Hindawi
Complexity
Volume 2018, Article ID 7242105, 18 pages
https://doi.org/10.1155/2018/7242105

Research Article

Using Multiple Criteria Optimization and Two-Stage Genetic Algorithms to Select a Population Management Strategy with Optimized Reliability

Jessica L. Chapman,[1] **Lu Lu**,[2] **and Christine M. Anderson-Cook**[3]

[1] *Department of Math, Computer Science, and Statistics, St. Lawrence University, Canton, NY 13617, USA*
[2] *Department of Mathematics & Statistics, University of South Florida, Tampa, FL 33620, USA*
[3] *Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 88544, USA*

Correspondence should be addressed to Lu Lu; lulu1@usf.edu

Received 23 March 2018; Accepted 14 October 2018; Published 13 November 2018

Academic Editor: Dan Selişteanu

An important aspect of good management of inventory for many single-use populations or stockpiles is to develop an informed consumption strategy to use a collection of single-use units, with varied reliability as a function of age, during scheduled operations. We present a two-phase approach to balance multiple objectives for a consumption strategy to ensure good performance on the average reliability, consistency of unit reliability over time, and least uncertainty of the reliability estimates. In the first phase, a representative subset of units is selected to explore the impact of using units at different time points on reliability performance and to identify beneficial consumption patterns using a nondominated sorting genetic algorithm based on multiple objectives. In the second phase, the results from the first phase are projected back to the full stockpile as a starting point for determining best consumption strategies that emphasize the priorities of the manager. The method can be generalized to other criteria of interest and management optimization strategies. The method is illustrated with an example that shares characteristics with some munition stockpiles and demonstrates the substantial advantages of the two-phase approach on the quality of solutions and efficiency of finding them.

1. Introduction

With increasing constraints on resources and budget, there are more decision-making situations that involve balancing of multiple objectives or criteria [1–7]. Pareto frontiers [8–12] have become a popular tool to simultaneously optimize multiple objectives. This approach aims to assemble a set of contending solutions by eliminating poor choices that are strictly inferior to others. It is advantageous in two ways: first, it allows us to see an objective full set of competing choices before considering the subjective aspects of the decision; second, it effectively reduces the candidate solutions to a smaller subset so that solutions robust to subjective choices that affect the final decision can be evaluated more carefully to support a coherent defensible final decision.

In recent years, strategies [13–17] that encourage separate and transparent considerations of objective and subjective aspects of a complex decision have been developed. Particularly, the Design-Measure-Reduce-Combine-Select (DMRCS) approach [13] offers more rigor and structure to guide the often messy and complicated decision-making process. After specifying the goals of the study with their appropriate metrics in the Define and Measure steps, the Pareto front approach is used in the Reduce step to allow us to identify the set of objectively superior solutions. Then user priorities are incorporated in the Combine and Select steps to understand the trade-offs and impacts of subjective weighting and scaling choices through a suite of graphical summaries [14–17] and guide the user to make sensible and justifiable decisions that are tailored to the study goals.

In many applications, the complete enumeration and evaluation of all the possible solutions is practically infeasible. Direct Pareto optimization can also be unduly time-consuming for a large population, given the number of solutions to be evaluated, as well as the computational intensity of evaluating solutions when some complex criteria are calculated. Thus, a numerical search algorithm is often needed to generate desirable solutions in a timely manner. Genetic algorithms (GAs), also called evolutionary algorithm (EAs) [18–20], are popular methods for finding the Pareto front based on multiple objectives. GAs are a family of heuristic optimization techniques that mimic the evolutionary process for optimizing objective functions of interest. At each evolving generation, solutions in the current population and a set of new solutions generated by crossover and mutation operations compete to survive in the next generation.

There are generally two ways that a GA can be used to optimize multiple objectives simultaneously. One approach is to guide the GA search based on optimizing a combined objective function using the desirability function approach [21]. The Pareto front can be populated along the GA search process. The drawback to such an approach is that it generally relies on subjective choices made by the user before the search, such as the scaling of the criteria and the relative weights or priority given to each objective. To mitigate this issue, multiple GAs can be run using different weighting and scaling choices to maximally populate the front and avoid local optimization based subjective user choices [17]. However, sometimes this comes with an unnecessarily increase in run time because the multiple parallel GA searches can result in the repeated evaluation of the same solution many times.

Another approach is to use a class of evolutionary algorithms specifically designed to simultaneously optimize multiple criteria to identify the Pareto front. At the end of each generation, the best available set of solutions that simultaneously optimize different prioritizations of the criteria are obtained. One such algorithm is the fast and elitist nondominated sorting genetic algorithm (NSGA-II) [22]. NSGA-II is an improvement upon the previous nondominated sorting algorithm, NSGA, which was one of the first evolutionary algorithms for simultaneously optimizing multiple objectives [23]. The major criticisms of NSGA, which have all been addressed by NSGA-II, were the computational complexity of the nondominated sorting algorithm, its lack of elitism, and the need for users to specify additional parameters [22]. The two key elements of NSGA-II are that (1) it sorts the population of solutions into *tiers* of nondominated solutions and (2) within a tier, solutions are ranked according to their *crowding distance* to promote a diverse set of solutions across all portions of the PF [22].

However, some of the computational challenges for populating a rich Pareto front relate to the potentially large solution space and/or the large number of objectives. This has substantially influenced its application to even broader areas of problems. In this paper, we propose a new two-phase optimization process to accelerate the search for a Pareto front over a large solution space. In the first phase, a representative subpopulation is selected to explore the general features of the set of superior solutions. In the second phase, the main features of the more promising solutions found in Phase I are matched back to the original population to form more informative starting points for an accelerated search for the full Pareto front in the full population. The new method is illustrated with a stockpile management example, which aims to management a collection of single-use units to ensure good performance on several reliability characteristics including average reliability, consistency of unit reliability over time, and uncertainty of the reliability estimates. The method demonstrates improvements on both the quality of solutions and computational efficiency of the search algorithm. The NSGA-II algorithm is adapted for this stockpile management application by optimizing several reliability-based characteristics, and it is further enhanced by using an adaptive population size [24] to allow more efficient population of the rich Pareto front.

The remainder of the paper is organized as follows. Section 2 provides more details about the problem, data, and the basic statistical modelling and data analysis for the stockpile management application. Section 3 details the methods for the general genetic search algorithm and its two-phase implementation for enhancing computational efficiency. Section 4 demonstrates the application of the method to the stockpile example. Section 5 offers some conclusions and more insights on implementation for other applications.

2. Application

For many stockpiles or populations of single-use units whose reliability degrade with age, the characteristics of the different individual units vary as units were added to the stockpile at different times and their reliability when used differ as units are consumed at different times. Hence, deciding what order to use the units can make an important difference to their overall performance and impact the utility of the stockpile. We consider a scenario where $N = 200$ units with different ages are available in the stockpile and have different estimated reliability. While the original Department of Defense data cannot be shared for proprietary reasons, this example shares characteristics with many stockpile management scenarios. The data sets for this example are included in Tables 1 and 2. The units are planned to be used in scheduled operations at the end of each year for four years at a rate of 50 units per year, and the goal is to find a best strategy to use all of the units to achieve good performance on several aspects of reliability. For the practitioners, obtaining a timely solution with solid performance on several metrics of interest represents a substantial improvement over current practice where haphazard samples are used with unpredictable reliability performance.

To estimate the reliability, a sample of 227 units (summarized in Table 1) from the same stockpile as the 200 units (shown in Table 2) were previously destructively tested with a pass or failure observed for each tested unit. The units were at different ages when they were tested. The reliability was estimated by fitting a Probit regression model [25] where the binary response (pass or failure) is assumed to follow a Bernoulli distribution, $Y_i \sim Binomial(1, R_i)$, and R_i is the

Table 1: The data table for the 227 sample units used for estimating the reliability. The listed values are the age of the unit at the testing time (measured in months) with the 34 units that failed bolded and listed in the first two columns. The remaining unbolded entries are the ages for units that passed.

Age													
0	**66.7**	0	0	0	0	0	0	0	4.9	40.7	65.5	68	78.8
0	**67.4**	0	0	0	0	0	0	0	4.9	43.6	65.7	68	81.8
0	**67.5**	0	0	0	0	0	0	0	4.9	47.3	66.1	68	83.6
0	**67.7**	0	0	0	0	0	0	0	14.5	47.9	66.1	71.3	83.7
0	**67.8**	0	0	0	0	0	0	0	15.3	48	66.3	71.7	83.8
0	**68**	0	0	0	0	0	0	0	18.2	48.4	66.7	71.8	89.5
0	**71.5**	0	0	0	0	0	0	2.1	19.3	50.1	66.9	72.7	
4.8	**71.7**	0	0	0	0	0	0	2.1	22.9	50.4	67	72.9	
11.7	**73.1**	0	0	0	0	0	0	2.4	23.8	50.5	67	73	
31.6	**75.7**	0	0	0	0	0	0	2.4	26.9	52.4	67.3	73.1	
31.6	**75.9**	0	0	0	0	0	0	2.4	29.5	54.4	67.4	73.1	
36.5	**77.6**	0	0	0	0	0	0	2.4	29.6	55.1	67.5	73.1	
47.9	**78.3**	0	0	0	0	0	0	2.5	31.6	65	67.5	73.8	
55.1	**78.6**	0	0	0	0	0	0	2.7	32.3	65.1	67.5	74.1	
59.5	**81.9**	0	0	0	0	0	0	3.2	32.3	65.1	67.7	77.6	
64.9	**82.1**	0	0	0	0	0	0	3.2	33.6	65.2	67.8	77.7	
65	**87.8**	0	0	0	0	0	0	4.7	36.5	65.5	67.9	78.6	

Table 2: The ages of the 200 units from the current population to be consumed.

