4 SIMPLIFYING OPTIMIZATION IN PARETO SPACE

4.1 Introduction

Chapter 3 showed that with proper design and parameterization, the NSGA is able to accurately quantify 2 dimensional tradeoffs. This chapter builds on Chapter 3 by introducing a design methodology for the NSGA-II, a second generation EMO genetic algorithm (see *Deb et al.* 2000) that improves upon the NSGA by (1) invoking a more efficient nondomination sorting algorithm, (2) eliminating the sharing parameter (i.e., the niche radius), and (3) adding an implicitly elitist selection method that greatly aids in solving high order problems (i.e., problems with more than 2 objectives). The NSGA-II is the focus of this work because the algorithm has been shown to perform as well or better than the other second generation EMO algorithms on difficult, high order problems (see *Zitzler et al.* 2001 and *Deb et al.* 2001). The NSGA-II design methodology presented in this chapter builds on previous GA design methodologies (*Reed et al.* 2000b, *Lobo* 2000, and *Reed et al.* 2001) to introduce a multi-population approach that automates parameter specification while significantly reducing the computational effort required to solve multiobjective applications. The design methodology fully exploits the efficiency of the NSGA-II to enable the automatic solution of a new class of high order multiobjective applications in which users can select, understand, and balance more than two performance criteria (see Chapter 6). Readers familiar with Chapter 3 can skip forward to Section 4.3 because Section 4.2 describes the same test case and problem formulation.

4.2 Monitoring Application

4.2.1 Test Case Data

The NSGA-II design methodology presented in this chapter is demonstrated on the same long-term monitoring (LTM) application from Chapter 3. The test was developed using data

drawn from a 38 million-node flow-and-transport simulation performed by *Maxwell et al.* (2000). The simulation provided realistic historical data for the migration of a hypothetical perchloroethylene (PCE) plume in a highly heterogeneous alluvial aquifer. The hydrogeology of the test case is based on an actual site located at the Lawrence Livermore National Laboratory in Livermore, California, currently being managed under the United States' Comprehensive Environmental Response, Compensation and Liability Act (CERCLA) program. Data were provided for a total of 50 hypothetical sampling locations within the 20-well multi-level monitoring network. The data represent a snapshot in time, 10 years after a continuous point source began contaminating the aquifer system. The monitoring wells can sample from 1 to 3 locations along their vertical axis and have a minimum spacing of 60-m between wells in the horizontal plane.

The site is assumed to be undergoing long-term monitoring, in which groundwater samples are used to assess the effectiveness of current remediation strategies. During this long-term monitoring phase of a remediation, sampling and laboratory analysis can be a controlling factor in the costs of remediating a site. Quarterly sampling of the entire network has a potential cost of over $70,000 annually for PCE testing alone, which could translate into millions of dollars if the site had a typical life span of 20 to 30 years. To reduce these costs, the application seeks to identify *redundant* sampling locations (i.e., points that minimally affect interpolation-based plume estimates when not sampled). Specifically, the application seeks to quantify the tradeoff between LTM sampling costs and the accuracy of the plume interpolation estimates. These objectives are discussed in more detail in Section 4.2.2, below.

4.2.2 Problem Formulation

Equation (4.1) gives the multiobjective problem formulation for quantifying the tradeoff between sampling costs and maintenance of a high quality interpolated picture of the plume.

$$Minimize\ F(\bar{x}_\kappa) = \left[f_1(\bar{x}_\kappa),\ f_2(\bar{x}_\kappa)\right], \quad \forall \kappa \in \Omega$$

$$f_1(\bar{x}_\kappa) = \sum_{i=1}^{nwell} C_S(i) x_{\kappa i}$$

$$f_2(\bar{x}_\kappa) = \sum_{j=1}^{n_{est}} \left(c^*_{all}(\bar{u}_j) - c^\kappa_{est}(\bar{u}_j)\right)^2 \tag{4.1}$$

$F(\bar{x}_\kappa)$ is a vector valued objective function whose components $[f_1(\bar{x}_\kappa), f_2(\bar{x}_\kappa)]$ represent the cost and squared relative estimation error (SREE), respectively, for the κ^{th} monitoring scheme $\bar{x}_\kappa$ taken from the collection of all possible sampling designs Ω. Equation (4.2) defines the binary decision variables representing the κ^{th} monitoring scheme.

