored the issue of convergence. Another drawback of most of these algorithms/implementations is the need for parameterized sharing, mating restrictions, and/or some other diversity preserving operator. This is discussed further in Section 2.2.2.

2.2 A Critical Review of Basic Features of Multiobjective Genetic Optimization

In an attempt to closely analyze the similarities and differences between multiobjective GAs along with their strengths and sensitivities, we abstract the basic features of many algorithms. A multiobjective GA can be designed with the right mix of some or all of the following features.

2.2.1 Population Evolution Strategy and Selection

A classical GA is a generational GA that replaces the whole population at each generation. This contrasts with the steady-state GA, where one member of the population is replaced at a time. In terms of *evolutionary strategy* (ES), these two classes can be termed as (μ, λ)-ES and (μ + 1)-ES, respectively. (In (μ, λ)-ES, offspring replace the parents, but in (μ + λ)-ES, selection picks from both the parents and offspring to create the next generation.) There are different mechanisms for selection (SUS and tournament selection) and different ways to manipulate parents and offspring. De Jong (1975) and De Jong and Sarma (1993) introduced the concept of a generation gap G to distinguish between these two extremes; for generational GAs, $G = 1$, while for steady-state GAs, $G = 1/P$ where P is the population size.

In single objective optimization, GENITOR (Whitley, 1989) was the first steady-state algorithm and can be termed as a (μ + 1)-ES in terms of its selection mechanism. Another (μ +λ)-ES based algorithm is Eshelman's (1991) CHC algorithm, which is built on the concept that recombination should be the dominant search operator. It was shown that the CHC algorithm exhibits aggressive selection and elitism. Many studies in the past have shown that steady-state operation can generate higher selection pressure than generational GAs. In a recent study, Rogers and Prügel-Bennett (1999) showed that genetic drift is another factor that affects the convergence along with the more commonly understood selection pressure, and they showed that the rate of genetic drift in steady-state selection is twice that of generational selection. Rogers and Prügel-Bennett conjectured both analytically and empirically that a steady-state GA can reproduce the dynamics of a generational GA at half the computational cost. This is consistent with the experience of GENITOR's researchers that, in practice, steady-state algorithms are often better optimizers than generational GAs.

In this paper, we use a steady-state algorithm for multiobjective optimization. To the best of our knowledge, no such attempt has been made, however, Valenzela-Rendón and Uresti-Charre (1997) worked on a nongenerational GA for multiobjective optimization in which they optimize a single objective by aggregating all the multiple objectives using a weighted ranking scheme due to Bentley and Wakefield (1997).

2.2.2 Diversity-Achieving Functions and Operators

Almost all multiobjective optimization implementations explicitly use some diversity preserving mechanism in the form of fitness sharing, mating restriction, density count (crowding), or some preselection operator.

Apart from its heuristic nature, the selection of the domain in which to perform sharing (variable (genotype) or objective (phenotype)) is also debatable. For example, NSGA uses genotype sharing while MOGA uses phenotype space; both obtained the desired spread of solutions over the Pareto-front. Deb (1999) demonstrated the effectiveness of genotypic over phenotypic sharing while testing a single-objective function that was recast into a dual-objective function. We feel that such a claim is problem dependent. In the case of multimodal functions and the test problems of Deb (1999), genotype sharing may give superior results because a spread of solutions across variables was needed, but this cannot be generalized across all problem domains. Additionally, for effective sharing, one needs to know the niche count, σ_{share} and distance metric (over some norm), which are known for analytic functions but not generally for real problems. Sharing in objective space in practice means an implicit knowledge of the solutions in some limited sense, which is a paradox. Further, the computational complexity of implementing sharing increases with dimensionality. Thus fitness sharing is sensitive to many parameters including population size.

Mating restrictions are similar to fitness sharing where we need to know σ_{mate}. Results obtained by many researchers are not very encouraging, indeed for some researchers, mating restrictions are of no value. For example, Zitzler and Thiele (1999) observed no improvement compared to omitting mating restrictions; similarly, Shaw and Fleming (1996) found no gain. De Jong (1975) introduced a crowding model for multimodal optimization. Knowles and Corne (2000) incorporated density count in their PAES, but this scheme is very sensitive to the hypercube volume that controls the density count. For multimodal function optimization, Cavicchio (1970) introduced the preselection operator later modified by Mahfoud (1992).

In summary, any explicit diversity preserving method needs prior knowledge of many parameters, and the efficacy of such a mechanism depends on successful fine tuning of these parameters. It is the experience of almost all GA researchers that proper tuning of sharing parameters is necessary for effective performance. It is worth noting the "No Free Lunch" theorems (Wolpert and Macready, 1997) — that if an algorithm performs well on a certain class of problems, then it will exhibit degraded performance on the remaining problems.

In this work, we adopt another approach of excluding any explicit mechanism to preserve diversity and allowing a natural selection process to maintain diversity.

2.2.3 Elitism and Secondary Population

In earlier implementations of multiobjective optimization, e.g., VEGA, MOGA, and NSGA, the biggest challenge was how to maintain the older nondominated solutions in the current population. Algorithms such as MOGA and NSGA maintain diversity by sharing and require long runs with the attendant large computing resources. They do not include any elitism strategy, perhaps, in the absence of a convergence metric, due to a fear of premature convergence, which is the disadvantage of elitism in a single-objective GA. Parks and Miller (1998) investigated the efficacy of various elitist selection strategies in a multiobjective implementation and concluded that strong elitism improves convergence by preserving all the promising solutions without suffering the disadvantage of premature convergence.

Preserving all the promising solutions has the advantage that we do not lose nondominated solutions and thus achieve diversity, maybe at a faster rate. SPEA and PAES both use elitism and keep a separate archive of nondominated solutions that is updated for each new generation. The major problem with keeping all nondominated solutions is that the archive may grow very large even for simple problems. To avoid this, SPEA uses clustering to reduce the archive size. (Cunha et al. (1997) also used clustering.) Clustering is an NP-problem that uses heuristics but involves significant additional overhead because every generation involves Pareto-ranking of the combined set and then clustering. *Incrementing Multi-Objective Evolutionary Algorithm* (IMOEA) (Tan et al., 2001) uses a dynamic population size with an archive of nondominated individuals that is also trimmed by clustering.

In our work, we adopt a selection strategy that does not lose nondominated solutions. Elitism is inherent, which gives faster convergence without losing diversity.

2.2.4 Convergence and Performance Metrics

Gauging convergence with a single objective is almost trivial and has been the subject of much work (Aytug and Koehler, 1996, 2000; Hart, 2001; Rudolph, 1998; Rudolph and Agapie, 2000). We perhaps pioneered the work on assessing the convergence of rank-based multiobjective genetic optimization and proposed an initial framework (Kumar and Rockett, 1997). Since then, many performance metrics have been proposed to quantify the obtained Pareto-front (Zitzler and Thiele, 1999; van Veldhuizen, 1999; Tan et al.,2001; van Veldhuizen and Lamont, 2000; Laumanns et al., 1999). Some of these metrics (generational distance, volume of space covered, error ratio measures of closeness of the Pareto-front to the true Pareto-front) are only applicable where the solution is known. Other metrics (ratio of nondominated individuals, uniform distribution) quantify the Pareto-front and can only be used to assess diversity. Thus most of these measures can be divided into two classes: one that measures distance and another that quantifies spread. These may be used in comparing the performance of two or more implementations but are problem specific and involve the selection of parameters. In particular, the relative progress measure of Tan et al. (2001) is noisy and has to be averaged over several generations.

In this work, we gauge global convergence using intra- and inter-island rank histograms and optionally adopt a nonmigrating island model.

2.2.5 Subpopulation Strategy

Much work with single objectives has been done using the island GA model with multiple subpopulations (see Cantú-Paz (2000) for an overview). The steady-state GENITOR algorithm has also been implemented in island mode (Whitley and Starkweather, 1990). Since each island can potentially follow a different solution trajectory, the island model can help to promote genetic diversity. It has the advantage that improved solutions may be obtained with reduced effort and with the scope of parallelizing the computations. The island model may have synchronous/asynchronous migration of individuals — most island model implementations involve migration but migration is not mandatory. A timely migration of proper individuals may help speed convergence but too frequent migration may result in premature convergence. Here we use a nonmigrating island model, particularly for complex problems, to emphasize global convergence. Such an approach does not add additional parameters and is an optional feature of PCGA.

In our work, we adopt many of the above features in such a way that we obtain a Pareto-front for a completely unknown problem with the twin goal of achieving diversity and obtaining convergence with minimal use of parameters and procedures. We stress that we aim for *global* convergence.

3 Pareto Converging Genetic Algorithm (PCGA)

PCGA is a steady-state algorithm and can be seen as an example of $(\mu + 2)$-ES in terms of its selection mechanism. The basic concept of the work presented here is that the individuals are compared against the total population set according to tied Pareto-ranking (Fonseca and Fleming, 1993, 1998), and the population is selectively moved towards convergence by discarding the two lowest ranked individuals in each evolution. In doing so, we require no parameters such as the size of subpopulation in tournament selection or sharing/mating parameters. Additionally, we remove all subjective decision(s) about prioritizing the objectives.