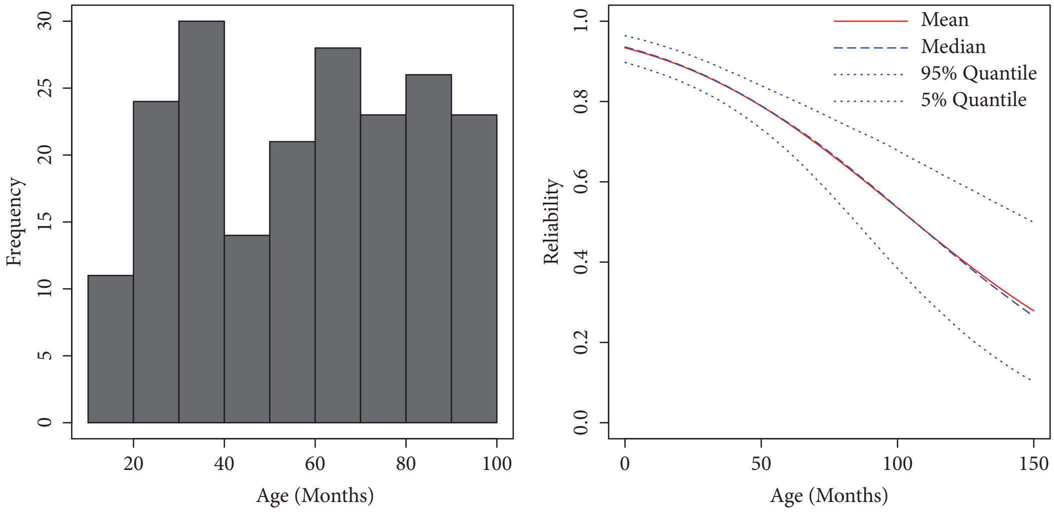
24.6	68.4	44.8	82.5	47.9	20.7	51.0	86.5	32.6	51.3
56.9	54.6	70.9	54.4	53.0	67.3	66.2	91.5	97.1	18.1
87.7	60.8	27.9	74.9	32.5	64.4	79.3	66.3	38.0	38.5
88.6	85.5	42.4	22.4	39.3	30.0	97.3	53.8	67.1	72.9
46.0	50.0	80.5	53.7	68.0	52.9	80.3	66.1	98.7	20.5
78.8	36.0	76.6	89.1	54.7	30.4	16.6	25.9	87.2	80.2
68.1	19.0	49.4	61.4	28.6	67.2	69.1	20.3	37.7	92.5
35.1	58.8	19.4	84.9	30.9	88.7	39.9	72.1	81.5	91.9
80.4	35.5	46.2	29.1	30.3	88.7	90.8	52.6	19.8	74.7
88.3	76.6	77.0	23.9	69.5	59.0	67.1	60.2	95.2	57.1
90.3	57.7	68.7	29.4	49.7	83.4	62.9	42.7	61.4	20.8
20.6	67.3	78.6	19.4	61.5	18.9	26.5	69.7	90.9	80.4
71.5	92.4	93.5	82.9	17.8	63.4	45.2	39.1	83.4	49.5
52.2	38.2	56.5	86.1	60.0	64.7	34.8	45.4	25.7	58.2
31.2	29.3	19.0	92.1	69.6	94.1	67.8	98.0	21.3	39.1
32.7	39.1	73.6	88.9	98.8	48.8	71.8	23.2	96.3	74.0
95.9	73.8	96.2	37.4	84.8	68.5	73.4	59.3	73.6	73.7
93.1	15.2	74.1	87.8	77.7	31.3	32.7	32.2	49.3	36.5
60.0	22.1	95.6	31.9	91.2	23.4	86.8	74.5	20.8	83.5
24.0	18.7	99.5	70.4	39.5	23.9	30.1	39.7	53.4	28.1

probability of observing a pass, which is dependent on the age of the unit (denoted by A_i) as given by

$$R_i = \Phi\left(\beta_0 + \beta_1 A_i\right). \quad (1)$$

The Φ in Eq. (1) is the standard normal cumulative distribution function, and β_0, β_1 are the unknown coefficient parameters. Bayesian analysis [26, 27] was used to estimate the model parameters with diffuse prior distributions, $\beta_0, \beta_1 \sim \mathit{Uniform}(-1000, 1000)$, assumed for the coefficient parameters. Markov Chain Monte Carlo (MCMC) simulation was used to approximate the posterior distribution of the model parameters given the observed data. The reliability of any unit from the stockpile can be predicted from Eq. (1) using the age of the unit at its use time.

Figure 1 shows the histogram for the current age distribution for the 200 units in the stockpile, as well as the estimated reliability of units across the range of anticipated ages when they will be consumed. From Figure 1(a), we can see that



(a) (b)

Figure 1: (a) Current age distribution for the stockpile of 200 units to be used; (b) Summary of posterior reliability distribution as a function of the age of the unit.

the current age of the units in the stockpile ranges from 15 to 100 months, and the age distribution is relatively close to a uniform distribution. The estimated reliability curve in Figure 1(b) shows the posterior mean, median, and 5% and 95% quantiles of the posterior distribution of the reliability as a function of the age of the unit. From the plot, we can obtain the estimated reliability for any unit in the current stockpile when it is consumed. For example, if a unit is currently 20 months old, then a year from now the unit will be 32 months old and its reliability is predicted to be between 0.8 and 0.9 with a 90% probability.

Given the current age distribution and the reliability curve as a function of age, we can calculate the age of the units at their scheduled use time and then predict the reliability of the units and its uncertainty for any consumption strategy (where a strategy describes the order to use the units during scheduled operations). With this information, we can then summarize the overall performance of the strategy based on some chosen metrics to evaluate and compare strategies. Three aspects of the reliability performance are of interest to the stockpile manager including the overall average reliability of all units at their use times, the consistency of performance for the collection of units used in different time intervals, and the precision of the estimated reliability. The goal is to find a best strategy to simultaneously optimize all three objectives given current understanding of the stockpile.

Matching the Define and Measure steps in the DMRCS (Define-Measure-Reduce-Compare-Select) [13] process, we convert the general goals for the optimization into quantitative metrics that are calculated and compared for possible consumption strategies. Since the reliability of the units has been estimated from previous testing, it is important to incorporate the uncertainty associated with the estimated reliability into the assessment of different strategies. Let R_{ij} denote the probability that the jth unit works properly if it is used at the end of the ith year, where $i = 1, \ldots, 4$ and $j = 1, \ldots, n_i = 50$. For a stockpile manager, the quantity of main interest is not the individual reliability R_{ij} but the success rate of all 50 units used in each time interval, which can be estimated with the average reliability of the 50 units as $R_i = (1/50)\sum_{j=1}^{50} R_{ij}$. Then the reliability characteristics of most interest can be quantified as follows. The overall reliability is calculated by the average reliability summarized over the four years, $\overline{R} = (1/4)\sum_{i=1}^{4} R_i$. The consistency over time can be measured by the standard deviation of the success rates over the four years, i.e. $SD(R_i) = \sqrt{(1/3)\sum_{i=1}^{4}(R_i - \overline{R})^2}$. The overall uncertainty of the estimated reliability can be quantified by $s(\widehat{R}) = \sum_{i=1}^{4} s(\widehat{R}_i)$, where $\widehat{R}_i = (1/50)\sum_{j=1}^{50} \widehat{R}_{ij}$ is the estimated success rate at the end of ith year, and $s(\widehat{R}_i)$ is the posterior standard deviation of $\widehat{R}_i$ calculated from the MCMC simulation. In the MCMC simulation in the Bayesian analysis, we generate $M = 3000$ draws of parameters to approximate the posterior distribution after the burn-in runs; we use a superscript "B" to denote Bayesian estimates computed using the posterior draws. Let $\widehat{R}_{ijk}^{B}$ denote the estimate of R_{ij} using the k^{th} draw ($k = 1, \ldots, M$) of the model parameters given by $\widehat{R}_{ijk}^{B} = \Phi(\widehat{\beta}_{0,k}^{B} + \widehat{\beta}_{1,k}^{B} A_{ij})$, where A_{ij} is the age of the unit j when it is used at the end of the i^{th} year and $\widehat{\beta}_{0,k}^{B}, \widehat{\beta}_{1,k}^{B}$ are the k^{th} posterior draws of the coefficient parameters. Then $\widehat{R}_{ik}^{B} = (1/50)\sum_{j=1}^{50} \widehat{R}_{ijk}^{B}$ is the estimate of the expected success rate R_i using the k^{th} posterior draw from MCMC. The posterior mean $\widehat{R}_i^B = (1/M)\sum_{k=1}^{M} \widehat{R}_{ik}^{B}$ can give a point estimate of R_i and its uncertainty can be estimated by $s(\widehat{R}_i^B) = \sqrt{(1/(M-1))\sum_{k=1}^{M}(\widehat{R}_{ik}^{B} - \widehat{R}_i^B)^2}$. The three objectives for reliability on which consumption strategies are compared can be calculated using the following metrics:

$$\text{Overall average reliability (maximize)}: \widehat{\overline{R}}^{B} = \frac{1}{4}\sum_{i=1}^{4} \widehat{R}_i^{B} \quad (2)$$

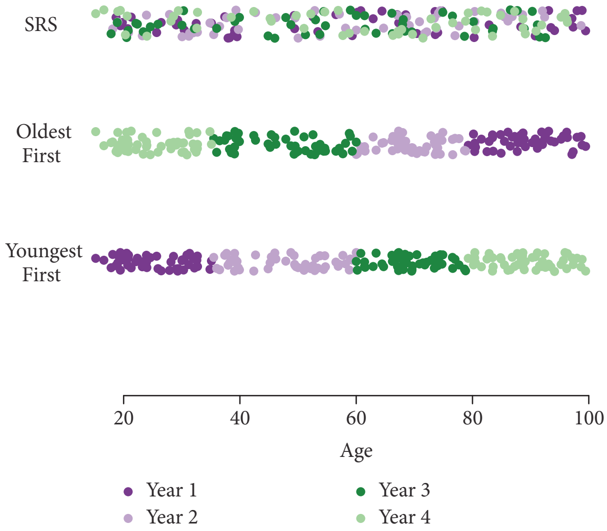


Figure 2: Typical consumption strategies: (1) Youngest first (always uses youngest units at any time until all units are consumed), (2) oldest first (always uses oldest units), and (3) SRS (selects a simple random sample to use in each time interval).

$$\text{Consistency over time (minimize)}: SD\left(\widehat{R}_i^B\right) = \sqrt{\frac{1}{3}\sum_{i=1}^{4}\left(\widehat{R}_i^B - \widehat{\overline{R}}^B\right)^2} \quad (3)$$

$$\text{Overall uncertainty (minimize)}: s\left(\widehat{R}^B\right) = \sum_{i=1}^{4} s\left(\widehat{R}_i^B\right) \quad (4)$$

The previous practice before our study was to choose the units to use haphazardly from the stockpile, which sometimes yielded disappointing and unpredictable results. To gain a quick understanding about how using units at different times affects their reliability, we examine three simple strategies that practitioners might consider. Figure 2 shows the current age distribution for three possible usage strategies, where different colors are used to display the year in which each unit it used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4). The circles are jittered vertically to better display all the units with similar ages. The three strategies considered include (1) the *youngest first* strategy, where the youngest available units in the stockpile are always used first in each time interval; (2) the *oldest first* strategy, where the oldest available units in the stockpile are used first; (3) the *simple random sampling (SRS)* strategy, which selects an unstratified random sample of units to use in each time interval, which results in the ages of units used to be in each time period to be approximately evenly spread across the age distribution. It is difficult to match any of these strategies to the haphazard strategy, since the storage of the units can induce particular patterns and may lead to results that are similar to any of the above strategies.

Figure 3 compares the reliability performance over the four years for these three strategies. For each strategy shown in a subfigure of Figure 3, the black dots are the posterior mean estimates of reliability for individual units at their use times ($\widehat{R}_{ij}^B$). The dashed line connects the posterior mean estimates of the expected success rate of the 50 units used at the end of each year, $\widehat{R}_i^B$, which is equivalent to the average reliability of the units used in each time interval. The vertical intervals (with the dashed lines) display the 90% credible interval of the expected success rates over the four years, which are bounded by the 5% and 95% quantiles of the empirical posterior distribution of $\widehat{R}_i^B$.

We can see that the youngest first strategy starts with the highest success rate with the most precision, but that decreases quickly over time with increasing uncertainty. By leaving the oldest units to be used at the end, reliability drops even faster and its uncertainty grows much faster for units older than 70 months (see the reliability curve in Figure 1(b)). The oldest first strategy starts with the lowest success rate, but this rate increases over time because the aging of the units is offset by leaving the youngest units to be used at the end. It also has more uncertainty in early years which reduces gradually over time. The SRS strategy has more variable individual reliability within each time interval, but the estimated success rate is between those of the other two strategies. The success rate decreases over time, mainly due to the natural aging of the units. The uncertainty also increases over time due to the aging of the stockpile. In comparison, the youngest first strategy has the lowest average reliability, while the oldest first strategy has the highest average reliability. Also, the oldest first strategy has the most consistent performance over the four years, while the youngest first strategy has the least consistency over time. The SRS strategy has the smallest overall uncertainty among the three strategies. Therefore, none of these strategies has universally best performance over all aspects. To select a best strategy, we need to balance the trade-offs between the multiple aspects of reliability performance to make a decision tailored to the needs of the stockpile manager.