$$x_{\kappa i} = \begin{cases} 1, & if\ the\ i^{th}\ well\ is\ sampled \\ 0, & otherwise \end{cases}, \quad \forall \kappa, i \tag{4.2}$$

If the i^{th} well is sampled it is assumed that all available locations along the vertical axis of that well will be sampled at a cost of $C_S(i)$. $C_S(i)$ ranged from \$365 to \$1095 for 1 to 3 samples analyzed for PCE solely (*Rast* 1997). Sampling all available levels within each well reduces the size of Ω from 2^{50} to 2^{20}, where 50 and 20 represent the total number of sampling locations and monitoring wells (*nwell*), respectively. Reducing the size of Ω enabled the entire decision space of this application to be enumerated. Enumeration was employed to identify the true tradeoff (or Pareto frontier) between cost and SREE, shown in Figure (4.1), to aid in assessing the performance of the NSGA-II design methodology presented in this chapter.

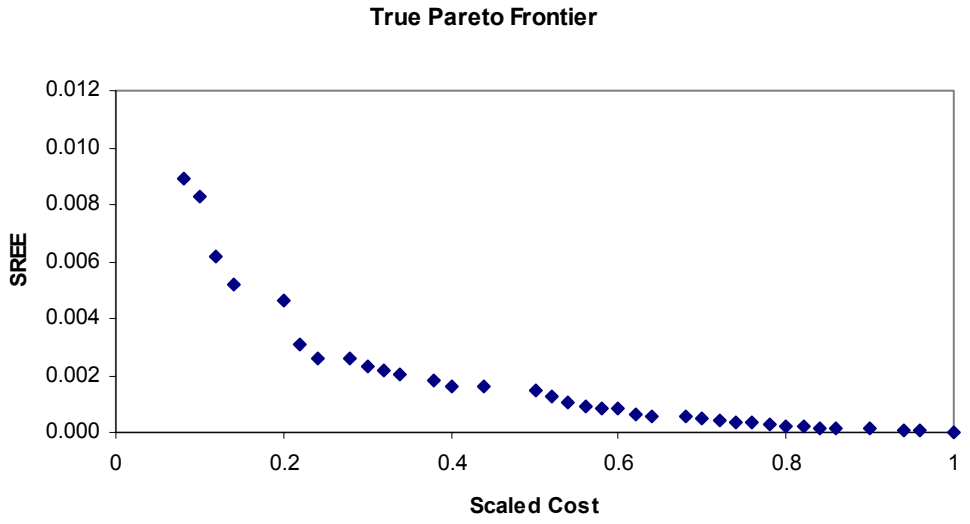


Figure 4.1 The enumerated tradeoff between Cost and SREE

The SREE provides a measure of how the interpolated picture of the plume using data only from wells included in the κ^{th} sampling plan compares to the result attained using data from all available sampling locations. The measure is computed by summing the squared deviations between the PCE estimates using data from all available sampling locations, $c^{*}_{all}(\overline{u}_j)$, and the estimates based on the κ^{th} sampling plan $c^{\kappa}_{est}(\overline{u}_j)$ at each location $\overline{u}_j$ in the interpolation domain. Each $\overline{u}_j$ specifies the coordinates for the j^{th} grid point in the interpolation domain. The interpolation domain consisted of a total of 3300 grid points (n_{est} in equation (4.1)). Nonlinear least squares inverse distance weighting was used to interpolate plume concentrations in this chapter (for more details see Chapter 3 or Chapter 5).

4.3 The NSGA-II Basics

Basic principals and operators used by the NSGA-II in quantifying design tradeoffs are discussed in this section. Genetic algorithms search a decision space using a process that is analogous to Darwin's "natural selection". The decision variables associated with the

optimization model to be solved are encoded as 0-1 binary strings of length l (termed chromosomes). The fitness of each member of a randomly generated initial population of these strings is determined by how well the design satisfies the objectives of an application. After each individual is assigned a fitness value, GAs find optimal solutions using three basic operators: (1) selection, (2) crossover (mating), and (3) mutation.