3.1 Selection

Initially, the whole population of size N is ranked using Fonseca and Fleming's ranking algorithm. In this algorithm, ranking with ties is employed; the rank of an individual is equal to the number of individuals by which it is dominated, and all nondominated individuals are given the same rank. If two individuals have the same objective vector, we lower the rank of one of the pair by one; this way, we are able to remove the duplicates from a set of nondominated solutions without loss of generality. For a meaningful comparison of two real numbers in ranking, we restrict the floating-point precision of the objective values to a few units of precision. Fitness is assigned by interpolating from the best individual (rank = 1) to the worst (rank $\leq N$) according to a simple monotonic function to map the individuals onto a conventional roulette wheel for mating selection (Goldberg, 1989). In this work, we have taken fitness as a linear function of tied Pareto ranks. A pair of mates is randomly chosen biased in the sizes of the roulette wheel segments and crossed over and/or mutated to produce offspring. The offspring are inserted into the population set according to their ranks against the whole population. At this stage, the population set also includes the parents, and the population consists of $(N + 2)$ members, so the two lowest ranked individuals of the population are eliminated, and the size restored to N. In case of a tie, i.e., there are more than two solutions of lowest rank, we resolve this by random selection. The process is iterated until a convergence criterion (discussed in a following section) is satisfied.

Before proceeding, we introduce some terminology specific to the PCGA:

1. **Epoch**: An epoch is equal to $N/2$ pairwise evolutions from a population of size N. In this way, the total number of offspring generated in an epoch of PCGA is equal to the number of offspring evolved in a single iteration of a generational algorithm. We use the term *epoch* instead of *generation* to facilitate quantitative comparison between algorithms. In PCGA, each member of the population can participate in reproduction irrespective of its generational age tag.

2. **PCGA Islands**: A randomly initialized population is an *island*, and another randomly initialized population set belongs to another island. We employ this term to emphasize that we are dealing with a sampling of some parent population. All the genetic operations (selection, crossover) are constrained to the bounds of one island only. Each island is a nonmigrating in nature, i.e., individuals do not migrate from one island to another as is the case in a conventional island scheme. There is no cross-fertilization among the individuals of two islands; two (or many) islands however can be merged (only) for the purposes of testing for convergence. The merging of islands is optional; all the results presented here and compared with earlier approaches are drawn from a single island, i.e., one initially randomized population only. We discuss later the merits of merging of islands for complex problems.

3.2 Rank Histograms

We define a Pareto-rank histogram as a frequency distribution of tied ranks in a population. A rank histogram derived from a single island is called an *intra-island rank histogram*, and is generated for successive epochs of evolution within an island. As stated above, there is no cross-fertilization among the individuals of different islands.

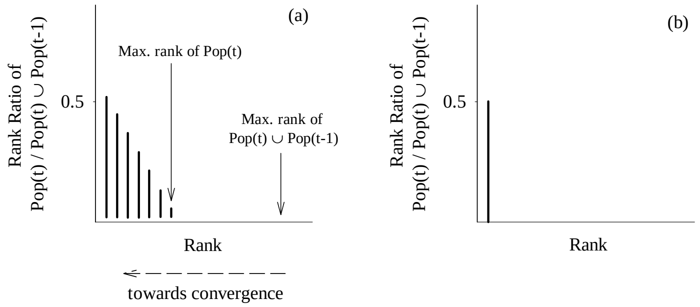

Figure 1: Intra-island rank histogram of a population where the population set consists of (a) both nondominated and dominated individuals and (b) nondominated individuals only.

An additional rank histogram is defined for comparing and merging the solutions only among islands and is termed an *inter-island rank histogram*.

3.2.1 Intra-Island Rank Histogram

The intra-island rank histogram entries are generated from the ratio of the number of individuals at a given rank in the current population to that of combined and reranked populations of the current and the preceding epochs. We are interested, for convergence, in the shift of the set of nondominated solutions between epochs, hence the ratioing of rank entries. Intra-island rank histograms are shown in Figure 1 for two situations: (a) the sample consists of both dominated and nondominated solutions, and (b) the whole sample consists of nondominated solutions only. The rank ratio of the bin belonging to rank unity should remain at 0.5 in an ideal converged state, although the ratio of other ranks can take any arbitrary positive value. A value lower or higher than 0.5 of the bin corresponding to nondominated rank indicates the amount of shuffling among the nondominated individuals between two successive epochs. A decreasing length of histogram tail denotes the movement of total population towards convergence. The histogram of Figure 1(b) shows that the whole population in two successive epochs remains nondominated without any shuffling.

We stress that the situation depicted in Figure 1(b) does not necessarily mean a total converged state but rather that improvement in the population with further iterations is improbable. It thus represents a sensible point at which to stop the computation. We shall examine such a situation while presenting and discussing the results.

3.2.2 Inter-Island Rank Histogram

Among the users of genetic algorithms for solving complex multiobjective optimization problems, there is a common concern whether the obtained solution is close to the true Pareto-front or not. We argue that there is always a certain inheritance of genetic material or content belonging to one island and there may not be much appreciable evolutionary gain beyond a certain number of generations. This implies that the genetic precursors available within a finite population (or island) may be inherently incapable of evolving to the true Pareto-front. Instead, we suggest that alternative genetic mate-

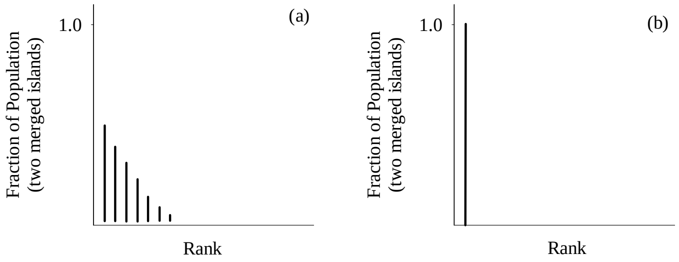

Figure 2: Inter-island rank histogram on combining two islands: (a) some solutions become dominated and (b) none of the solutions is dominated.

rial should be acquired in the form of another island. Each population sample (island) is run to (intra-island) convergence, the obtained solutions are then merged across islands and compared through Pareto-ranking. The shift of the Pareto-front is monitored with an inter-island rank histogram. Clearly, it is only worthwhile comparing the nondominated solutions from two (or more) islands. We thus form an inter-island rank histogram by merging the nondominated individuals from each island.

In a scenario where some of the nondominated solutions of either of the contributing islands are demoted to being dominated, the inter-island rank histogram is depicted in Figure 2(a). The smaller the entry in the bin corresponding to unity rank and the wider the histogram tail, the larger the shift in the set of best solutions and the greater the reshuffling that has taken place. The desired outcome from merging the nondominated members of two or more islands is that none of the nondominated solutions is downgraded to dominated status, and all solutions combine to form a similar or better sampled Pareto-front. Here the inter-island rank histogram of the combined solutions indicates unity in the bin corresponding to nondominated rank (see Figure 2(b)).

PCGA can encourage diversity and competition across islands through reinitialization of whole samples and without copying the nondominated set in the current island. In an elitist genetic algorithm (with generational replacement), the nondominated individuals of the past generations are copied into the current generation. Copying nondominated individuals in the population has the undesired effect of constraining genetic diversity, though it has the benefit that nondominated solutions are not lost. Our approach of combining different islands has the benefit of an elitist model in that the nondominated solutions are not lost but at the same time PCGA does not constrain diversity. We suggest that this approach is particularly suited to complex problem domains of multioptimization and leads the solution space towards convergence.

3.3 Stopping Criteria

A match between two intra-island histograms from successive epochs or match of one histogram with another from an ideal converged state can dictate the stopping criteria. Other metrics may be an information measure of histogram bias towards nondominance, some fitting criteria, or a significance level of testing a null hypothesis. The literature on comparing two frequency coded data distributions contains a number of statistical metrics, e.g., see Fukunaga (1990). By observing the initial behavior of the population on the problem domain and adjusting the parameters of population size, crossover and mutation probabilities, some threshold or probabilistic measure could serve as the stopping criterion. This will give a principled way of examining convergence and stopping the evolution process and is an area of active research. In the present work, we have simply inspected the histograms visually and extracted the heuristics to stop further evolutions within single and multiple islands using intra-rank and inter-rank histograms, respectively.

It has been shown that an infinite number of generations of a simple elitist genetic algorithm can yield an optimal solution (Bhandari et al., 1996). Aytug and Koehler (1996, 2000) established tighter theoretical bounds on the number of GA iterations, and Rudolph and Agapie (2000) found an upper bound on the finite number of function evaluations in a finite search space. However, none of the above ensures diversity. Secondly, it is clearly not a sensible use of computer resources to let a population evolve indefinitely without achieving appreciable gain. Rather, we suggest that a computation should terminate if there is no longer any significant progression towards the Pareto-front. The multi-island population concept is complementary to both (i) a better sampling of Pareto-front and (ii) minimizing the computation involved. The computation of a single island population is terminated on reaching a stopping criterion instead of going through a fixed, and possibly very large number of generations. This saving of computer time can be utilized for the computation of multiple islands, which has the potential for parallelism without the complications of synchronous/asynchronous exchanges between islands.