To simultaneously optimize the multiple reliability objectives given in Equations (2)-(4), we use a Pareto front approach with the DMRCS procedure to guide the structured decision-making process. After specifying the goals of the study with their appropriate metrics in the Define and Measure steps, the Pareto front approach from the Reduce step allows us to identify the set of objectively superior solutions (consumption strategies, in our application). To find the Pareto front, a customized search algorithm is needed when there is not a complete enumerated list of all possible solutions available. Genetic algorithms are often used to search through a large solution space to find optimal solutions for multiple objective optimization problems. Particularly, the NSGA-II, is a leading search algorithm for multiple objective optimization. However, with all of the advantages of finding the complete Pareto front, the search for the front is time consuming relative to optimizing on a single overall summary built upon the multiple objectives. For even this moderate sized stockpile of 200 units, there are an enormous number of strategies to choose from $\binom{200}{50}\binom{150}{50}\binom{100}{50} = 200!/(50!)^4 \approx 10^{117}$. Given the size of the solution space, a genetic algorithm to search directly over the entire solution

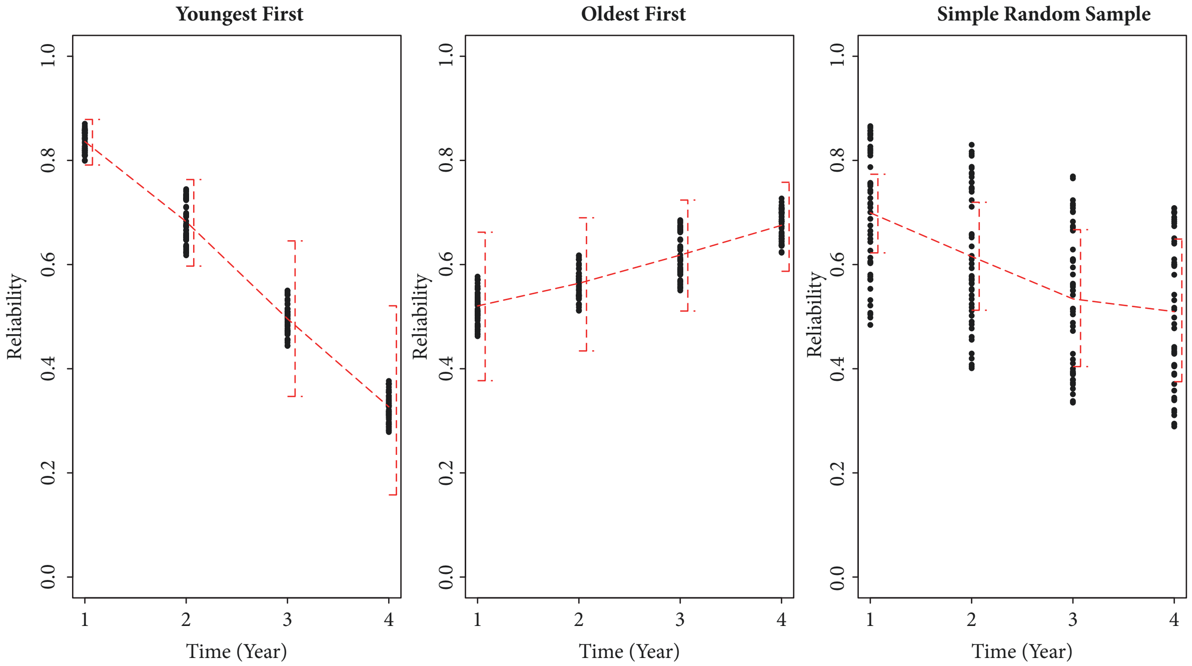


Figure 3: Reliability over time of typical consumption strategies. The dots represent the estimated reliability for individual units used at the end of each year. The dashed curve shows the posterior mean estimate of the expected success rate (i.e., average reliability of 50 units used in each time interval) over the four years. The vertical intervals in red show the 90% credible interval of the expected success rate for each year.

space can be time consuming and could require multiple long runs of GA searches to obtain an adequately populated front. Considering that swapping times of use for any individual units of similar ages is unlikely to yield a large difference in the overall reliability performance quantified in Equations (2)-(4), the search algorithm to find the best strategies can be accelerated by sacrificing a bit of accuracy in an early phase to seek the general patterns associated with leading candidates rather than searching through every single possible candidate solution. To make the problem tractable and improve the likelihood of finding the best possible solutions, a more efficient search can be conducted as a two-phase process. In the first phase, a quicker search among a subset of representative units can provide a broader assessment of the more promising candidates. Then in the second phase, the general features of the identified solutions for the subset can be projected back to the full stockpile to obtain a collection of promising solutions, which can then serve as good starting points to accelerate the search over the entire solution space for the full stockpile. This approach of finding general patterns for the best performance on a representative miniature version of the problem is powerful for accelerating the time needed to obtain good solutions, and dramatically increases the likelihood of not getting trapped at local optima and missing the true Pareto front.

3. Methods

In this application, there are a large number of possible consumption strategies and selecting a single strategy from those possibilities could be overwhelming, and would likely lead to a suboptimal choice, if a structured process were not followed. In the "Reduce" step of the DMRCS process [13], eliminating some potential solutions based on their objective inferiority to other solutions can result in significantly fewer choices that need to be considered. Once the number of contending solutions has been reduced to a more manageable quantity, decision-makers can then incorporate their tailored priorities in the "Combine" and "Select" steps to further narrow down the possibilities to a single, defensible choice. In this section we primarily focus on how to "Reduce" the number of solutions to be considered by using a genetic algorithm paired with the Pareto front approach to decision-making.

3.1. General Nondominated Sorting Genetic Algorithm. Evolutionary computation describes the broad class of numerical optimization techniques and algorithms based on the Darwinian principle of natural selection, or "survival of the fittest" [20]. While a variety of approaches to evolutionary computation exist, the common thread of these algorithms and techniques is that they mimic evolution by maintaining a population of "solutions" that compete to survive and reproduce. Such algorithms are typically run for many generations, with each generation being incrementally better, or more fit than the previous generation, until an acceptable solution is reached.

In each generation of the algorithm, every solution has an inherent "fitness" that impacts the likelihood it is chosen to create new solutions for the next generation. New solutions are created via the operations of recombination (sometimes referred to a crossover) and mutation. Recombination is often considered a multiparent operation while mutation is usually a single parent operation. In general, the purpose of these two operations depends on the type of evolutionary algorithm being implemented. For our purposes, recombination is used to preserve the desirable features of two existing solutions by merging those solutions into a new solution [20], while mutation is used to inject “fresh blood” into the population by randomly generating a slight modification of an existing solution to maintain population diversity [20]. Intuitively, we can think of recombination as incrementally moving the population forward while mutation allows it to occasionally branch out and explore other directions. At the end of a generation, it is determined which solutions should survive to the next generation. Again, there are numerous ways in which this could be implemented; we prefer an elitist strategy in which offspring solutions compete against their parent solutions for survival into the next generation—this ensures that each generation is at least as “fit” as the previous generation [20].

This cycle of fitness evaluation, reproduction, and survival is continued until some termination criterion is achieved. Eiben and Smith [20] describe two typical strategies to determine when an evolutionary algorithm should terminate. In cases where the optimal value is known, the algorithm could terminate once the objective function being optimized is within a tolerable threshold, $\epsilon > 0$, of that known value. Even if the optimal value is known, there is no guarantee that the algorithm with converge upon that value. Further, there are likely to be many situations in which the optimal value is unknown. Thus, this termination strategy could be modified slightly so that the algorithm stops when incremental improvements in the objective function remain below a pre-specified threshold, $\delta > 0$, for a certain number of generations. Another reasonable strategy is to terminate the algorithm after a pre-specified amount of time, which could be defined by elapsed processing time, a fixed number of function evaluations, or a fixed number of generations [20].

There are two main ways that an evolutionary algorithm can be used to solve multiobjective optimization problems. One approach is to combine all criteria into a single objective, such as a desirability function [21], to be optimized by the evolutionary algorithm. To limit the impacts of subjective weighting and scaling choices on the search, multiple evolutionary algorithms are often run using different subjective choices to explore robustness of results across different subjective choices. However, this often comes with the cost of repetitively evaluating the same solutions in searches directed with different subjective choices.

The other approach is to use a class of evolutionary algorithms designed to simultaneously optimize all criteria under consideration, often by identifying the Pareto front. At the end of each generation, the best available set of solutions that simultaneously optimize different prioritizations of the criteria are obtained. One such algorithm is the fast and elitist nondominated sorting genetic algorithm (NSGA-II) [22]. NSGA-II is an improvement upon the previous nondominated sorting algorithm, NSGA. The reduced computational complexity of NSGA-II is achieved with the following nondominated sorting algorithm. For each solution S in the current population, two quantities are calculated: (1) the number of solutions that dominate S (*domination count*) and (2) a list of all solutions dominated by S. All solutions that have a domination count of 0 are nondominated and hence on the first tier Pareto front. After each solution has been assigned a domination count and the solutions it dominates have been determined, the algorithm loops through the solutions on the first tier PF. For each solution S on the first tier PF, the solutions it dominates are considered; each solution dominated by S has its domination count reduced by one; if it’s new domination count is 0, then that solution is on the second tier PF. This process is repeated for all solutions on the lower tier PFs.

Once the population of solutions is sorted into tiers of PFs, NSGA-II ranks solutions on a given tier according to their crowding distance [22], where solutions further separated from others are valued for the diversity they bring to the algorithm. The crowding distance measures how far a solution’s criteria values are from those of other solutions, with solutions that are farther from others assigned a larger crowding distance and a higher rank. The overall crowding distance for a solution is computed as the sum of crowding distances for the individual criteria. To compute the crowding distance for a single criterion, the observed criterion values are sorted from smallest to largest. The minimum and maximum observed values are assigned an infinite crowding distance. For all other values of the criterion, the crowding distance assigned to a specific value is computed by taking the range of values immediately flanking that value and scaling that according to the range of the criterion. Once the crowding distances have been computed for each criterion separately, the overall crowding distance for a solution is determined as the sum of that solution’s crowding distances for all criteria.

Putting these key elements together, the general NSGA-II has five steps. Some details specific to our implementation of NSGA-II, which match the presentation of the NSGA-II steps, are provided below:

(1) Generate an initial population of size *popsize*. This initial population can be randomly generated, or utilize starting points strategically chosen by the user. Details about how solutions are represented in our implementation immediately follow the outline of NSGA-II. In Section 3.2 we discuss how random points are used to initialize the Phase I GA search while Phase II uses starting points strategically chosen based on the results of the Phase I search.

(2) Rank solutions based on (a) which tier PF they are on and (b) their crowding distance. Deb et al. [22] note that solutions on a higher tier PF (i.e., lower domination count) are preferred, but given two solutions on the same tier, a larger crowding distance (i.e., higher rank) is preferred to maintain population diversity.

(3) Create a fixed number of offspring via recombination and mutation. Details regarding our specific implementation of recombination and mutation follow the outline of NSGA-II.