The NSGA-II uses a two-step selection process, which combines both binary tournament selection and $(\mu + \lambda)$ selection. First, binary tournament selection allows only the fittest individual from two strings randomly drawn from the current population to be placed into the mating population. Next, the crossover operator couples members of the mating population to mate. Mating was done using uniform crossover in which the strings exchange each of their component bit values with a specified probability of crossover (P_c). Tournament selection and crossover are repeated until a population of N children has been produced. Mutation then randomly flips binary bits from 1 to 0 or vice versa within the new child population with a given probability of mutation (P_m). Lastly, the NSGA-II uses $(\mu + \lambda)$ selection to choose which of the parent and child designs will survive. In this selection scheme, the populations of N parent designs and N child solutions in the current generation t are combined to yield a selection pool of $2N$ individuals, from which the N best individuals are allowed to pass to generation $t+1$. This selection method aids the algorithm in quantifying high order Pareto surfaces because it is implicitly elitist (i.e., the best designs are guaranteed to survive into the next generation).

These three operators act to create a new population (or generation) of individual sampling plans with improved average fitness. The Schema Theorem is the general theory describing how these three operators combine to evolve high quality near-optimal solutions [see *Goldberg*, 1989 or *Holland*, 1975 for more information]. It states that highly fit strings are

composed of small chunks of information (or building blocks) that are relevant to the solution of the problem. The GA exerts a *selection pressure* where only highly fit members are allowed to pass their traits or building blocks to the next generation. The phrase "selection pressure" refers to the probability of survival for individuals in a given population. For example, if population members are picked at random without being compared with one another in terms of their fitness values than each population member has an equal probability of surviving and the system has low selection pressure. Alternatively in a system with high selection pressure, a population member must compete with several other population members and only the fittest design has a high probability of survival. Highly fit parent strings are allowed to mate, yielding offspring that inherit building blocks from both parents. It is in this manner that the GA assembles optimal or near-optimal solutions to a problem.

The primary difference between the NSGA-II and single objective GAs is in how fitness is assigned. The NSGA-II evaluates sampling designs in terms of a vector of objectives. A sampling design cannot be assessed in terms of its performance in any single objective because it may perform poorly with respect to the remaining objectives. The NSGA-II employs the concepts of *Pareto dominance* and *crowding* to assign fitness values to sampling designs in the two steps described below (*Deb et al.* 2000).

The first step in fitness assignment employs the concept of *Pareto dominance* in which a design $\bar{x}$ dominates another design $\bar{x}'$ if and only if it performs as well as $\bar{x}'$ in all n_{obj} objectives and better in at least one. In the NSGA-II's improved nondomination sorting, the algorithm first ranks each design by the number of designs that dominate it. The second step in fitness assignments utilizes the concept of *crowding* (for more details see *Deb et al.* 2000) in which the average distance between an individual design and those designs nearest to it within

the population (in terms of their objectives' values). The crowding distance represents "…the largest cuboid enclosing the [κ^{th} design] without including any other point in the population" (*Deb et al.* 2000). The crowding operator eliminates the sharing parameter calculations that where required for the original form of the algorithm.

After these two steps, the fitness value of an individual design is assigned either its rank or its crowding distance. In cases were two designs have different ranks, the individual with the lower rank is preferred (i.e., individuals that are dominated by fewer solutions). Alternatively, if both solutions possess the same rank then the individual with larger crowding distance is preferred (i.e., the individual that adds the most diversity to the population). A diverse population ensures that the NSGA-II will find solutions along the full extent of the Pareto surface.

4.4 Simplifying Multiobjective Search & Optimization

The NSGA-II design methodology presented in this chapter combines concepts from previous GA design methodologies (*Reed et al.* 2000b, *Reed et al.* 2001) and the "parameter-less GA" methodology presented by *Lobo* (2000). The NSGA-II requires the specification of the four following parameters: (1) population size, (2) probability of crossover, (3) probability of mutation, and (4) run length. This set of 4 parameters represents an immediate simplification relative to the NSGA design methodology presented in Chapter 3, which also required the user to set a niche radius, an elite radius, and the appropriate selection pressure. In Chapter 3, the NSGA design methodology required a total of 10 runs and substantial user interaction to ensure the algorithm was able to capture a 2-dimensional tradeoff between costs and SREE. The NSGA-II design methodology presented in this chapter reduces the complexity of solving

multiobjective problems by introducing 3-step approach that can be used to completely automate parameter selection, minimize user interaction, and substantially reduce computational costs.