4 Tests on Analytic Functions

We have tested our PCGA algorithm on three problems considered by earlier researchers on multiobjective optimization. For fair comparison, we have used exactly the same coding and parameters (as far as is known), though they all used the *number of generations* for reporting their results. Instead, we use the comparable term *number of epochs* to quantify our results. An epoch is defined to be $N/2$ pairwise evolutions from a population of size N. Hence, the terms "generation" and "epoch" are related to each other from the point of algorithmic complexity and production of the same number of offspring in each generation. (Using the term "generation" instead of "epoch" in PCGA would be misleading from genetic concepts.)

The unbiased initial populations were randomly generated and decoded within the variable space under consideration, and the experiments were repeated many tens of times for functions F1 and F2 and many hundreds of times for the set F3 each with a different initial population. Typical results selected on the basis of their average qualitative performance (i.e., visualizing a plot for diverse sampling of the Pareto-front drawn in objective space) are presented in the following sections. Since earlier work does not report spreads of results, we omit such results here because it does not aid comparison.

4.1 Function F1

First, we consider the well-known function of Vincent and Grantham (1981), which was used first by Schaffer in his pioneering work on VEGA and subsequently by many other researchers. This is a two-objective problem of a single variable; we use the notation $f_{11}(x)$ for the first objective and $f_{12}(x)$ for the second, where x is an independent variable:

$$Minimize\ f_{11}(x) = x^2 \quad and \quad minimize\ f_{12}(x) = (x-2)^2$$

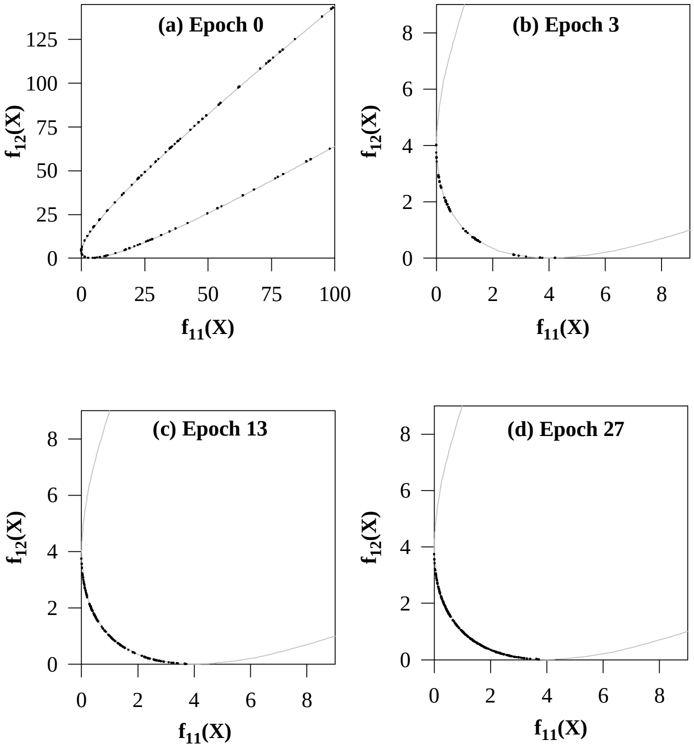

Figure 3: Population (a), (b), (c), and (d) (in objective space) at different stages, i.e., at epochs 0, 3, 13, and 27 for Function F1 (population size = 100, crossover probability = 1.0, mutation probability = 0.0). The population size for plot (d) is 150 (see text for details). Note the scale changes between (a) and the other plots.

and the nondominated region lies within the real-valued range of x lying between (0.0 $\leq x \leq$ 2.0). The decision variable was encoded as a 32-bit binary integer, which was mapped to the initial range of the variable in (-10.0 $\leq x \leq$ 10.0). This is a very simple problem from the perspective of genetic algorithms, and a few solutions already exist in the nondominated region with the randomly initialized population. We tested three different situations: (i) a population size of 100 with zero probability of mutation (as done by Srinivas and Deb (1994)), (ii) a population size of 100 with mutation (for showing the speedup in performance and proving the effectiveness of PCGA), and (iii) a smaller population of 30 (similar to that used by Schaffer (1985) and Horn et al. (1994)).

4.1.1 Population of 100 without Mutation

In this test, the probability of mutation was kept at zero to observe the effectiveness of PCGA alone and for comparing the results with NSGA; the population size was 100 and probability of crossover was 1.0. We also used identical parameters to Srinivas and Deb but without sharing.

The populations are shown in objective space in Figure 3 at its initial stage, and at epoch numbers 3, 13 (at this point the total population lies in the nondominated region), and epoch number 27. The intra-island rank histogram at epoch 3 in Figure 4(a) has a long tail, and this is a measure of the population shifting towards fitter individuals

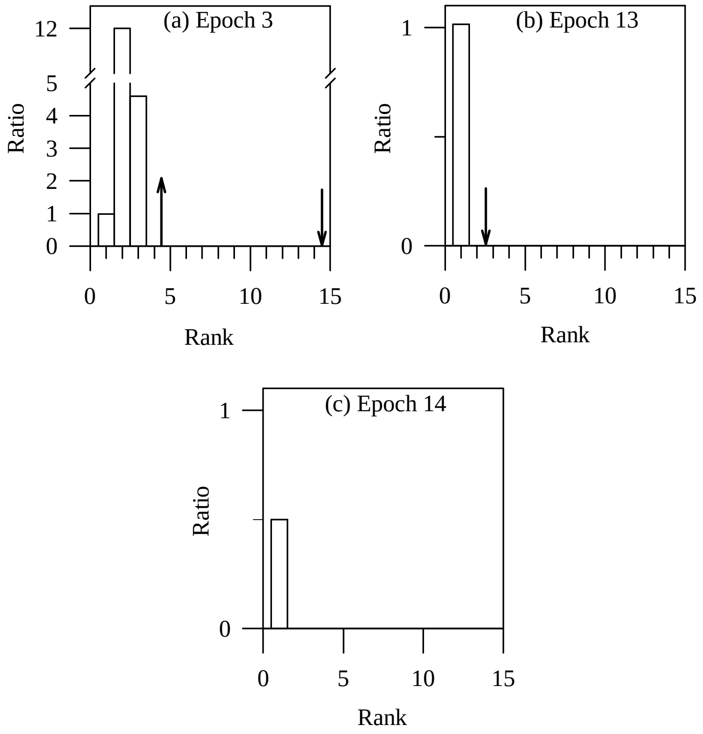

Figure 4: Intra-island rank histograms (a), (b), and (c) for Function F1 (population size = 100, crossover probability = 1.0, mutation probability = 0.0) at epochs 3, 13, and 14, respectively. The up arrow indicates division by zero, and the down arrow marks the largest rank number in the combined population. Note the scale change between (a) and the other plots.

between epochs 2 and 3. The population at epoch 12 has a few dominated individuals but they are transformed to nondominated by epoch 13 (see Figure 4(b)). At epoch 14, when the population consists solely of nondominated solutions, the intra-island rank histogram (Figure 4(c)) indicates total convergence. The simplicity of this problem is also evident from the observation that all the individuals in the final population are nondominated.

Further evolution of the nondominated set produces the solutions shown in Figure 3(d). This experiment of further computation beyond the converged state (epoch 14 and onwards) demonstrates the capability that PCGA can still obtain new nondominated solutions by evolving older nondominated solutions. For this, we need additional sample space in most of the cases, hence we reserved an additional 50% of the initial population size space for evolutions through nondominated samples. During this process, the rate of obtaining newer solutions is slower because of zero mutation probability and the fact that the space left for new solutions to be accommodated in the Pareto-front is limited. By comparison, the population at different stages shows better diversity than that reported in Srinivas and Deb (1994), and in a significantly smaller number of epochs. We needed only 27 epochs, while NSGA (with generational replacement and sharing restrictions) was run for 500 generations without a significant

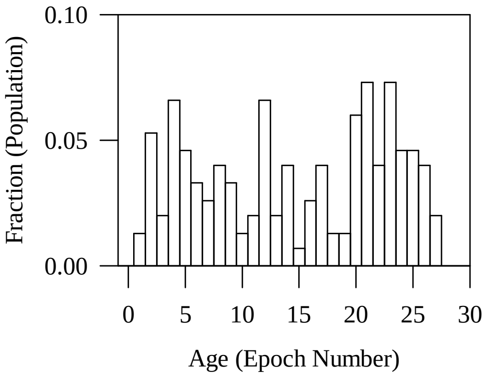

Figure 5: Distribution of the age of appearance of nondominated individuals in the final population for Function F1 (population size = 100, crossover probability = 1.0, mutation probability = 0.0, number of epochs = 27.

improvement beyond 100 generations in the absence of a stopping criterion. This is also one of the merits of PCGA inherent in its termination.