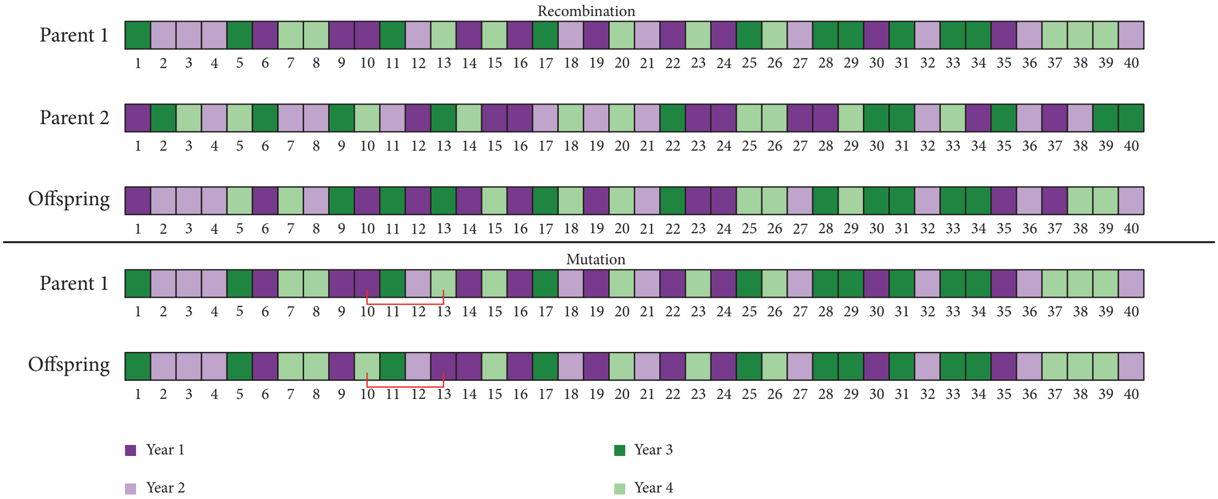


FIGURE 4: Representation of sample solutions, sorted by unit ID, for $N = 40$ and $n_i = 10$. (Top) Construction of a recombination solution from two randomly selected parent solutions (Bottom) Construction of a mutation solution. The coloring of each unit indicates the time period in which the solution assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; and light green = Year 4).

(4) Combine parents and offspring into a single population, rank solutions based on (2a) and (2b) above, and the top *popsize* solutions move on to the next generation.

(5) Repeat Steps (3) and (4) for a specified number of generations. For both phases of the search (Section 3.2), we use 200 generations in our implementation; the impact of that choice is explored in the Supplementary Materials (available here).

In our application, the units in the stockpile are to be used over a period of four years. We assume that the units in the stockpile have a unique ID number from 1 to $N = \sum_{i=1}^{4} n_i = 200$. In our implementation of NSGA-II, we represent a solution as a permutation of the unit IDs with the first n_1 units in the permutation being assigned to Year 1, the next n_2 used in Year 2, and so on. Figure 4 illustrates some sample solutions sorted by their ID number, where different colors are used to indicate assignment to the different time period (dark purple = Year 1; light purple = Year 2; dark green = Year 3; and light green = Year 4); for ease of representation, $N = 40$ units are used in Figure 4, with $n_i = 10$ units used per time period.

To create a single recombination solution in our implementation of NSGA-II, we randomly select two parents from the population of solutions, with a probability inversely proportional to their rank so that those with higher ranks are more likely to be chosen [20]. Once two parents have been selected, their offspring solution is formed by considering each unit and randomly deciding which parent's assigned usage year for that unit will be used in the offspring solution as seen in Figure 4. In the example in Figure 4, Parent Solution 1 uses the first unit in Year 3 (dark green) while Parent Solution 2 uses the first unit in Year 1 (dark purple); to create the offspring solution, we randomly decide in which of these two years the offspring solution should use the first unit, in this example it was Year 1. This process is repeated for each additional unit in the solution. In the event that this random recombination results in an excess number of units assigned to one or more years (and thus one or more of the other time periods have a deficit), units are randomly selected from a year with the overage and allocated to a year with a shortage. In the example Offspring solution from Figure 4, Year 1 has an overage, with 11 units assigned to it, while the Year 2 has a shortage, with nine units. Thus, to finalize this offspring solution the shortages and overages need to be addressed; in this example, this would be resolved by randomly selecting one unit assigned to be used during Year 1 and instead designating it to be used in Year 2. To mutate a single parent solution from the population in our implementation, the usage periods for two randomly selected units assigned to different years are swapped as seen in Figure 4, where the years assigned to units 10 and 13 in the parent solution are swapped in the offspring solution.

In a single run of NSGA-II, the solutions in the population with a nondomination count of 0 form the Pareto front. In our implementation, we use multiple parallel runs of the algorithm to ensure good exploration of the solution space. The final populations from each run are combined together and the final Pareto front (PF) is identified.

We further modify NSGA-II by implementing adaptive population sizing [24]. When random starting points are used to initialize a multiobjective evolutionary algorithm, many of the solutions initially considered are far from optimal in the early generations of the algorithm. For instance, if randomly generated solutions are used to initialize NSGA-II, it is highly likely that only a relatively small number of those solutions would be on the first tier PF. With a fixed population size, this would mean that solutions on lower tier PFs would potentially be used to generate offspring solutions for the next generation, resulting in offspring that likely continue to be inferior. Thus, computation time would be wasted by evaluating these inferior solutions. Instead of using a fixed population size, adaptive population sizing dynamically grows the population based on user specified parameters and the size of the first tier PF [24]. The benefits of adaptive population sizing include faster convergence of the algorithm with reduced computational effort [24]. Similarly, the number of offspring solutions produced each generation can be determined dynamically.

To modify a multiobjective evolutionary algorithm to employ adaptive population sizing, the user specifies a maximum population size (*maxpop*) and maximum number of offspring (*maxoff*) to be created. To initialize the algorithm, $popsize = maxpop$ solutions would be randomly generated. The nondominated sorting algorithm would be used to determine which solutions are on the first tier PF (PF_1). The number of offspring $noff_g$ solutions to be created in Generation g is then dynamically determined as

$$noff_g = \min\left\{c_g + \left\lceil d_g \times \left|(PF_1)_g\right|\right\rceil, maxoff\right\}. \quad (5)$$

Here c_g represents the minimum number of offspring solutions to be created in Generation g, while d_g represents the rate at which the number of offspring created in Generation g grows as the number of solutions on the first tier PF grows. It is possible that c_g and d_g change over generations. We prefer d_g to be a even value so that an even number of offspring are created each generation, with half created via recombination and half via mutation. In our implementation, we use $c_g = 20$ and $d_g = 2$. Thus, each generation a minimum of 20 offspring solutions are created, and for every solution on the first tier PF, two additional offspring solutions are created.

If at the end of Generation g there are $(PF_1)_g$ solutions on the first tier PF, the population size for Generation $g+1$ is determined as

$$popsize_{g+1} = \min\left\{a_g + \left\lceil b_g \times \left|(PF_1)_g\right|\right\rceil, maxpop\right\}. \quad (6)$$

Here, a_g represents the minimum number of solutions to include in the population beyond the first tier PF while b_g represents how the size of the population grows in relation to the size of the first tier PF. Again, a_g and b_g can change over generations. In our implementation, we define a_g relative to the user specified maximum population size (specifically, $a_g = 0.2 * maxpop$) and $b_g = 1$. This ensures that each generation, the population size is slightly larger than the number of first tier PF solutions.

The $popsize_{g+1}$ highest ranked solutions are the ones that move on to Generation $g+1$. There is a drawback to specifying a maximum population size: if $|(PF_1)_g| > maxpop$, then some solutions on the first tier PF do not move on to the next generation—that is, the PF will be truncated. This could mean that a reasonable, contending solution is not made available to the user in the subjective decision-making stage. To prevent this, if $|(PF_1)_g| > maxpop$, we allow *maxpop* to grow based on the size of the first tier PF.

An exploration of the impact of some of the settings used for the GA and R code for our implementation of NSGA-II with adaptive population sizing can be found in the Supplementary Materials.

3.2. Two-Phase Search via Genetic Algorithms. While a directed search via a genetic algorithm is likely to be less time-consuming than a direct optimization approach that considers all possible solutions, it can still be computationally intensive when there are a large number of possible solutions, particularly if random starting points are used in the initial population. To improve the computational efficiency of the search, we propose a two-phase search that begins by implementing a genetic algorithm for a subset of representative units from the stockpile (Phase I) and then mapping the Phase I solutions to the entire stockpile to conduct a fine-tuned search over the entire stockpile (Phase II). The main steps in the two-phased Pareto search algorithms are summarized below:

(1) Phase I begins by selecting a representative subset of units from the full stockpile. Parallel runs of NSGA-II, with random starting points, are used to find the PF for the subset of units. The first tier PFs from each of the multiple runs are combined to form the final PF from Phase I.

(2) Phase II begins by projecting each of the solutions from the Phase I PF to the larger stockpile based on mapping the usage pattern (captured by the age distribution of units used in different time intervals), with these projected solutions used as the starting points for the Phase II GA.

(3) The Phase II GA search algorithm is run multiple times with the same starting population formed by the projected solutions to search over the solution space of the entire stockpile. The resulting first tier PFs are combined to obtain the final Phase II PF.

The Phase II PF provides users with an objective set of solutions from which to make their final decision. Graphical tools [13–17] can then be used by the decision-makers to select a final decision about a consumption pattern that best fits their priorities. In the Supplementary Materials, we compare the required computation time and quality of the PFs identified using the two-phase search for our application to those found by running the GA only on the full stockpile with random starting points. We find that the two-phase search is dramatically more effective in populating the PF in a shorter amount of time. Based on the examination of the single stage search, the completeness and quality of the solutions obtained in a reasonable time are inadequate.

4. Results and Discussion

4.1. Phase I GA Search. To begin the first phase of our search for a consumption strategy with optimized reliability, we choose a representative subset of units from the stockpile of units. This representative subset was chosen by sorting the stockpile of 200 units by age and, then, for every group of five consecutive units in the sorted stockpile of units, one unit was chosen at random for the representative subset, yielding a subset of 40 units for Phase I. This reduces the size of the search space from about 10^{117} possible consumption strategies to about $\binom{40}{10}\binom{30}{10}\binom{20}{10} \cong 10^{21}$; this smaller solution space can be adequately explored in a shorter amount of time.