Step 1 consists of a preliminary analysis where the user sets bounds for the population size and the total time of computation. Step 2 utilizes control map theory for genetic algorithms (see *Thierens* 1995) to automatically set the probabilities of crossover and mutation. Finally, in Step 3 the algorithm automatically increases population sizes in successive runs until either *offline analysis* shows the NSGA-II sufficiently quantified the nondominated set (i.e., the tradeoff surface) or the user-specified maximum time of computation has been reached. *Offline analysis* keeps track of the best collection of nondominated individuals from all of the designs evaluated in successive runs. Each of these steps is described in more detail in Sections 4.4.2-4.4.4 below.

4.4.1 Initial Considerations

The design methodology assumes that computationally intensive fitness functions for water resources applications preclude identifying parameter settings for a distribution of initial random number seeds and instead focuses on finding optimal parameter settings for a single random number seed. Additionally, the method assumes that the user has successfully formulated their problem such that the NSGA-II will converge to a feasible nondominated set.

4.4.2 Step1: Preliminary Problem Analysis

The first step in this design methodology requires users to answer the four following questions:

- What is your initial goal for the number of nondominated solutions, R_{ND} that is acceptable?
- What is the minimum percentage change in the number of nondominated solutions, Δ_{ND}, for two successive runs to be considered identical?

- What is the average time required to evaluate a design, T_{eval}?
- What is the maximum acceptable run time, T_{tot}?

In the first question, R_{ND} defines the user's preliminary goal for the number of nondominated solutions she or he would want to obtain on the tradeoff surface. R_{ND} is then used to calculate a lower bound population size N_0 for the NSGA-II using the empirical rule-of-thumb presented in equation (4.3).

$$N_0 \approx 2R_{ND} \tag{4.3}$$

Users should not concern themselves with exactly specifying R_{ND} because it is generally impossible to know a priori the exact number of Pareto optimal solutions that exist for a given application.

Since it is impossible to know a priori the exact number of Pareto optimal solutions that exist, the user must consider problem specific information to set this value. For the LTM application presented in this chapter, the NSGA-II is seeking to find the minimum SREE value for each level of cost. Theoretically, the nondominated set can be composed of no more than 58 solutions because the cost objective is a discrete, linear function of the number of sampling locations (which ranges between 1 and 58). R_{ND} was set equal to 30, which is approximately 50 percent of the theoretical maximum number of nondominated solutions that could exist. R_{ND} was set to a relatively small number of nondominated solutions because only in rare cases will the nondominated set be composed of the theoretical maximum number of individuals. Figure (4.1) demonstrates this fact, showing that only 36 nondominated solutions exist for the LTM application because some of the designs at lower cost levels dominate higher cost designs (i.e., they have both a lower cost and a lower SREE value). Setting R_{ND} to a reasonably small value

has the additional advantage of ensuring that the NSGA-II can be tested for small population sizes, which can significantly reduce the total number of designs that must be evaluated if the algorithm successfully quantifies the tradeoff surface using these reduced population sizes. Moreover, the successive increases in population size described in Step 3 of this methodology will correct for an undersized population automatically.

The second question directly addresses the issue of *solution exactness*, by requiring the user to specify the minimum percentage change in the number of nondominated individuals Δ_{ND} for two successive runs to be considered identical. This parameter is best explained using an illustrative example. Consider two successive runs of the NSGA-II in which the first run uses a population of N designs to evolve a nondominated set composed of A individuals while the second run uses a population of $2N$ designs to evolve a nondominated set of K individuals. The results of these runs are used in Equation (4.4) to define which of the two following courses of action will be taken: (1) population size is again doubled, resulting in $4N$ individuals to be used in an additional run of the NSGA-II or (2) the algorithm stops because the nondominated set has been quantified to sufficient accuracy.

$$\begin{aligned} &\textit{if } \Delta_{ND} < \left(\frac{|K - A|}{A} \right) 100 \textit{ then double } N \textit{ and continue search} \\ &\textit{else stop search} \end{aligned} \tag{4.4}$$

In equation (4.4), if the run using a population size of $2N$ does not increase the size of the nondominated set by more than Δ_{ND}-percent then the search has reached a point of diminishing return where further population size increases are no longer justified. Equation (4.4) is relevant to both discrete and continuous problems because in both cases the NSGA-II is building a discrete approximation to the true Pareto surface. Δ_{ND} was set to 10 percent for the LTM application presented in this chapter, implying that a close approximation of the true Pareto front

shown in Figure (4.1) is being sought. Setting Δ_{ND} directly addresses the tradeoff between *solution exactness* and the *computational complexity* of an application. By setting Δ_{ND} to any value above zero, the user is accepting a less exact representation of the nondominated set to achieve a decrease in computing time.