The distribution of the fraction of nondominated solutions generated in each epoch relative to the population after 27 epochs is plotted in Figure 5. Here the age tag of an individual is preserved if the identical individual is evolved again in a later epoch; this way the distribution should be biased towards initial epochs if a larger fraction of nondominated individuals were evolved in the early stages. To the contrary, the population distribution by age in Figure 5 indicates that a significant number of nondominated individuals are evolved in the latter part of the evolution, thus proving the effectiveness of PCGA up to the last epoch. Although the population only consists of nondominated individuals from epoch 13 onwards (Figure 4(c)), the final population contains a large fraction of nondominated individuals with age tags higher than epoch 13. This behavior demonstrates the capability of PCGA to continuously refine the nondominated solutions in successive epochs. We reiterate, this particular problem was solved without mutation and without any sharing, and so the selection procedures of the PCGA algorithm alone have produced to the evident diversity.

In solving Function F1 with a single population, we obtain a very uniformly sampled Pareto-set with only a few small gaps (see Figure 3(d)). In an attempt to demonstrate our approach of multiple populations, we combined the nondominated individuals of three independently evolved populations. On merging, the sampling of the Pareto-front is further improved with fewer gaps, and this is reflected in the respective inter-island rank histograms of Figures 6(a) and 6(b). The histogram entry at nondominated rank is below the ideal converged state of unity in both cases, and a small histogram tail exists indicating that further small improvement is possible. In this way, a multiple population approach for sampling new genetic material in the form of another subpopulation helps in expediting the process of obtaining more solutions and faster convergence.

4.1.2 Population 100 with Mutation

We repeated the above experiment with a small mutation probability of 0.05 in order to assess the potential speedup and the effectiveness of PCGA in a more conventional genetic framework. Again the initial population size was kept at 100. Here, we observe

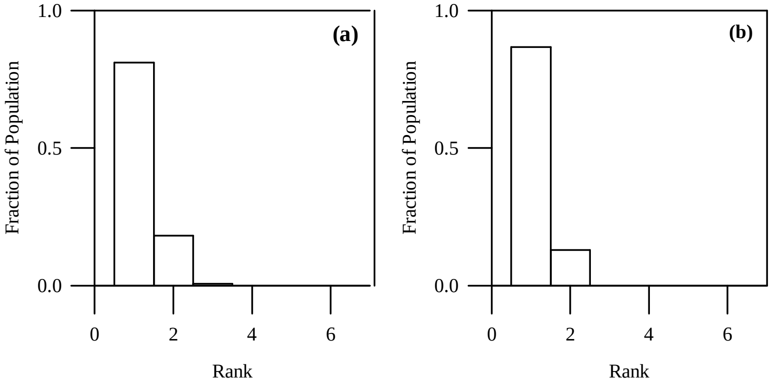

Figure 6: Inter-island rank histograms (a) and (b) for Function F1 (population size = 100, mutation probability = 0.0) on merging the first and second subpopulations, and then merging the third subpopulations.

that only six epochs are required for convergence to nondominated solutions. Seven more epochs generated additional nondominated solutions by evolving existing nondominated solutions only. This demonstrates the effectiveness of PCGA with mutation for increasing diversity, faster convergence and at less computational cost. The corresponding intra-island rank histograms at epochs 3, 6, and 7 are shown in Figure 7.

The histograms of Figure 7 also show a faster convergence than that obtained without mutation (Figure 4). This is an accepted fact of genetic algorithms, and an appropriate mutation probability can be estimated through monitoring speed of convergence. The corresponding age-nondominated population distribution is shown in Figure 8, which also confirms the earlier observation that a significant fraction of nondominated individuals in the final population comes from the later stages of processing.

To examine our multiple population approach, three independent subpopulations were merged and the resulting inter-island rank histograms are shown in Figure 9. The rank histogram shows the movement towards convergence with increasing histogram entries for nondominance and smaller tails. A small number of epochs was required to produce a close-to-ideal converged state. This confirms that a population of 100 individuals is probably more than is really needed for solving this simple problem, and only a single population is enough.

4.1.3 Smaller Population

Next we tested the performance of PCGA on the same function but with a smaller population size of 30. A similar size of population was used first by Schaffer (1985) in VEGA, and then by Horn et al. (1994) while comparing the results of their NPGA with VEGA. Since VEGA reported the results of generations up to 3 only, we present the population at epoch 3 in Figure 10(a). At epoch 3, we are getting some diversity but not a good sampling of the Pareto-front; this is improved at epoch 15. We have stopped the PCGA evolution at this stage for a comparison with NPGA results at generation 200. Though our results at this stage display several gaps on the Pareto-front, the same is true of the NPGA results. PCGA results can be further improved with a few more epochs, while the improvement in NPGA is questionable. Our results are presented without any sharing/mating restrictions, while NPGA involves sharing both in param-

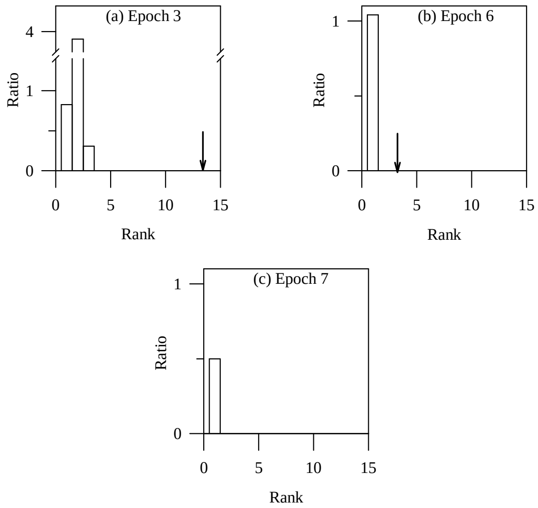

Figure 7: Intra-island rank histograms (a), (b), and (c) for Function F1 (population size = 100, crossover probability = 1.0, mutation probability = 0.05) at epochs 3, 6, and 7, respectively. Down arrow marks the highest rank in the combined population.

eter and objective space. Further, NPGA is very sensitive to selection of parameters, while PCGA is not based on any heuristic selection of parameters. Rather, the obvious genetic parameters like population size, crossover, and mutation probability can be fine tuned while observing the progress on rank histograms. In our approach, we are not losing nondominated solutions once they are generated unless and until they become dominated; hence the disadvantage of VEGA is overcome. The PCGA population at epoch 3 may not be much better than that of VEGA at that stage, but the sampling is much improved with a few more epochs. These are the major obvious advantages of PCGA.

The respective intra-island rank histograms are shown in Figure 11. The nature of the histograms is identical to those shown in Figure 4. This confirms the basis of intra-island rank histograms as provided in Section 3.2 and shown in Figure 1. The age-population distribution displays similar features to Figures 5 and 8 in that later evolutions contribute significantly to the nondominated solutions.

The population size of 30 seems to be just sufficient for this problem. Therefore, in competition with other subpopulations, there is a significant improvement on the Pareto-front, which is reflected by a filling of the gaps. The situation in earlier experiments with population sizes of 100 was different and not much improvement was noticed on combining different solutions, which reflects the advantages of combining multi-islands for improved solution at the cost of a few more epochs.

Thus the improved performance of PCGA on solving the Function F1 problem is clear over the previous approaches. PCGA yields diverse sampling in fewer epochs but without heuristic procedures to suppress genetic drift.

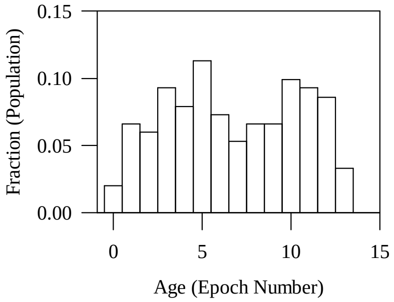

Figure 8: Distribution of the age of appearance of nondominated individuals in the final population for Function F1 (initial population size = 100, crossover probability = 1.0, mutation probability = 0.05, number of epochs = 13.

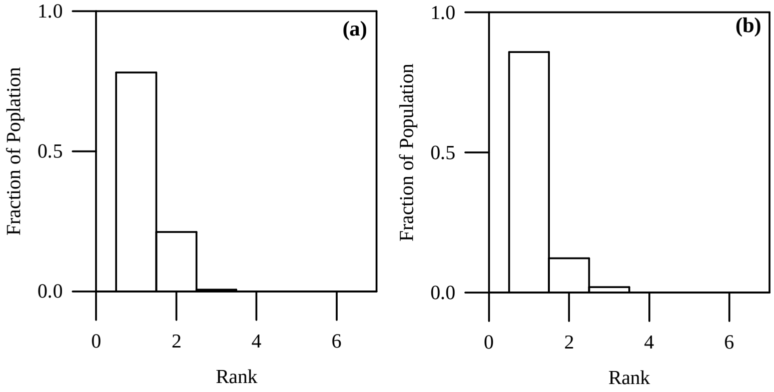

Figure 9: Inter-island rank histograms for Function F1 (initial population size = 100, mutation probability = 0.05). Histogram (a) shows the merging of first and second subpopulations and (b) merging the resultant with the third subpopulation.