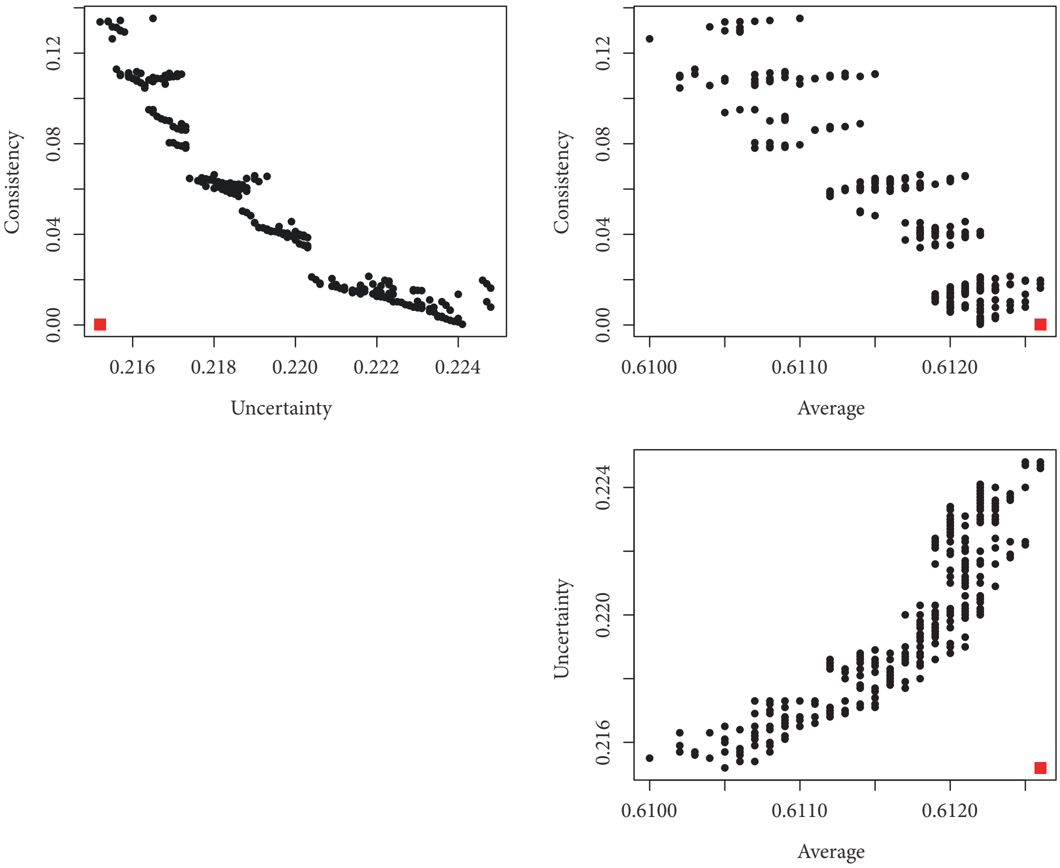


Figure 5: Pairwise scatterplots of Phase I Pareto front. The square symbol in each plot represents the Utopia point, where both criteria achieve their optimal values.

In the first phase of the search, for our implementation of NSGA-II, we specify $maxoff = 200$ and $maxpop = 200$, though the maximum population size is allowed to grow if the number of solutions on the first tier PF exceeds 200 so that the PF is not truncated. Additionally, we choose to run the GA for a fixed amount of time, defined as 200 generations. Since the goal of this phase of the search is to create informed starting points for the second phase of the search, it is less critical to completely populate the full PF with every single solution. We explore the impact of these choices in the Supplementary Materials. For every solution generated by the algorithm, the three metrics ((2)–(4)) to be optimized are computed as described in Section 1 and rounded to four decimal places. We find that using more decimal places increases the number of solutions on the PF and thus the runtime of the GA, without producing meaningful differences in the estimated performance of the solutions identified to be on the PF.

We run five instances of NSGA-II in parallel to facilitate broad exploration of the solution space to effectively populate the PF. Using a machine with an Intel^{R} Xeon^{R} 3.3GHz processor and 16GB RAM, the run times of the five instances ranged from 34.98 minutes to 36.33 minutes, with an average run time of 35.58 minutes. The first tier PFs identified by the five instances of the algorithm ranged in size from 222 to 246 consumption strategies; across the five instances, 1102 unique solutions were identified as being on a first tier PF. The first tier PFs from the five instances of the algorithm were combined, and from that set of over 1100 solutions, the final Phase I PF, with 225 strategies, was identified. The pairwise scatterplots in Figure 5 display the Phase I PF based on a subset of stockpile units while Figure 6 shows the consumption pattern, by current unit age, for the 225 strategies on the Phase I PF.

The panels in Figure 5 show the trade-offs between the computed criteria values for each pair of criteria. Note that the points on the PF for three criteria are now projected onto the plane of each pair of criteria, and the points on the three-criterion PF are not necessarily on the two-criterion PFs. Hence, there are many points located in the inner region of the projected two criteria spaces instead of on the edge where the PF is located. For each pair, the *Utopia point*, where both criteria would simultaneously achieve their optimal values, is plotted with a square symbol. The pairwise scatterplots indicate that the largest trade-offs involve the uncertainty metric with each of the other two metrics, as evidenced by the larger distances between the points on the PF and the Utopia point (bottom left corner in the plot of consistency versus uncertainty and bottom right corner in the plot of uncertainty versus average reliability). These trade-offs indicate that to do well on the uncertainty metric, you must sacrifice on

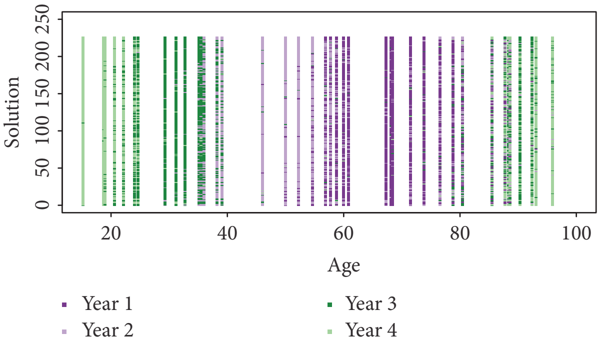


Figure 6: Consumption pattern, for a subset of 40 units, for the 225 strategies identified on the Phase I Pareto front. The units are sorted based on their current ages with the coloring of each unit representing the time period in which the solution assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4).

the consistency and average reliability metrics. On the other hand, there seems to be little trade-off between consistency and average reliability, given the close proximity of the PF to the Utopia point in this case.

Figure 6 shows the consumption patterns for the usage strategies on the Phase I PF, according to the current age of the units in the subset. Here, the coloring of each unit indicates the time period in which the solution assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4). The plot is similar to Figure 2 except the points are not jittered due to the larger number of options to display. The strategies on the PF tend to use units that are currently in the 60 to 75 month range during Year 1 (as indicated with dark purple appearing prominently in this age range), those in the 40 to 60 month range in Year 2 (light purple appears prominently in this range), and those in the 25 to 40 month range in Year 3 (indicated with dark green). In Year 4 (light green), the strategies on the PF tend to use the very youngest and very oldest units in the subset. When the roughly 80-month-old units are used by strategies on the PF tends to be fairly evenly split across the four years. In the next section we discuss how these 225 strategies can be projected on to the entire stockpile of units to create solutions that will form the initial generation of the Phase II GA search.

4.2. Phase II GA Search and Further Decision-Making. In Phase II, we use the PF of 225 strategies identified from Phase I for the subset of 40 units to obtain a set of strategies projected to the full stockpile; these projections preserve the promising consumption patterns identified for the subset of units. Then the projected strategies are used as the starting points for NSGA-II to obtain the PF for the full stockpile of 200 units. The goal is to accelerate the Phase II search by strategically choosing good starting points that are relatively close to the anticipated final PF.

We automatically project each strategy from the Phase I PF to a strategy for the full stockpile based on matching the consumption patterns captured by the current age distribution. Now, we undo the process in Phase I, where a representative subset was selected from the full stockpile by grouping five adjacent solutions with similar ages and randomly selecting one unit from each group. At this stage, we map each unit in the subset back to the same group from which it was selected. Therefore, the subset strategy and the projected strategy have similar consumption patterns, with a similar current age distribution of units used at different times. Figure 7 illustrates how the projection works using a single strategy from the Phase I PF. The horizontal axis is the current age for individual units in the full stockpile (the top row) and the subset of 40 units (the bottom row). Again, the coloring of each unit indicates the time period in which the solution assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4). Each closed circle represents a unit. The circles in Figure 7 show very similar color distributions between the full stockpile and subset strategies, which indicates that the automatic projection procedure works well for mapping the consumption patterns from the subset of units to the full stockpile.

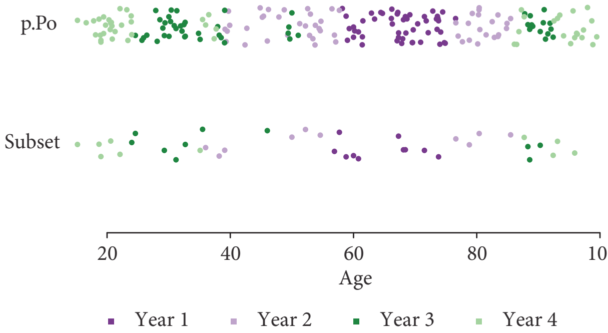


Figure 7: Projection of a Phase I solution (consumption strategy for the subset of 40 units) onto the full stockpile by mapping each unit in the subset to the five adjacent units in the full stockpile with similar ages.

By using the projection procedure to map each strategy on the Phase I PF to a full stockpile strategy, we obtain a collection of 225 consumption strategies for the full stockpile which share the promising features from the subset of units. Then, we use these 225 strategies to form an initial generation to run five independent GA searches for 200 generations, with $maxpop = 200$ and $maxoff = 200$; again, these choices are explored in the Supplementary Materials. Next, we combine the five PFs generated from the five independent searches to obtain an overall combined PF. Figure 8 shows the pairwise scatter plots for the Phase II PF (the combination of five independent PFs from five GA searches using the same initial generation of projected strategies). The 229 strategies on the Phase II PF are shown in black and the 225 starting points are displayed in light gray. First of all, we can see that the Phase II PF has a shape similar to the Phase I PF in Figure 5. Among the three objectives, the most trade-off exists between the uncertainty metric and the other two objectives. There is no strategy that can do very well on both uncertainty and either one of the consistency or average reliability. This can be seen by having no solution close to the Utopia points

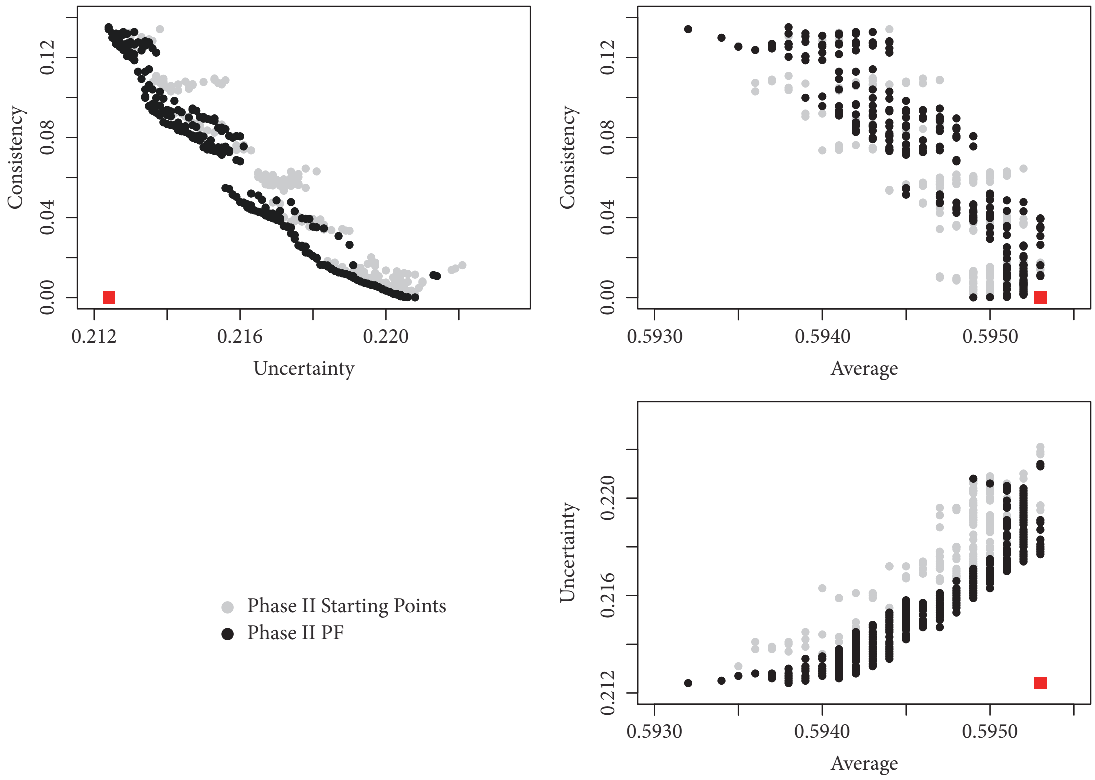


Figure 8: Pairwise scatterplots of Phase II staring points (light gray circles) and Phase II Pareto front (black squares). The square symbol in each plot represents the Utopia point, where both criteria achieve their optimal values.