The third and fourth questions require the user to specify how long he or she is willing to wait for a solution (i.e., T_{tot}), which allows the upper bound population size to be computed given how long each function evaluation takes on average, T_{eval}. Note that any timing variables presented in this chapter with a capital *T* are in terms of clock time and those presented in lower case *t* are in terms of the number of generations the NSGA-II is run. For the LTM application, T_{tot} was set equal to 4 hours to allow the NSGA-II ample opportunity to quantify the nondominated solution set given that each design took an average of 0.044 seconds to evaluate on a Dell XPS T800r running Windows 2000. The maximum number of designs, E_{tot}, that can be evaluated given both T_{eval} and T_{tot} can be computed using equation (4.5) and had a value of 327,000 for the LTM application.

$$E_{tot} = \frac{T_{tot}}{T_{eval}} \tag{4.5}$$

The results of equation (4.5) are used in equation (4.6) to compute the maximum number of times, β_{MAX}, the population size can be doubled beyond its lower bound value N_0 computed using equation (4.3).

$$\sum_{\beta=0}^{\beta_{MAX}} 2^{\beta} N_0 \, t \le E_{tot} \tag{4.6}$$

In equation (4.6), t represents the total number of the generations used in each run of the NSGA-II to search for the nondominated set. The run length t is estimated to be approximately equal to $2l$, where l is the length of the binary strings that represent designs, as was recommended by *Reed et al.* (2000b). The LTM designs had a string length l = 20 yielding a run time of 40 generations. Conservatively, this estimate assumes that the combination of $(\mu + \lambda)$ and binary tournament selection in the NSGA-II converges as fast as a system undergoing pure binary tournament selection (for details see *Thierens et al.* 1998). The quantity $N_0 t$ represents the total number of designs that are evaluated when the lower bound population size is used for t generations and had a value of 2400 for the LTM application. Given the maximum number of designs that can be evaluated (E_{tot} = 327,000), equation (4.6) shows that N_0 can be doubled a total of six times (i.e., β_{MAX} = 6), yielding an upper bound population size of 3840.

4.4.3 Step 2: Balancing Innovation and Disruption

Step 1 of the methodology requires users to supply information for performing population sizing and specifying run length, which are two of the four parameters required for the NSGA-II. This step explains how the two remaining parameters, the probability of crossover P_c and the probability of mutation P_m, are set in this methodology. There are two forces that must be balanced when setting these parameters' values: (1) disruption and (2) innovation. Disruption occurs when excessive crossover or mutation "disrupts" pertinent building blocks within solutions, which makes it impossible for GAs to assemble optimal solutions. Equation (4.7) presents the disruption boundary relationship developed in *Thierens* (1995) and used in the simple GA design methodology of *Reed et al.* (2000b).

$$P_c \leq \frac{s-1}{s} \tag{4.7}$$

The parameter s represents the total number of individuals that compete in tournament selection. Equation (4.7) is intended to protect pertinent building blocks from being destroyed due to excessive crossover. Since the child populations in the NSGA-II are selected using binary tournaments (i.e., $s = 2$), the probability of crossover must be set less than or equal to 0.5 or 50 percent. As the number of members that compete to survive [i.e., tournament size s in equation (4.7)] increases, selection pressure also increases because it becomes increasingly more difficult for unfit population members to survive. Equation (4.7) shows that the disruption boundary is less restrictive (i.e., higher P_c values can be used) when selection pressure is increased. The disruption boundary value of $P_c \leq 50$ percent attained from equation (4.7) is conservative for the NSGA-II because the algorithm is guaranteed to always have higher selection pressure relative to a system undergoing pure binary tournament selection.