4.2 Function F2

Next, we tested PCGA on an n-variable function used by Fonseca and Fleming (1995b) as a minimization problem of two symmetrical objectives:

$$Minimize\ f_{21}(x_1, ..., x_n) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i - 1/\sqrt{n})^2\right)$$
$$Minimize\ f_{22}(x_1, ..., x_n) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i + 1/\sqrt{n})^2\right)$$

This optimization is harder than F1. Objective functions are defined for any number of variables n. Individually, the minima of f_{21} and f_{22} are defined, for every value of n, at:

$$(x_1, x_2, ..., x_n) \equiv (1/\sqrt{n}, 1/\sqrt{n}, ..., 1/\sqrt{n})\ \ for\ minimum\ f_{21}$$
$$(x_1, x_2, ..., x_n) \equiv (-1/\sqrt{n}, -1/\sqrt{n}, ..., -1/\sqrt{n})\ \ for\ minimum\ f_{22}$$

Due to symmetry of f_{21} and f_{22}, the actual Pareto-front corresponds to all the points on the curve given by:

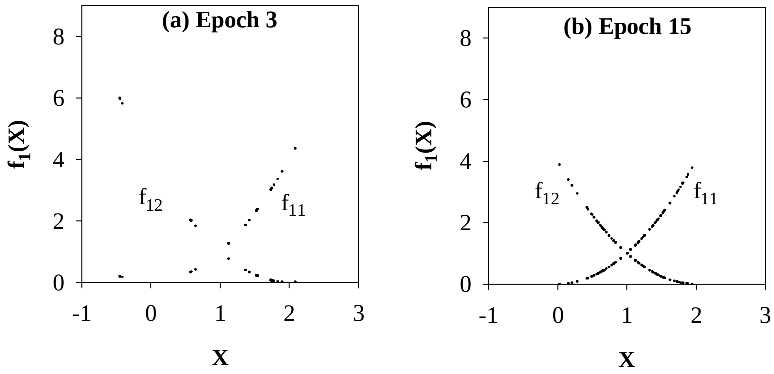

Figure 10: Population at epochs 3 and 15 for Function F1 (population size = 30, crossover probability = 1.0, mutation probability = 0.05). Population for plot (b) is 45.

$$(x_1 = x_2 = ... = x_n) \wedge (\forall_i)\,(-1/\sqrt{n} \leq \; x_i \leq 1/\sqrt{n})$$

Where appropriate, we have taken the same genetic parameters for PCGA as far as is known (Fonseca and Fleming, 1995b) for a comparison with MOGA:

Number of variables (n)	8
String length (binary code)	16 bits per variable
Decoding range of variables	[-2.0, 2.0]
Population size	100
Probability of Crossover	1.0
Probability of Mutation	0.05
Mating/Sharing	Nil

The degree of difficulty of this problem is evident from the initial population itself, where almost all the points are concentrated on one extreme of the objective space. Though there are two nondominated points in the initial population, they are far away from the actual Pareto-front. The initial population and that at epochs 50 and 100 are shown in Figure 12. The nondominated individuals are plotted with filled black circles, dominated points with grey circles, and the actual Pareto-front is marked with a grey curve. It is very evident from the obtained populations at various epochs that the whole population is progressively moving towards convergence. This is a distinct feature of PCGA and in contrast to other approaches.

Fonseca and Fleming (1995b) presented MOGA results for the population at generation 100 only, both with and without sharing. They obtained better sampling with mating restrictions but not as good as desired. With PCGA, the Pareto-front is uniformly sampled with nondominated solutions in very close proximity to the actual front; this has been achieved without sharing. Even the dominated points are proximate to the Pareto-front (but nonetheless still dominated). With a few more epochs, the Pareto-front, which is progressively moving towards convergence, can be further improved with PCGA, but we suspect this will be a very slow process with generational

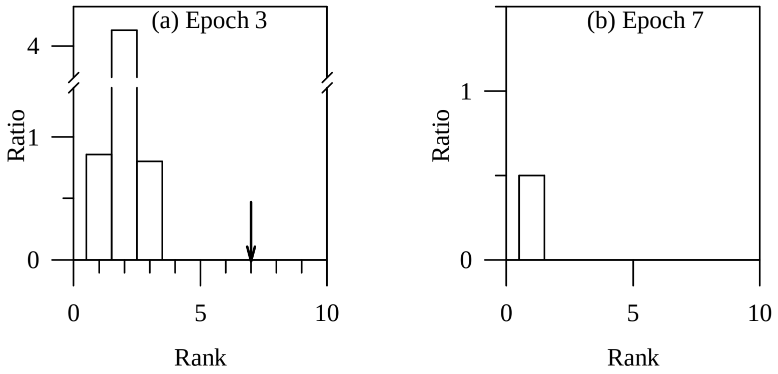

Figure 11: Intra-island rank histograms for Function F1 (population size = 30, crossover probability = 1.0, mutation probability = 0.05) at epochs 3 and 7, respectively. Down arrow marks the highest rank in the combined population.

replacement type of genetic strategies such as MOGA.

The corresponding rank histograms display results entirely consistent with previous observations. At epoch 100 there is a small histogram tail confirming the results in Figure 12 that the population has not quite converged.

The population age distribution for this problem is unlike those for previous problems in that nondominated individuals in the final population only start to be generated from epoch 38 onwards — this reflects the challenging nature of the F2 problem. The population state at epoch 50 of Figure 12 shows a significant fraction of nondominated individuals very near to the Pareto-front, but the age distribution plot indicates that the contribution of such individuals to the final population is only 17%. This reconfirms that the PCGA evolution strategy produces better and better offspring in each epoch. It is this ability of PCGA that leads the population towards convergence.

At epoch 100, we have stopped further evolution for comparison with the results obtained with mating restrictions by Fonseca and Fleming (1995b), though the population could further be improved with some more epochs. Their results without mating restrictions are concentrated in a very small region of the Pareto-front. On comparing Fonseca and Fleming's results with mating restrictions with the results of PCGA, PCGA achieves diversity without mating restrictions. Continuing with our approach of multiple subpopulations, we independently generated another two populations and merged the nondominated solutions from each. The solutions are further improved with a more uniform sampling and gap filling. The inter-island rank histograms also indicate the situation moving towards convergence but more solutions could be obtained for a total filling of the front.

4.3 Function F3

Third, we considered a bimodal function $g(x_2)$ given by:

$$g(x_2) = 2.0 - \exp\left\{-\left(\tfrac{x_2-0.2}{0.004}\right)^2\right\} - 0.8\ \exp\left\{-\left(\tfrac{x_2-0.6}{0.4}\right)^2\right\};\ g(x_2) > 0$$

For ($0 < x_2 < 1$), $g(x_2)$ is a function with a broad local minima at x_2 = 0.6 and a spike-like global minima at x_2 = 0.2. Retargeting this single objective problem to a multiobjective one (see Deb (1999)), the corresponding two-objective problem having

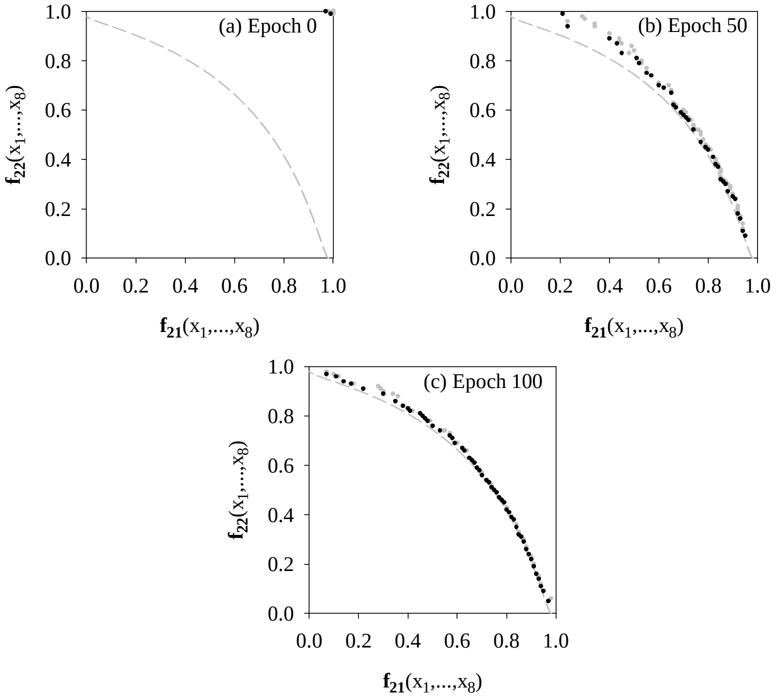

Figure 12: Population (a), (b), and (c) at different stages of evolution, i.e., at epochs 0, 50, and 100, respectively for Function F2 (population size = 100, crossover probability = 1.0, mutation probability = 0.05).