(square symbols located on the bottom left corner of the top left subplot for consistency and uncertainty, and on the bottom right corner of the bottom right subplot for average and uncertainty). On the other hand, there is much less trade-off between the overall average reliability and the consistency, which is evidenced by having some strategies achieving both nearly best average reliability and most consistency. Secondly, while the starting points for Phase II are located very close to the final PF, the Phase II GA search did help push the solutions on the PF slightly closer to the Utopia points. However, the proximity of the projected solutions to the final PF may indicate that if a quick decision needed to be made in a short time period, then one may consider using the most computing power on the Phase I search to make sure a satisfying PF with all promising solutions and their main features is found for a representative subset. Then a set of near-optimal solutions can be obtained by only projecting the Phase I solutions to the full stockpile without conducting the fine-turned search in Phase II.

To examine the actual consumption patterns among the 229 strategies on the Phase II PF, for each strategy Figure 9 shows the current age distribution for the units with coloring used to indicate the assigned usage time for all units (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4). The plot is similar to Figure 6, but now displays the consumption strategy for all 200 units in the stockpile. One prominent feature of Figure 9 is that some of the units are dominated mostly by a single color. For example, Unit 60, which is the oldest unit in the stockpile, is mostly colored light green for being used in the last year across all 229 strategies, while Unit 136, which is currently 67 months old, is mostly colored dark purple for being used in the first year. By looking across all 229 strategies, we can see they share one common general pattern: the units aged between 60 and 75 months are mostly used in the first year (with dark purple being the dominant color in this age range); the majority units of age 45 to 60 months and also around 80 months are mostly

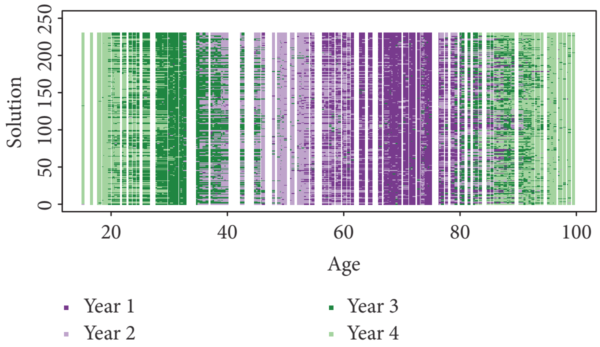


Figure 9: Consumption pattern for the 229 strategies identified on the Phase II Pareto front. The units are sorted based on their current ages with the coloring of each unit representing the time period in which the strategy assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4).

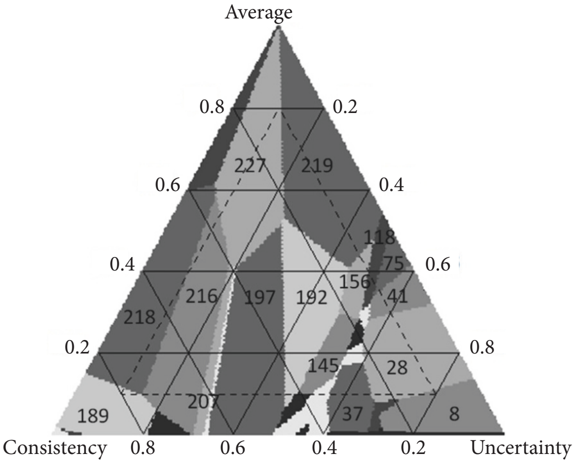


Figure 10: Mixture plot showing the 35 optimal strategies selected for different weighting choices based on using the additive desirability function (DF). The 16 most robust strategies, which are optimal for at least 1% of the total weighting area, are labeled with index numbers.

used in the second year (shaded light purple); a large fraction of units aged between 30 and 35 months and also between 80 and 85 months are used in the third year (dark green), and then lastly most of the units younger than 25 to 30 months old and also above 85 months old are used in the last year (light green). However, there are frequent deviations from the prominent general pattern, particularly around the boundary ages between different time intervals observed in the general pattern.

While the two-phase GA search has greatly reduced the number of solutions that need to be considered from about 10^{117} to 229, there are still a large number of possible solutions. Further, there is a considerable amount of variability in the consumption patterns for the 229 solutions on the Phase II PF, and some of these strategies may be better suited to the user's specific priorities than others. Hence, it is desirable to further reduce the number of solutions to a more manageable number.

In the Combine step of the DMRCS process, we use the desirability function (DF) approach [13] to combine the three objectives into a single metric and identify a smaller set of solutions that are more robust choices based on different possible user priorities. We use the tools outlined in Lu et al. [14], Lu and Anderson-Cook [15], and Lu et al. [17] to guide the remaining decision-making. First, to combine the objectives that are measured on different scales, we need to convert different criteria values to a more comparable scale. Hence, we convert all the criteria values on the PF to a desirability scale between 0 and 1, where the worst value on the PF for each criterion is scaled to 0 and the best value is scaled to 1. Then, we choose an additive DF of the form $DF = w_1 d_1 + w_2 d_2 + w_3 d_3$, where d_j, $j = 1, 2, 3$ is the desirability score for j^{th} objective and the w_j's are the weights subject to $\sum_{j=1}^{3} w_j = 1, w_j > 0$, to combine the scaled criteria values for different weight combinations. Note that we chose this DF form based on our preference for having a less severe penalty for the worst performance on an objective and wanting to allow the best performance on one criterion to offset the worst performance on another. If one prefers more severe penalty for the worst performance, then a different DF form such as the multiplicative DF, $DF = d_1^{w_1} d_2^{w_2} d_3^{w_3}$, can be used.

Figure 10 is the mixture plot showing the best strategies for different weight choices. Each point in the plot corresponds to a single weight combination. For example, the centroid corresponds to the equal weights of $(1/3, 1/3, 1/3)$ for three objectives. The corners and the edges correspond to optimizing based on a single criterion or two of the three criteria, respectively. Different shades of gray are used to distinguish the regions for different strategies. The 35 strategies selected in Figure 10 are the optimal for at least one set of weights out of the 20301 weight combinations explored over the entire weighting space. Out of the 35 strategies, 16 are further selected and labeled with their index numbers (among the 229 strategies on the PF) which are the more robust solutions that are optimal for at least 1% of the total weighting area (i.e., optimal for at least 203 weight combinations evaluated).

Figure 11 is the trade-off plot showing the criteria values for the 35 strategies selected in Figure 10. In the trade-off plot, the x-axis is the selected strategies sorted based on small-to-large average reliability. The y-axis shows the desirability scale for all of the objectives, with the original measurement scales for the objectives labelled on the side. Different criteria values are displayed using different symbols, and the 16 more robust strategies are shown in black to distinguish from the remaining less robust solutions shown in light gray. The line with triangles (consistency) approximately tracks the line with the closed circles (average reliability) with little fluctuations, which indicates there is relatively little trade-off between consistency and average reliability. While the line with squares (uncertainty) generally goes in the opposite direction of the other two lines, which says the main trade-offs exist between the uncertainty and the other two objectives. The 35 strategies can be divided into three groups of different trade-off relationships. The strategies on the left side of Figure 11 all have very little uncertainty, but relatively poor to fair average reliability and the worst

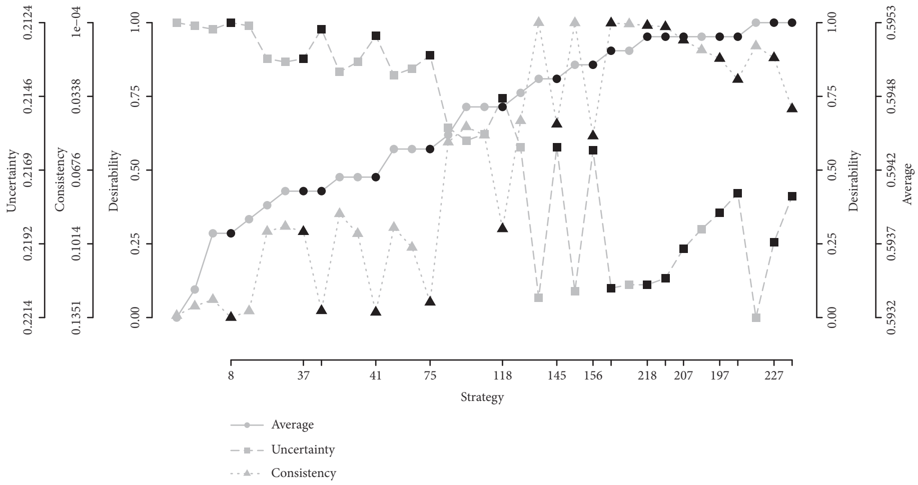


Figure 11: Trade-off plot for comparing the criteria values for the 35 strategies selected in Figure 10. The 16 numbered strategies are displayed in black while the remaining 18 less robust strategies are displayed in light gray. The criteria values are plotted in desirability scale with the original scale shown on the side.

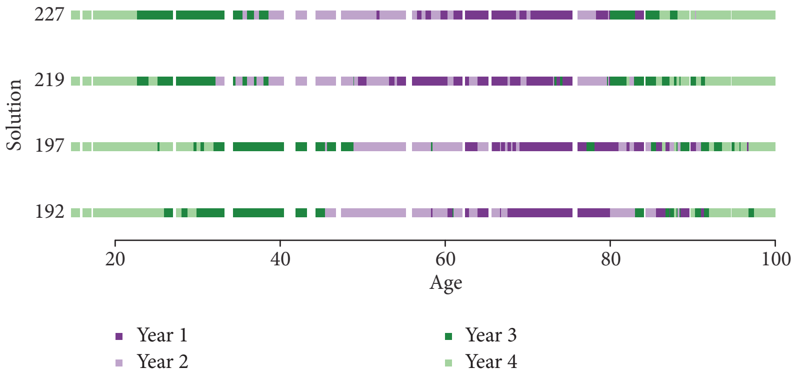


Figure 12: Consumption pattern, displayed by current age distribution, for the four most promising solutions selected based on the mixture plot in Figure 10. The coloring of each unit indicates the time period in which the strategy assigns the unit to be used (dark purple = Year 1; light purple = Year 2; dark green = Year 3; light green = Year 4).

consistency. These strategies are only optimal if uncertainty is considered as the single dominating objective. The strategies in the middle of the trade-off plot have more compromised performance on all three criteria with fair average reliability and uncertainty and poor to fair consistency. These strategies correspond to the relatively smaller areas located in the mid-right region of the mixture plot in Figure 10, which are the optimal choices when uncertainty is valued slightly higher in general. The last group of strategies are located on the right side of the trade-off plot in Figure 11 which all have very good average reliability and consistency but relatively poor uncertainty. These strategies correspond to the larger regions located on the top and bottom left side of the mixture plot (Figure 10) which are generally best when uncertainty is valued as least important.