The potential disruptive effect of crossover must be balanced with the positive effect of this operator. *Thierens and Goldberg* (1993) show that selection and crossover combine to exert an innovative force on a system that guides GAs towards promising solutions. This force of innovation should be maximized by using the largest P_c possible below the upper bound defined by equation (4.7), which predicts solution instability for P_c near and above 50 percent. Because the methodology presented in this chapter is self-adaptive, if $P_c = 0.5$ is initially too low for the NSGA-II to successfully mix or discover promising building blocks, then doubling the population size in a successive run would overcome this problem because both the population's diversity and the expected number of crossovers that the algorithm can use to successfully assemble optimal solutions would double. Doubling the population size from N to $2N$ also increases the size of the selection pool from $2N$ to $4N$, greatly increasing selection pressure

(designs must compete with far more individuals to survive) and decreasing the probability of disruption due to crossover and mutation.

The probability of mutation P_m is set using equation (4.8) based on the recommendations of the empirical studies of *De Jong* (1975) and *Schaffer et al.* (1989).

$$P_m \approx \frac{1}{N} \tag{4.8}$$

Selection and mutation combine to locally refine promising solutions. Equation (4.8) ensures that mutation is minimally disruptive.

4.4.4 Step 3: Multi-Population Trial Runs

Steps 1 and 2 of this design methodology provide all of the parameters required by the NSGA-II to perform trial runs. In a manner analogous to *Lobo* (2000) and *Reed et al.* (2000b), the NSGA-II automatically initiates trial runs for successively doubled population sizes. These trials runs use offline performance analysis to *sufficiently* quantify tradeoffs. From Step 1, tradeoffs are defined to be sufficiently quantified when the minimum percentage change in the number of individuals in the nondominated set for successive runs is less than 10 percent, defining the point of diminishing returns when successive increases in population size are no longer justified.

The design methodology presented in this chapter requires a minimum of two runs using two successively doubled population sizes to determine if the nondominated set has been sufficiently captured (i.e., $\Delta_{ND} \leq 10$ percent for this case). Figure (4.2a) shows the NSGA-II's performance for Run 1 using the lower bound population size of 60 members running for 40 generations. The figure shows that the NSGA-II is able to closely approximate 19 members of the nondominated set, most of which are "compromise solutions" in the center of the Cost—

SREE tradeoff. Figure (4.2a) illustrates the importance of allowing the NSGA-II to search for the nondominated set with small population sizes. Often practitioners are primarily focused on "compromise solutions" in the central region of tradeoffs for water resources applications because extreme solutions have a low likelihood of being implemented (although extreme solutions can provide insight into how individual objectives are affecting designs).