two variables x_1 (> 0) and x_2 is:

$$Minimize\ f_{31}(x_1, x_2) = x_1 \quad and \quad minimize\ f_{32}(x_1, x_2) = \frac{g(x_2)}{x_1}$$

For a fixed value of $g(x_2)$, each f_{31}–f_{32} plot is a hyperbola (see Figure 13). So for the two-objective f_{31}–f_{32} problem, we get one local and global Pareto-front for the local and global solutions of $g(x_2)$, respectively. (See Deb (1999) for function characteristics and the related theorem.) The Pareto-optimal solutions vary in x_1 value (Figure 13: the local and global Pareto-optimal fronts are shown by grey curves). With the randomly generated initial populations (population size = 60), almost all the individuals are close to the local optimal front and very few to the global front. (We generated many hundred of sets of populations and only a few tens had a single individual closer to the global front; this is a typical problem due to the bimodality of the function.) Deb (1999) studied this problem using NSGA; we have taken the same genetic parameters for PCGA for a comparison with NSGA:

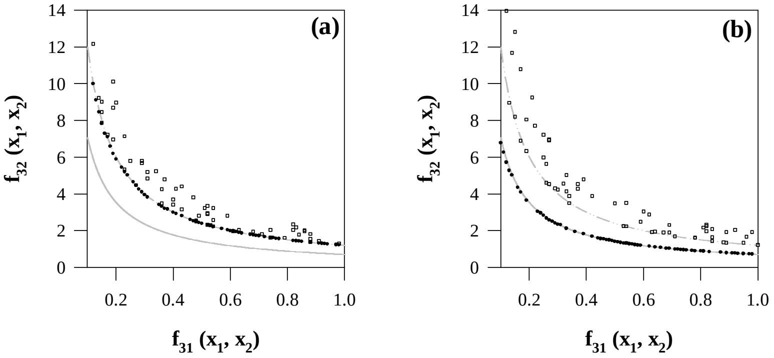

Figure 13: Two typical sets (a) and (b) of populations converging to the local and global minima, respectively; the local and global optimal fronts are shown by grey curves. In (a), the whole population has converged to the local Pareto-front in an average of 19 epochs per run. The population in (b) has converged to the global Pareto-front in an average of 31 epochs per run. The average across both minima was 23.8 epochs.

String length (binary code)	20 bits per variable
Decoding range of variables	x_1 [0.1, 1.0] and x_2 [0.0, 1.0]
Population size	60
Probability of Crossover	1.0
Probability of Mutation	0.0
Mating/Sharing	Nil

We ran the experiments tens of hundreds of times so that we can make generalized observations about the results. For such exhaustive experiments, we did not change the genetic parameters, only the initial populations. Based on the results, we can make the following observations. For each of the runs, the whole population of sixty individuals converged (all nondominated individuals) within the range of 12 to 41 epochs with an average of 23.8 epochs per run. (As with results reported earlier Deb (1999), some of the populations converged to the local Pareto-front and others to global front.) This is a significant improvement over NSGA, which was run for 100 generations in the absence of a convergence criterion. We achieved a superior sampling of the Pareto-fronts to that reported in Deb (1999). Two typical sets of results are shown in Figure 13 (one where the solution converges to the local minimum and the other where global minimum is found). The initial populations in Figure 13 are shown with open square symbols and final populations with the filled circles. In most of the runs, the initial population did not contain even a single individual in the narrow global basin.

For some solutions, the population gets trapped in the local Pareto-front. We were able to locate the global Pareto-front in 36 - 44% of the independently initialized runs. This observation is identical to Deb's. The fact that we had a similar success rate to Deb's NSGA in finding the local-to-global Pareto-front suggests that this ratio may be an intrinsic feature of the problem connected to a density-of-states type argument in the objective space. This is in tune with the observations of many researchers that multiobjective GAs can have difficulty with even a simple bimodal problem (e.g., Deb (1999, p. 10)). PCGA, however, produces a diverse sampling of the Pareto-front without niching, and these results on F3 are consistent with what we have found for functions F1 and F2 and reported in previous subsections.

5 Discussion

The PCGA algorithm presented in this paper empirically demonstrates two distinctive features. First, it achieves diverse sampling of the Pareto-front without niching. Second, it introduces the notion of a gauge on convergence. While doing so, at every stage in PCGA, the population is compared in the most general sense. There is no subjective consideration of the priorities of the objectives, nor are they randomly picked (Fourman, 1985; Kursawe, 1991). In the absence of a true scalar fitness, we argue that the only way to compare a parent and its offspring is a rank-based metric. In practice, Pareto-ranks are scaled arbitrarily to fitness values (fitness is a direct measure of reproductive potential of the individuals). Moreover, we use Pareto-ranks, albeit heuristically, for gauging the convergence.

In the following subsections, we discuss the results together with the characteristics of PCGA with regard to how they help achieve diversity without explicitly employing a diversity preserving mechanism.

5.1 Achieving Diversity

PCGA produces only a single pair of offspring in one iteration; it is a steady-state algorithm and is elitist in nature. In contrast, almost all the other MO algorithms produce an entire new population at a single generation; the fitter individuals get more chances at reproduction, and genetic drift has to be suppressed through some explicit diversity preserving mechanism. This approach has the advantage that fitter individuals are liable to produce even fitter offspring, but the demerit that the functional mapping from ranks to fitness is important since it can strongly influence genetic drift. In this work, we have taken fitness as a simple linear function of Pareto-ranks. The effect of the steady-state algorithm on achieving wider sampling is further discussed in Subsection 5.1.2.

Though PCGA does not replace the whole generation in a single step, each pair of offspring is compared against the whole population. This is in contrast to tournament-based selection approaches (NPGA, SPEA, and the algorithm of Ritzel et al. (1994), which needs a comparison set). Such selection approaches are affected by the user-defined size of the comparison set. In a pairwise comparison, if the offspring are compared only against their parents, this will result in a tie in most practical situations as is the case with NPGA, and additional safeguards of mating/sharing restrictions have to be employed for resolving the tie.

5.1.1 Progressive Advancement of the Solution Front

Another feature of PCGA's steady-state strategy is that at every population comparison, we discard the two lowest ranked individuals. With every evolution, we either get superior offspring or at worst, the same population fitness. Thus, with every evolution, the population advances towards the Pareto-front or in the worst case, remains static. The common genetic operators of crossover and mutation ensure diversity (shown in the results included in this paper).

PCGA does not at any stage lose nondominated solution(s). We argue that replacing a nondominated member of a population with a dominated member is not a sensible approach in a generalized multiobjective optimization. In PCGA, a (erstwhile nondominated) solution can only be discarded if it is no longer nondominated. The results of discarding potentially valuable genetic material can be serious and are reported in Srinivas and Deb (1994), where a sufficient population of nondominated solutions at a given evolutionary stage is reduced at a later stage to a few solutions on the Pareto-front, a well-known drawback of Schaffer's VEGA approach even with mating restrictions. The tournament-based selection approaches are partly affected by this dilemma: the selection is guided by the size of the tournament set and thus controls the convergence — nondominated solutions may be eliminated, which results in slow convergence. Nonetheless, for such approaches, mating restrictions do have the potential to suppress genetic drift.

PCGA has elitism built in to the algorithm. In a single-objective optimization, elitism may have the disadvantage of premature convergence, but in the case of multiobjective optimization, Parks and Miller (1998) have shown that elitism improves the performance. Many of the recent algorithms (SPEA, PAES) have adopted an elitist approach to retain the nondominated solutions but use clustering as a post process to control the size of the nondominated set.

5.1.2 Reduced Range of Pareto-Ranks

Initially, when the population is randomly initialized, the range of the ranks is quite large (Figures 4, 7) and the fitter individuals (on the roulette wheel) have better chances of being selected for breeding. PCGA retains the merit that fitter individuals are able to produce better offspring, yet the functional consideration of the mapping is not significant. As the population is progressively moved towards convergence with the elimination of the weakest pair at each evolution, the dynamic range of the rank is consistently reduced (Figures 4, 7), and each individual has a greater chance of reproduction. The population at different stages of reproduction (number of epochs) shown in different results confirms this behavior for PCGA.