Choosing from the robust strategies offers more protection against the ambiguity in the specified weights. Therefore, we want to select strategies that match with the general user priorities but also have some robustness against weight ambiguity. For the stockpile manager, typically the average reliability is considered most important among all three criteria. Hence, its weight should be slightly higher than the weights for other objectives. Four strategies look most promising from Figure 10 for satisfying this preference. Strategies 227 and 219 are optimal for large weight regions when average reliability is considered the absolute dominating objective and has at least 50% of the total weight. For the other two objectives, Strategy 227 values consistency slightly more than uncertainty, and vice versa for Strategy 219. However, if the average reliability is considered most important and the two other objectives are also quite important to be weighed no less than 20%, then Strategies 197 and 192 are the most promising choices, which correspond to the large regions close to the centroid of the mixture plot with slightly more emphasis on the average reliability. These four strategies are all located in the right region of the trade-off plot (Figure 11) with high average reliability, good consistency, and poor to fair uncertainty.

Figure 12 offers a closer look at the consumption patterns for the four strategies selected from the mixture plot by focusing on the more preferred weight region. The general

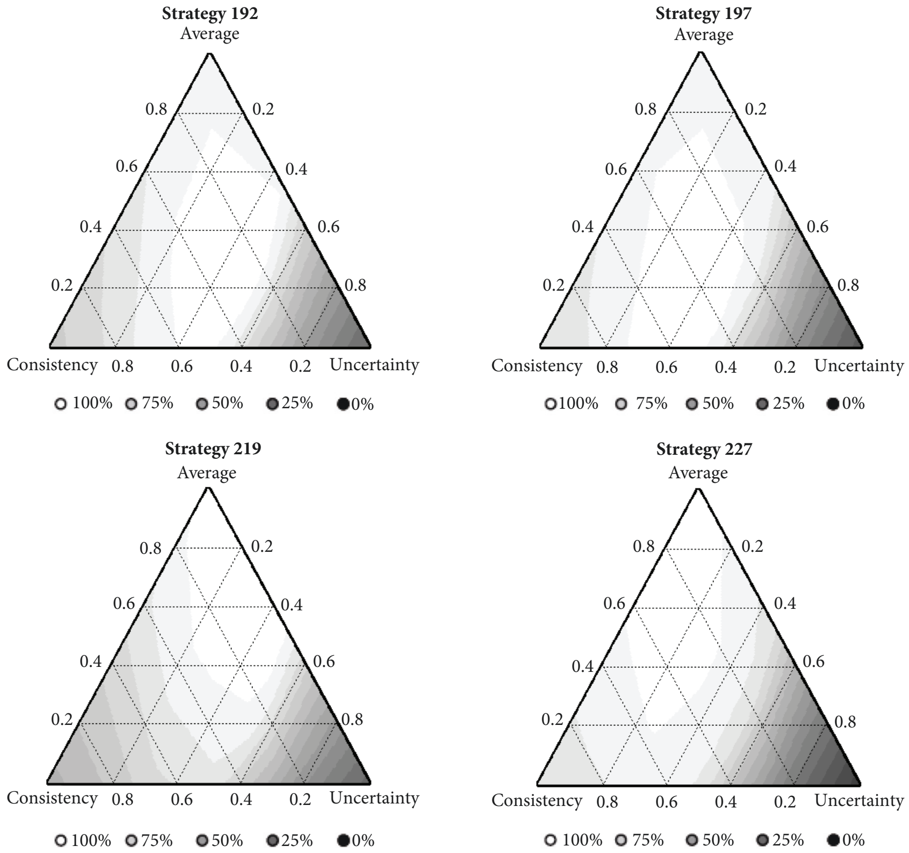


Figure 13: Synthesized efficiency plots for the four promising solutions. The white-gray-black coloring scale shows high to low synthesized efficiency, which measures the relative performance of a strategies relative to the best possible for any given weight.

pattern now looks more prominent across the four strategies, where the units with an age around 60 to 80 months are mostly used in the first year, and then a mixture of some younger units and a few older units are used in the next years, from Year 2 to Year 4, until all units are consumed. However, there are little differences between the strategies. Strategies 227 and 219 have more similar consumption patterns, which generally use fewer younger units and more older units in later years. On the other hand, Strategies 197 and 192 use more older units in early times and more younger units in later years, which consequently leads to slightly lower average reliability but more consistency over time. Hence, Strategies 227 and 219 are preferred when average reliability is valued more dominantly, while Strategies 197 and 192 are preferred when average reliability is considered only slightly more important than the other two objectives. By the end of the Combine stage, we have successfully reduced our choices from the 229 on the PF down to the 17 more robust solutions in the mixture plot, and then down to the four strategies that more closely match the stockpile manager's priorities.

In the final Select step, we compare individual options to make a final decision. Figure 13 shows the synthesized efficiency plots [15] for all four promising strategies by looking at the actual performance of individual strategies relative to the best possible for different weights. The synthesized efficiency measures the relative performance of a strategy compared to the global optimal performance for any given weight. When using a certain DF as an overall measure of the performance, the synthesized efficiency of a strategy, S, at a given weight combination, $\boldsymbol{w}$, is defined as $SE(S,\boldsymbol{w}) = DF(S,\boldsymbol{w})/\max_S DF(S,\boldsymbol{w})$, where $\max_S DF(S,\boldsymbol{w})$ is the global optimal for the given weight $\boldsymbol{w}$. The white-gray-black coloring scale is used to display high-to-low synthesized efficiency. We can see that Strategies 192 and 197 both have mostly white or light-gray corresponding to at least 90% efficiency around the preferred weight region. Strategies 219 and 227 have slightly lower synthesized efficiency for the region with slightly lower weight on the average reliability. All four strategies have pretty good performance for most of the weighting region except the bottom corners with more dominating weights on either of the uncertainty or consistency.

Figure 14 shows the fraction of the weight space (FWS) plot [17], which allows us to make a more compact comparison between the four strategies by summarizing the performance of individual strategies across the entire weighting space. Each curve in the FWS plot corresponds to a single strategy and it shows what fraction of the weighting space

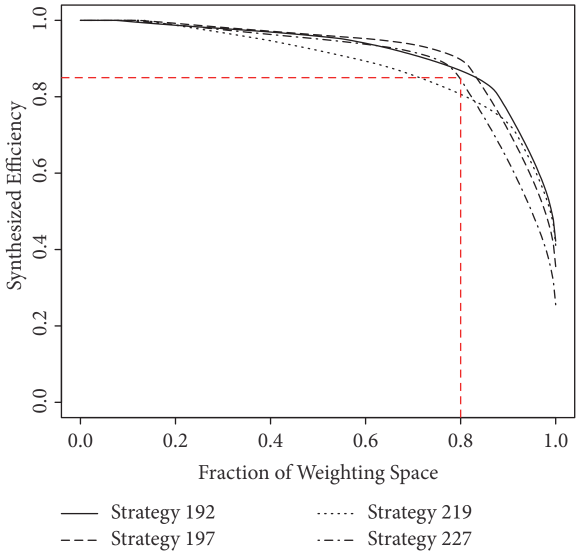


Figure 14: Fraction of Weight Space (FWS) plot for the four most promising strategies.

(x-axis) has synthesized efficiency at least a certain amount (shown on the y-axis). For example, we can see Strategy 192 is at least 85% efficient relative to the best possible choice for 80% of all possible weights. The higher up the curve is located in the FWS plot, the better overall performance a strategy has across the entire weight space. We can see that Strategies 192 and 227 have similar performance for 80% of the weighting space, but the curve for Strategy 227 then drops fast for the remaining 20% of the weighting space. Strategy 197 has a performance similar to the previous two strategies for 60% of the weight space and slightly higher efficiency for the additional 20% weights, but it then becomes less efficient than Strategy 192 for the remaining 20% of the weighting space. Strategy 219 has slightly worse performance than the other three across the entire weight space. Overall, Strategies 192 and 197 are among the top two choices. Strategy 197 has slightly higher efficiency for 80% of the weight space, but Strategy 192 has slightly better worst efficiency over the entire weight space. If we want more protection again the worst case scenario, we would select Strategy 192 as our final choice. We note that the FWS plot provides a compact summary for individual strategies regardless of the dimension of objectives, and hence allows for an easy assessment of overall performance (best, worst, and robustness) of individual solutions and hence easy comparison among many possible competing choices. Also, the FWS plot can be summarized over, not only the entire weighting space, but also any focused weight region [16].

Figure 15 shows the reliability performance for individual units as well as the sets used at the end of each year over the four years for the final selected Strategy 192. Due to the use of a mixture of some younger and older units in each time interval, the reliability of the units at their use times also present a split distribution. For each year, the predicted reliability is divided into two groups of units with high and low reliability, with the difference between the two groups of units increasing quickly over time. However, the expected success rate looks almost flat around 60% reliability over the four years and the uncertainty of the estimated reliability hardly changed over time with the credible interval of the success rate maintains around 50% to 70% over the years.

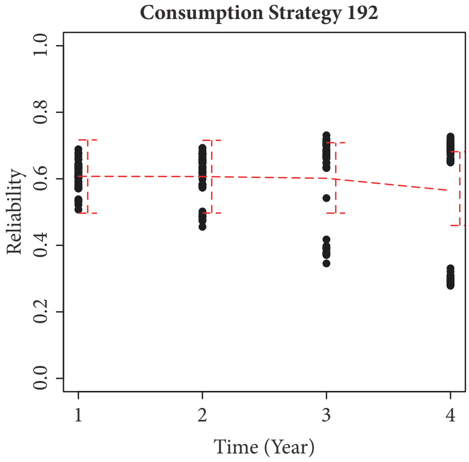


Figure 15: Reliability over time for the final selected strategy (192).

In summary, in Section 4 we used the Pareto front approach to strategically and effectively reduce the number of solutions to focus on some more promising choices tailored to the stockpile manager's priorities. The two-phase search process uses a quicker and smaller scale search on a representative subset to find the superior consumption patterns and maps the patterns found to obtain good starting points to accelerate the GA search for the full stockpile PF. Then the last two steps in the DMRCS process guide a structured quantitative decision-making by gathering useful information and facts to help deepen our understanding of different choices and their trade-offs and be able to make an informed decision tailored to our needs and priorities.

5. Conclusions

In summary, the two-phase approach for finding ideal consumption strategies leverages the formal process for decision-making of DMRCS and takes advantage of efficient computer search algorithms to identify promising solutions. Selecting metrics that quantify the priorities of the stockpile manager and capture the uncertainty in the estimated reliability were key to finding a tailored solution for the optimization. When we compare the strategy identified in Figure 15 with the naïve ones shown in Figure 3, substantial improvements to the average reliability, consistency of reliability over time, and reduced uncertainty of the reliability estimates were all obtained. Given the natural trade-offs between criteria, no strategy was able to perform best for the three metrics simultaneously. Considering only a single criterion can oversimplify the problem and lead to a poor decision with unanticipated poor performance for other importance aspects. However, good performance for all criteria was possible with the strategically optimized solutions found on the Pareto front. We emphasize the importance of having a conscious strategy for the consumption, since not only does a haphazard strategy invite suboptimal performance, but also provides no a priori indication of what performance to anticipate. The double issue of reduced performance with no advanced performance information can lead to dangerous consequences in many applications.