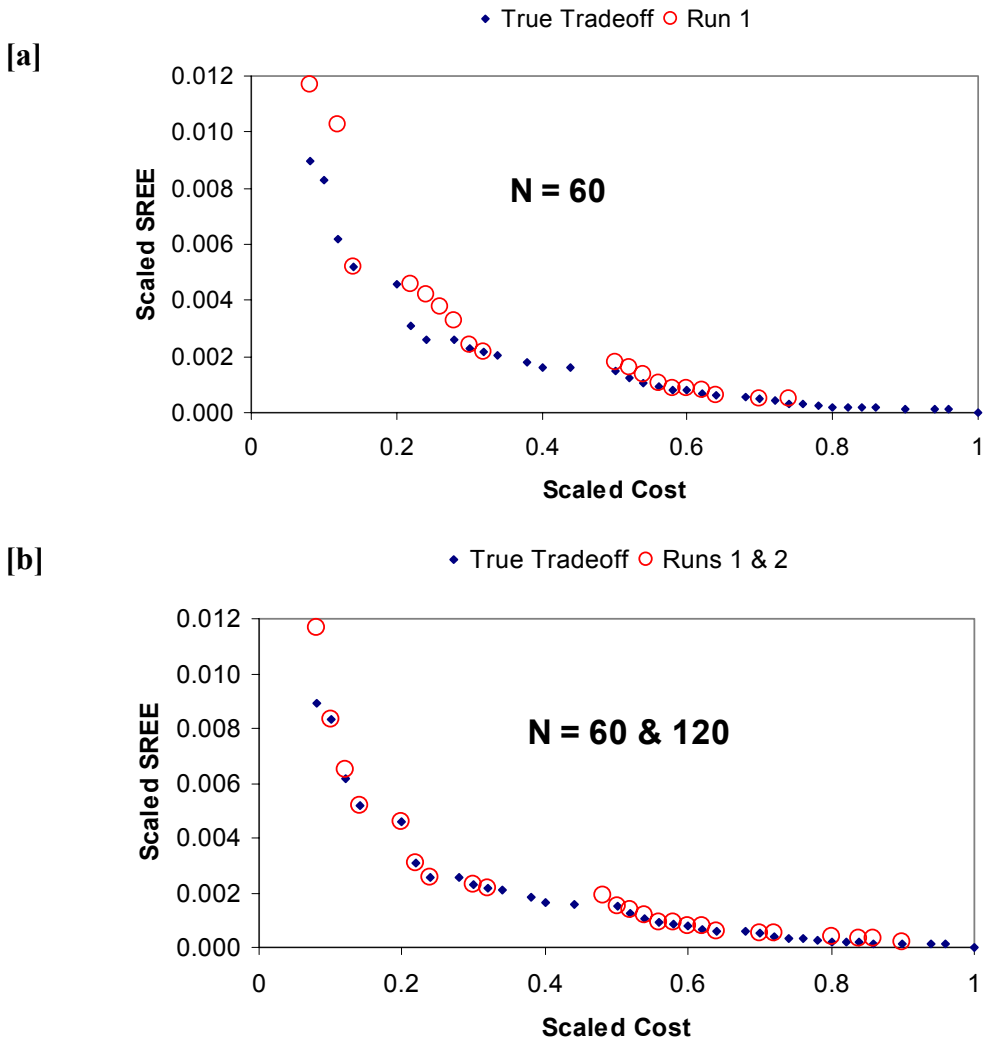
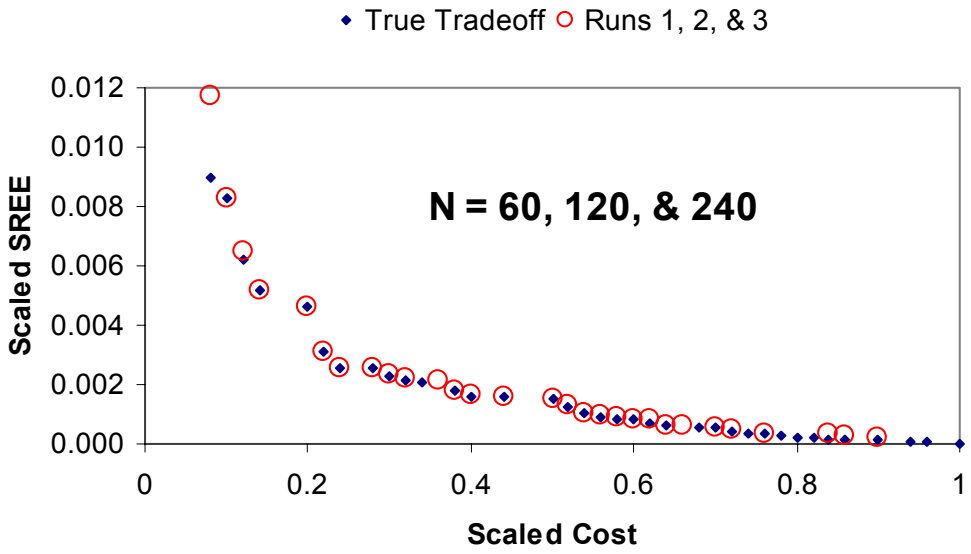


Figure 4.2 (a) Offline results for Run 1 (b) Combined offline results for Runs 1 & 2

Figure (4.2b) shows the combined offline results of Runs 1 and 2, in which all of the nondominated solutions found using $N = 60$ are combined with those found using $N = 120$ to

form an improved approximation of the Cost—SREE tradeoff. Combined offline analysis for multiple runs takes full advantage of every function evaluation required to solve an application and guarantees that the NSGA-II's performance will only improve for successive runs. In Figure (4.2b), the nondominated set grew by 5 members relative to Run 1 shown in Figure (4.2a), a 26 percent increase.