Taking into account the "tied" ranks, this is not the case with generational replacement strategies. In a generational GA, in addition to getting better solutions one also produces (and retains in the population) many worse solutions (for example, two very fit parents producing a runt offspring), and thus generational GAs maintain a greater range of ranks. If the rank-to-single-scalar fitness mapping is the same, this greater range of ranks translates into greater range of scalar fitness values, and so there is a tendency (bias) for generational GAs to select only the fittest individuals because they occupy so much of the roulette wheel. Nonetheless, this is in tune with the principles of genetic algorithms that fitter individuals are given better chances of breeding but at the cost of additional mating restrictions. In a steady-state PCGA, the population is always improving (or at least never getting worse) and so the range of ranks is more compressed giving a greater probability of all individuals breeding. PCGA gives all solutions a greater chance to breed thus leading to genetic diversity. The dynamics of the two evolutions (steady-state and generational) are different. With increasing numbers of epochs, the mating pool for a steady-state GA contains a compressed range of ranks, while the mating pool for a generational GA has larger range of ranks giving less fit individuals a lesser chance at breeding.

Although we believe this is a plausible argument, it remains a conjecture. Tracking the lineage of individuals and proving the connection between reduced ranges of tied ranks and better diversity is the subject of future research. Observationally, PCGA does maintain diversity without niching; exactly why is the subject of further investigation.

However, the dynamics of steady-state evolution with compressed range of ranks is analogous to explicit mating restriction. (By definition, mating restrictions (Deb and Goldberg, 1989) try to address the fact that individuals too different from each other are less likely to produce fit offspring through mating than similar individuals.) We believe, in PCGA, the compressed range of ranks is inherently producing the same effect as a mating restriction but without user-defined parameters. So, we can say that the PCGA's selection strategy does not explicitly need any diversity preserving techniques but employs implicit mating restrictions. Similarly, it can be argued that compressed range of ranks is producing the same effects as fitness sharing.

5.2 Comparing PCGA's Island Model with the Canonical Subpopulation Strategy

Swapping or migration of individuals from one island to another is the key feature of a canonical subpopulation strategy. Subpopulation strategy, commonly known as the island model, distributed or parallel GAs, is used mainly to get superior solutions, speed convergence, and parallelize the computing efforts — the success of such a strategy depends on many factors. Usually, the population size in a single island is comparatively small. Small islands are run to convergence (i.e., where the genetic diversity in the small population has disappeared), and then genetic material is synchronously/asynchronously mixed/swapped to re-introduce diversity (Braun, 1990; Whitley and Starkweather, 1990; Cantú-Paz, 2000). In many cases, however, islands have been found to give no speed-up. Where islands do work, it is known that isolated (i.e., no migration) islands can converge to different local optima at which point comparing the island populations can yield valuable information about the dynamics of the process.

We employed subpopulations of significantly larger sizes than is normal with conventional islands model although we did not employ any migration. We stress that we use isolated islands not to speed convergence but as a test on the quality of the final solutions; in the process, we are able to escape from local minima (Figure 13).

5.3 Analyzing Computational Load

PCGA has the additional computational overhead of inserting the offspring in the population and generating the rank histogram but it requires fewer epochs and eliminates heuristically selected parameters and the associated computations. The overall computations in genetic search can be divided broadly in two categories: one involving genetic manipulations, and those of objective evaluations. The genetic computations involve breeding operations, sharing, and ranking. Implementation of sharing in a high-dimensional objective space can be computation intensive. PCGA eliminates the need for sharing but adds additional overheads of inserting offspring into an already ranked list. Ranked generational replacement techniques on the other hand require complete reranking of the population after every iteration. The objective evaluations are problem specific but typically, in real-life problems, dominate the time required for most genetic optimizations.

5.4 Monitoring Convergence

There is no need to run a problem through a fixed number of iterations if it can be stopped when no further appreciable gain is being achieved. By examining the rate of convergence, the number of epochs can be further reduced by monitoring the state of an island with successive rank histograms and tuning any problem-specific parameter(s). For example, very large initial populations can be avoided, which can significantly reduce the computation and memory requirements. Similarly, the other parameters, e.g., probabilities of crossover and mutation, can be fine-tuned and their effects on population evolution can be examined. Monitoring convergence would help not only to avoid wasting CPU time in producing further generations for an optimization that has already converged, but also in knowing whether a particularly stubborn problem is still far from convergence. This is discussed in next subsection.

Although in the present work we have interactively stopped further population evolution using the rank histogram, the process could be automated by incorporating a suitable statistical test on the successive rank histograms. The rank histogram test of convergence introduced here is somewhat analogous to the aims of Fisher's F-test (Fukunaga, 1990) from standard statistical hypothesis testing, where a number of samples are compared for consistency (the null hypothesis).

5.5 Application to Real Problems

Although the results presented in this paper are for well-known multiobjective benchmark problems, elsewhere we have reported the successful application of PCGA to a number of real-world optimization problems of complex nature, particularly to problems of NP nature. For example, a classification problem in seven-dimensional objective space for near-optimal partitioning of a pattern space using variable-length, real-number chromosomes is presented in Kumar and Rockett (1998). We have applied PCGA to codebook construction for vector quantization (Kumar, 2000) and communication network design (Kumar et al., 2002). PCGA has also been applied to other applications, e.g., the study of continuous casting mold process (Chakraborty et al., 2001). Taken together with the results presented here, we have grounds to believe that PCGA is applicable to a wide range of multiobjective optimization problems.

6 Conclusions

In this paper, we have presented a novel approach to multiobjective optimization by genetic algorithms. Our PCGA is a steady-state algorithm and uses a (μ + 2)-ES selection scheme along with Pareto-ranking. The algorithm naturally prevents genetic drift while uniformly sampling the Pareto-front and eliminating heuristically selected mating/sharing parameters. A systematic, albeit heuristic, approach based on rank histograms for assessing convergence was discussed. PCGA was tested on three benchmark problems as a proof-of-concept, and the results compared with those of earlier approaches. The Pareto-converging approach is promising for complex problems involving optimization on a large number of competing objectives where independent evolution with isolated multiple islands of population can be employed effectively in difficult optimization domains; PCGA has been applied to a number of real-world multiobjective problems of NP-nature (and the results have been reported elsewhere).

Thus PCGA empirically demonstrates achieving diverse sampling of the Pareto-front without niching and sharing. Furthermore, we have introduced the notion of monitoring convergence by advancement of the population towards the Pareto-front. Other contemporary techniques for multiobjective optimization could easily adopt many of these approaches. Analysis on maintaining diversity without niching is the subject of further investigation. Some other related studies, e.g., setting some statistical tests on intra- and inter-island rank histograms, are suggested for future research.

Acknowledgments

The authors would like to thank Darrell Whitley and the anonymous reviewers for their valuable comments and helpful suggestions. The major part of this work was done while Dr. Kumar was at the University of Sheffield, and he acknowledges support from the Commonwealth Scholarship Commission and Ministry of Human Resource Development, Government of India, project during the course of this work.

References

Aytug, H. and Koehler, G. J. (1996). Stopping Criteria for Finite Length Genetic Algorithms. *ORSA Journal on Computing*, 8(2):183–191.

Aytug, H. and Koehler, G. J. (2000). New Stopping Criterion for Genetic Algorithms. *European Journal of Operational Research*, 126:662–674.

Bäck, T. (1995). Generalized convergence models for tournament- and (μ, λ)- selection. In Eshelman, L. S., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 2–8, Morgan Kaufmann, San Mateo, California.

Bäck, T. and Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1):1–23.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In Grefenstette, J. J., editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 14–21, Lawrence Erlbaum, Hillsdale, New Jersey.

Ben-Tal, A. (1980). Characterization of Pareto and lexicographic optimal solutions. In Fandel, G. and Gal, T., editors, *Multiple Criteria Decision Making: Theory and Application*, Lecture Notes in Economics and Mathematical Systems, 17:1–11, Springer-Verlag, Berlin, Germany.

Bentley, P. J. and Wakefield, J. P. (1997). Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In Chawdhry, P. K., Roy, R., and Pant, R. K., editors, *Soft Computing in Engineering Design an Manufacturing*, pages 231–240, Springer Verlag, London, UK. In *Proceedings of the Second On-line World Conference Soft Computing in Engineering Design and Manufacturing (WSC2 June)*.

Bhandari, D., Murthy, C. A., and Pal, S. K. (1996). Genetic Algorithm with Elitist Model and its Convergence. *International Journal on Pattern Recognition and Artificial Intelligence*, 10:731–747.

Braun, H. (1990). On solving travelling salesman problems by genetic algorithms. In Schwefel, H.-P. et al., editors, *Proceedings of the First Workshop Parallel Problem Solving from Nature, Lecture Notes in Computer Science*, 496:129–133..

Cantú-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms (Genetic Algorithms and Evolutionary Computation 1)*. Kluwer Academic Publishers, Boston, Massachusetts.

Cavicchio, D. J. (1970). *Adaptive Search using Simulated Evolution*. Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan.

Chakraborty, N., Kumar, R., and Jain, D. (2001). A Study of the Continuous Casting Mold Using a Pareto Converging Genetic Algorithm. *Applied Mathematical Modelling*, 25:287–297.

Chipperfield, A. and Fleming, P. J. (1996). Multiobjective Gas Turbine Engine Controller Design using Genetic Algorithms. *IEEE Transactions on Industrial Electronics*, 43:583–587.