The methodology described in the paper uses metrics well suited to the problem being solved, but is flexible enough to generalize to any set of quantitative measures. It is critical in selecting a best consumption strategy to make sure in the Define and Measure stages of the DMRCS process that care is taken to focus on meaningful metrics important for good performance of the stockpile as it is used.

The implementation of the genetic algorithm has an impact on the solutions obtained and how efficiently the Pareto front is populated. It is helpful to explore the performance of the algorithm with different choices for GA parameters to gain confidence that the implementation is finding near-optimal solutions. However, as results in the Supplementary Materials demonstrate, there are a number of choices for the GA parameters that can lead to solid robust optimization.

The computational intensity of the optimization was greatly reduced and improved results were obtained by using a two-phase approach to solve the problem at hand: general patterns were identified with the manageable first phase based on a representative fraction of the stockpile, and then these solutions and patterns were leveraged in the second phase to obtain promising starting points for further optimization. In general, this approach is beneficial both for finding a timely solution as well as gaining confidence that the best set of possible strategies have been identified. Based on a comparison of results, the single stage search led to an inadequate PF based on its completeness and quality. In addition, for this problem it was discovered that the projected solutions from the first phase were quite close to the overall optimum solutions, and hence represent an option for streamlined solutions. Given that there will be logistical constraints on the implementation of the solution that lead to some suboptimality potentially being introduced, finding regions of solutions with excellent performance rather than the absolute best solution may be a good practical option to pursue.

Data Availability

The data used in the paper are summarized in Tables 1 and 2. The R code for reliability data analysis, the two-phase Pareto front search algorithm, and the further design comparison and evaluation is available from the authors upon request.

Conflicts of Interest

The authors have no conflicts of interest.

Acknowledgments

This work did not receive special funding and was funded by our employers: St. Lawrence University, the University of South Florida, and Los Alamos National Laboratory.

Supplementary Materials

The Supplementary Materials investigate the impact of choices made in implementing the genetic search algorithm and contrast the outcomes from the two-phase search algorithm and a single phase search on the entire stockpile for demonstrating the substantial enhancement in computational efficiency. *(Supplementary Materials)*

References

[1] M. Köksalan, J. Wallenius, and S. Zionts, *Multiple Criteria Decision Making: From Early History to the 21st Century*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2011.

[2] M. K. Ghorabaee, E. K. Zavadskas, L. Olfat, and Z. Turskis, "Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS)," *Informatica (Netherlands)*, vol. 26, no. 3, pp. 435–451, 2015.

[3] W. Sałabun, "The characteristic objects method: a new distance-based approach to multicriteria decision-making problems," *Journal of Multi-Criteria Decision Analysis*, vol. 22, no. 1-2, pp. 37–50, 2015.

[4] M. Zare, C. Pahl, H. Rahnama et al., "Multi-criteria decision making approach in E-learning: a systematic review and classification," *Applied Soft Computing*, vol. 45, pp. 108–128, 2016.

[5] S. Hari Krishna and G. V. Chowdary, "Optimization of simultaneous saccharification and fermentation for the production of ethanol from lignocellulosic biomass," *Journal of Agricultural and Food Chemistry*, vol. 48, no. 5, pp. 1971–1976, 2000.

[6] H. A. Hasan, S. R. S. Abdullah, S. K. Kamarudin, and N. T. Kofli, "Response surface methodology for optimization of simultaneous COD, NH4+-N and Mn2+ removal from drinking water by biological aerated filter," *Desalination*, vol. 275, no. 1-3, pp. 50–61, 2011.

[7] R. P. John, R. K. Sukumaran, K. M. Nampoothiri, and A. Pandey, "Statistical optimization of simultaneous saccharification and l(+)-lactic acid fermentation from cassava bagasse using mixed culture of lactobacilli by response surface methodology," *Biochemical Engineering Journal*, vol. 36, no. 3, pp. 262–267, 2007.

[8] S. L. Correia, D. Hotza, and A. M. Segadães, "Simultaneous optimization of linear firing shrinkage and water absorption of triaxial ceramic bodies using experiments design," *Ceramics International*, vol. 30, no. 6, pp. 917–922, 2004.

[9] N. R. Costa, J. Lourenço, and Z. L. Pereira, "Multiresponse optimization and pareto frontiers," *Quality and Reliability Engineering International*, vol. 28, no. 7, pp. 701–712, 2012.

[10] W. Gronwald, T. Hohm, and D. Hoffmann, "Evolutionary Pareto-optimization of stably folding peptides," *BMC Bioinformatics*, vol. 9, article no. 109, 2008.

[11] E. M. Kasprzak and K. E. Lewis, "Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method," *Structural and Multidisciplinary Optimization*, vol. 22, no. 3, pp. 208–218, 2001.

[12] D. L. Craft, T. F. Halabi, H. A. Shih, and T. R. Bortfeld, "Approximating convex Pareto surfaces in multiobjective radiotherapy planning," *Medical Physics*, vol. 33, no. 9, pp. 3399–3407, 2006.

[13] C. M. Anderson-Cook and L. Lu, "Much-needed structure: a new 5-step decision-making process helps you evaluate, balance competing objectives," *Quality Progress*, vol. 48, no. 10, pp. 42–50, 2015.

[14] L. Lu, C. M. Anderson-Cook, and T. . Robinson, "Optimization of designed experiments based on multiple criteria utilizing a Pareto frontier," *Technometrics. A Journal of Statistics for the Physical, Chemical and Engineering Sciences*, vol. 53, no. 4, pp. 353–365, 2011.

[15] L. Lu and C. M. Anderson-Cook, "Rethinking the optimal response surface design for a first-order model with two-factor interactions, when protecting against curvature," *Quality Engineering*, vol. 24, no. 3, pp. 404–422, 2012.

[16] L. Lu, C. M. Anderson-Cook, and D. K. Lin, "Optimal designed experiments using a Pareto front search for focused preference of multiple objectives," *Computational Statistics & Data Analysis*, vol. 71, pp. 1178–1192, 2014.

[17] L. Lu, J. L. Chapman, and C. . Anderson-Cook, "A case study on selecting a best allocation of new data for improving the estimation precision of system and subsystem reliability using Pareto fronts," *Technometrics. A Journal of Statistics for the Physical, Chemical and Engineering Sciences*, vol. 55, no. 4, pp. 473–487, 2013.

[18] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, NY, USA, 2007.

[19] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.

[20] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Natural Computing Series, Springer, Berlin, Germany, 1st edition, 2003.

[21] G. Derringer and R. Suich, "Simultaneous Optimization of Several Response Variables," *Journal of Quality Technology*, vol. 12, no. 4, pp. 214–219, 2018.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

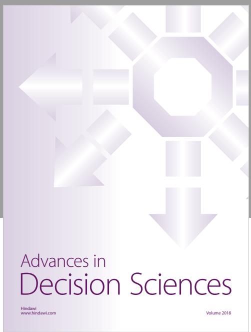
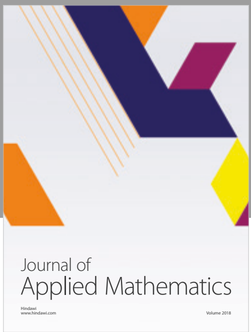
[23] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.

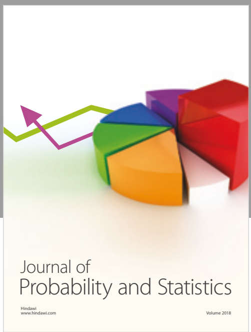
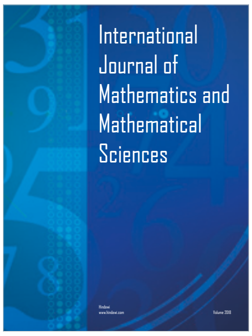
[24] H. Eskandari, C. D. Geiger, and G. B. Lamont, *FastPGA: A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems, In Evolutionary Multi-Criterion Optimization*, Springer, Berlin, Germany, 2007.

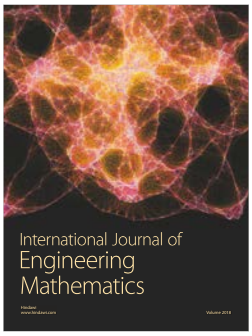
[25] L. Lu and C. M. Anderson-Cook, "Prediction of reliability of an arbitrary system from a finite population," *Quality Engineering*, vol. 23, no. 1, pp. 71–83, 2011.

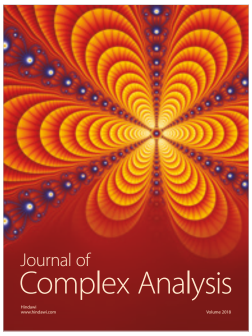
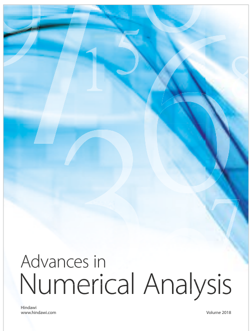
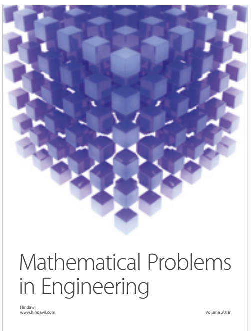
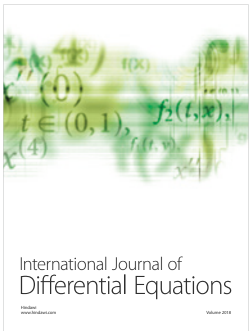
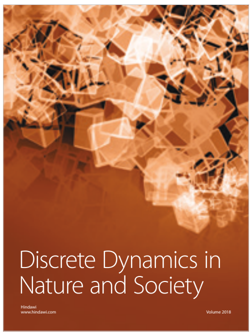
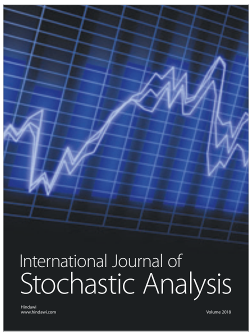
[26] A. Gelman, J. B. Carlin, H. S. Stern, and B. R. Rubin, *Bayesian Data Analysis*, Chapman & Hall/CRC, Boca Raton, Fl, USA, 2nd edition, 2014.

[27] M. S. Hamada, A. G. Wilson, C. S. Reese, and H. F. Martz, *Bayesian Reliability*, Springer, New York, NY, USA, 2008.

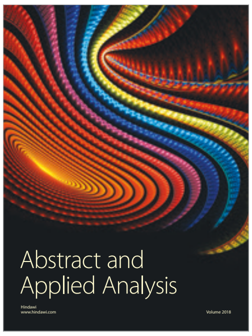
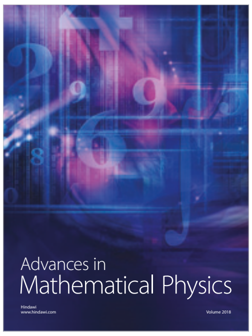
Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

The Scientific
World Journal

Journal of
Probability and Statistics

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Optimization

International Journal of
Engineering
Mathematics

International Journal of
Analysis

Journal of
Complex Analysis

Advances in
Numerical Analysis

Mathematical Problems
in Engineering

International Journal of
Differential Equations

Discrete Dynamics in
Nature and Society

International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces

Abstract and
Applied Analysis

Advances in
Mathematical Physics