[a]



[b]

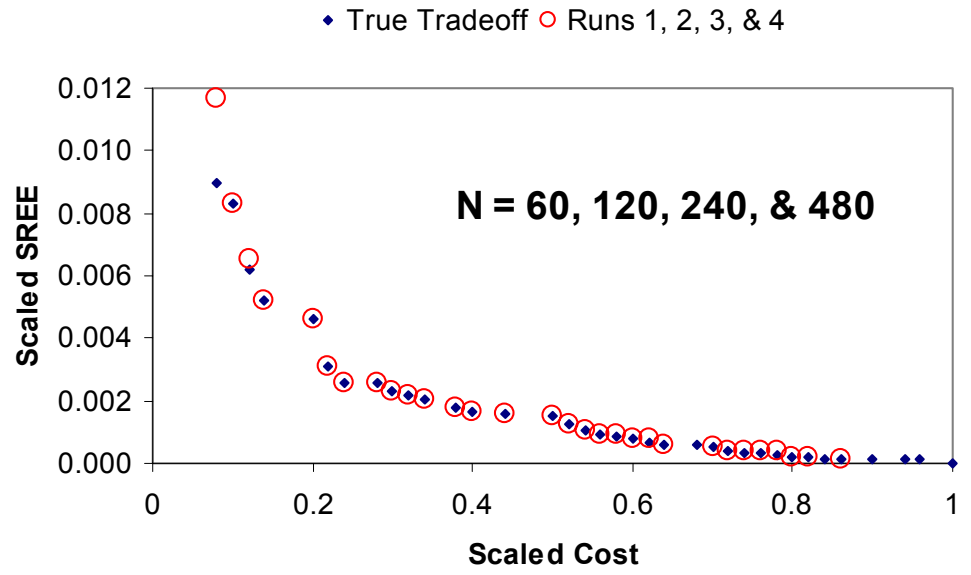


Figure 4.3 (a) Combined offline results for Runs 1, 2, & 3 (b) Combined offline results for Runs 1, 2, 3, & 4

Figure (4.2b) shows that the NSGA-II is more closely approximating solutions along the full extent of the Cost—SREE tradeoff; a third run is then initiated with a population size N = 240 because Δ_{ND} is less than 26 percent [see equation (4.4)]. Figure (4.3a) shows the combined offline results of Runs 1, 2, and 3 in which the NSGA-II identified 29 nondominated solutions. The figure shows that the NSGA-II has nearly captured the Pareto optimal set of solutions, but the tradeoff still has not been proven to be sufficiently quantified because the nondominated set increased by 20 percent [i.e., Δ_{ND} < 20 percent in equation (4.4)]. An additional run using a population size of N = 480 increased the nondominated set by only 1 solution, representing a 3 percent change. It is clear from both plots in Figure (4.3) that the point of diminishing return has been reached when subsequent increases in population size are no longer justified. The only task that remains for the user is to inspect the quality of the nondominated set quantified by the NSGA-II to ensure her or his expectations have been met successfully. If the NSGA-II does not satisfy the user's expectations then a more exact approximation of the nondominated set can be sought by decreasing Δ_{ND} and continuing to increase population size within the user-specified computational limits.

A significant advantage of this design methodology is that it requires only minimal initial user interaction, with successive runs completed automatically. Additionally, the design methodology of this chapter exploits the efficiency of the NSGA-II to substantially reduce computational costs. Consider that the NSGA design methodology of Chapter 3 required more than 160,000 function evaluations to meet the performance of the NSGA-II shown in Figure (4.3) for the same LTM application. The 4 trial runs discussed in this chapter required a total of 38,000 function evaluations, representing an 80 percent reduction in the computational costs relative to the previous NSGA design methodology.

4.5 Conclusions

The NSGA-II design methodology presented in this chapter builds on previous GA design methodologies (*Reed et al.* 2000, *Lobo* 2000, and *Reed et al.* 2001) to introduce a multi-population approach that automates parameter specification for the algorithm. The design methodology simplifies Pareto optimization using the NSGA-II into a simple 3 step process. The first step only requires users to answer 4 questions on their performance expectations and run time. Step 2 utilizes GA control map theory to automatically set the probabilities of crossover and mutation. The final step of the design methodology successively doubles population sizes until the NSGA-II has sufficiently identified the nondominated set or acceptable run time is exceeded. The methodology successfully solved the LTM design application using 80 percent fewer function evaluations than the NSGA design methodology presented in Chapter 3. The combined efficiency of the NSGA-II and the design methodology presented in this chapter allows more challenging higher order Pareto optimization problems (i.e., problems with more than 2 objectives) to be solved [see Chapter 6].