Coello, C. A. C. (2001). List of References on Evolutionary Multiobjective Optimization. Available at: `http://www.lania.mx/ ccoello/EMOO/EMOObib.html`

Coello, C. A. C. and Toscano, G. (2001). A micro genetic algorithm for multiobjective optimization. In Zitzler, E. et al., editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, 1993:126–140.

Cunha, A. G., Oliveira, P., and Covas, J. A. (1997). Use of genetic algorithms in multicriteria optimization to solve industrial problems. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 682–688, MIT Press, Cambridge, Massachusetts.

Deb, K. (1999). Multiobjective Genetic Algorithms: Problem Difficulties and Construction of Test Problems, *Evolutionary Computation*, 7(3):1–26.

Deb, K. (2001). *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK.

Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In Schaffer, J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, Morgan Kaufmann, San Mateo, California.

De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan.

De Jong, K. A. and Sarma, J. (1993). Generation gap revisited. In Whitley, L. D., editor, *Foundations of Genetic Algorithms II*, pages 19–28, Morgan Kaufmann, San Mateo, California.

Eshelman, L. J. (1991). The CHC adaptive search algorithm. How to have safe search when engaging in non-traditional genetic recombination. In Rawlins, G., editor, *Foundation of Genetic Algorithms I*, pages 265–283, Morgan Kaufmann, San Mateo, California.

Eshelman, L. J. and Schaffer, J. D. (1991). Preventing premature convergence in genetic algorithms by preventing incest. In Belew, R. and Booker, L., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 115–122, Morgan Kaufmann, San Mateo, California.

Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalisation. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, Morgan Kaufmann, San Mateo, California.

Fonseca, C. M. and Fleming, P. J. (1995a). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16.

Fonseca, C. M. and Fleming, P. J. (1995b). Multiobjective genetic algorithms made easy: selection, sharing and mating restriction. In *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, IEE Conference Publication No. 414, IEE, London, UK.

Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: a Unified Formulation, *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 28:26–37.

Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153, Lawrence Erlbaum, Hillsdale, New Jersey.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Second Edition. Academic Press, San Diego, California.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G., editor, *Foundations of Genetic Algorithms I*, pages 69–93, Morgan Kaufmann, San Mateo, California.

Goldberg, D. E., Korb, B., and Deb, K. (1989). Messy Genetic Algorithms: Motivation, Analysis, and First Results. *Complex Systems*, 3:493–530.

Hajela, P. and Lin, C. Y. (1992). Genetic Search Strategies in Multicriteria Optimal Design. *Journal of Structural Optimization*, 4:99–107.

Hart, W. E. (2001). A Convergence Analysis of Unconstrained and Bound Constrained Evolutionary Pattern Search. *Evolutionary Computation*, 9(1):1–23.

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 82–87, IEEE Press, Piscataway, New Jersey.

Hwang, C. L. and Masud, A. S. M. (1980). Multiple objective decision making – methods and applications: a state of the art survey. *Lecture Notes in Economics and Mathematical Systems*, Volume 164, Springer-Verlag, Berlin, Germany.

Knowles, J. D. and Corne, D. W. (2000). Approximating the Non-dominated Front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172.

Kumar, R. (2000). Codebook design for vector quantization using multiobjective genetic algorithms. In *PPSN/SAB Workshop on Multiobjective Problem Solving from Nature*, College de France, Paris.

Kumar, R. and Rockett, P. I. (1997). Assessing the convergence of rank-based multiobjective genetic algorithms. In *Proceedings of the Second IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 97)*, pages 19–23, IEE Conference Publication No. 446, IEE, London, UK.

Kumar, R. and Rockett, P. I. (1998). Multiobjective Genetic Algorithm Partitioning for Hierarchical Learning of High-Dimensional Pattern Spaces: A Learning-Follows-Decomposition Strategy. *IEEE Transactions on Neural Networks*, 9:822–830.

Kumar, R., Parida, P. P., and Gupta, M. (2002). Topological design of communication networks using multiobjective genetic optimization. In *Proceedings of Congress Evolutionary Computation (CEC-2002)*, pages 425–430, IEEE Press, Piscataway, New Jersey.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature I*, Springer Verlag, Berlin, Germany. *Lecture Notes in Computer Science*, 496:193–197.

Larcombe, S. P. (1996). *Implementing Heterogeneous Systems in a Three Dimensional Packaging Technology*. Ph.D. Thesis, Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, UK.

Laumanns, M., Rudolph, G., and Schwefel, H.-P. (1999). Approximating the Pareto set: concepts, diversity issues and performance assessment. Technical Report CI-72/99, University of Dortmund, Dortmund, Germany.

Loughlin, D. H. and Ranjithan, S. (1997). The neighborhood constraint method: a genetic algorithm-based multiobjective optimization technique. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 666–673, MIT Press, Cambridge, Massachusetts.

Mahfoud, S. W. (1992). Crowding and preselection revisited. In *Parallel Problem Solving from Nature II*, pages 27–36, Springer Verlag, Berlin, Germany.

Miller, B. L. and Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. IlliGAL Report No. 95006, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois.

Neri, F. and Saitta, L. (1995). Analysis of genetic algorithms evolution under pure selection. In Eshelman, L. S., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 32–39, MIT Press, Cambridge, Massachusetts.

Pareto, V. (1896). *Cours d'Economie Politique*, Rouge, Lausanne, France.

Parks, G. T. and Miller, I. (1998). Selective breeding in a multiobjective optimization. In Eiben, A. E. et al., editors, *Parallel Problem Solving from Nature V*, Springer Verlag, Berlin, Germany. *Lecture Notes in Computer Science*, 1498:250–259.

Ritzel, B. J., Eheart, J. W., and Ranjithan, S. (1994). Using Genetic Algorithms to Solve a Multiple Objective Groundwater Pollution Containment Problem. *Water Resources Research*, 30:1589–1603.

Rogers, A. and Prügel-Bennett, A. (1999). Genetic Drift in Genetic Algorithm Selection Schemes. *IEEE Transactions on Evolutionary Computation*, 3:298–303.

Rudolph, G. (1998). On a multiobjective evolutionary algorithm and its convergence to the Pareto set. In *Proceedings of the Fifth IEEE Conference on Evolutionary Computation*, pages 511–516, IEEE Press, Piscataway, New Jersey.

Rudolph, G. and Agapie, A. (2000). Convergence properties in some multiobjective evolutionary algorithms. In *Proceedings of the Congress of Evolutionary Computation*, pages 1010–1016, IEEE Press, Piscataway, New Jersey.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Lawrence Erlbaum, Hillsdale, New Jersey.

Shapiro, J., Bennett, A. P., and Rattray, M. (1994). A statistical mechanical formulation of the dynamics of genetic algorithms. In Fogarty, T. C., editor, *Evolutionary Computing*, Springer-Verlag, Berlin, Germany. *Lecture Notes in Computer Science*, 865:119–129.

Shaw, K. J. and Fleming, P. J. (1996). Initial study of multi-objective genetic algorithms for scheduling the production of chilled ready meals. In *Proceedings of the Second International Mendel Conference on Genetic Algorithm*, Technical University of Brno, Brno, Czech Republic.

Srinivas, N. and Deb, K. (1994). Multiobjective Optimization using Non-dominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(4):221–248.

Stanley, T. J. and Mudge, T. (1995). A parallel genetic algorithms for multiobjective microprocessor design. In Eshelman, L. S., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 597–604, MIT Press, Cambridge, Massachusetts.

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley and Sons, New York, New York.

Tan, K. C., Lee, T. H., and Khor, E. F. (2001). Evolutionary Algorithms with Dynamic Population Size and Local Exploration for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 5:565–588.

Thierens, D. and Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. In Davidor, Y., Schwefel, H.-P., and Männer, R., editors, *Parallel Problem Solving from Nature III*, Springer Verlag, Berlin, Germany. *Lecture Notes in Computer Science*, 866:119–129.

Valenzela-Rendón, M. and Uresti-Charre, E. (1997). A non-generational genetic algorithm for multiobjective optimization. In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 658–665, MIT Press, Cambridge, Massachusetts.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classification, Analyses and New Innovations*. Ph.D. Thesis, AFIT/DS/ENG/99-01, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Van Veldhuizen, D. A. and Lamont, G. B. (2000). Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*, 8(2):125–147.

Vincent, T. L. and Grantham, G. J. (1981). *Optimality in Parametric Systems*, John Wiley and Sons, New York, New York.

Whitley, L. D. (1989). The GENITOR algorithm and selective pressure: why rank-based allocation of reproductive trials is best. In Schaffer, J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–121, Morgan Kaufmann, San Mateo, California.

Whitley, L. D. (2001). An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls. *Information and System Technology*, 43:817–831.

Whitley, L. D. and Starkweather, T. (1990). GENITOR II: a Distributed Genetic Algorithm. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:189–214.

Wolpert, D. H. and Macready, W. G. (1997). No free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82.

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3:257–271.