Unwanted Feature Interactions Between the Problem and Search Operators in Evolutionary Multi-objective Optimization

Chad Byers(✉), Betty H.C. Cheng, and Kalyanmoy Deb

Michigan State University, East Lansing, MI 48824, USA
{byerscha,chengb}@msu.edu, kdeb@egr.msu.edu

Abstract. Providing self-reconfiguration at run-time amidst adverse environmental conditions is a key challenge in the design of dynamically adaptive systems (DASs). Prescriptive approaches to manually preload these systems with a limited set of strategies/solutions before deployment often result in brittle, rigid designs that are unable to scale and cope with environmental uncertainty. Alternatively, a more scalable and adaptable approach is to embed a search process within the DAS capable of exploring and *generating* optimal reconfigurations at run time. The presence of multiple competing objectives, such as cost and performance, means there is no single optimal solution but rather a *set* of valid solutions with a range of trade-offs that must be considered. In order to help manage competing objectives, we used an evolutionary multi-objective optimization technique, NSGA-II, for generating new network configurations for an industrial remote data mirroring application. During this process, we observed the presence of a hidden search factor that restricted NSGA-II's search from expanding into regions where valid optimal solutions were known to exist. In follow-on empirical studies, we discovered that a variable-length genome design causes unintended interactions with crowding distance mechanisms when using discrete objective functions.

Keywords: NSGA-II · Diversity maintenance · Crowding distance · Discrete objectives · Granularity · Variable-length genome

1 Introduction

Dynamically adaptive system (DAS) are intended to address the challenges posed by adverse environmental conditions [12,14] and varying user requirements. Unlike traditional software systems that can be taken offline and modified by hand, DASs must self-reconfigure at *run time* to avoid staggering financial penalties and/or critical data loss [10]. Smart energy grids, telecommunication systems, smart traffic systems, and similar emerging applications necessitate the deployment of DASs to cope with the various forms of environmental and system uncertainty commonly faced by these applications. These real-world applications

A. Gaspar-Cunha et al. (Eds.): EMO 2015, Part I, LNCS 9018, pp. 19–33, 2015.
DOI: 10.1007/978-3-319-15934-8_2

often contain multiple competing concerns (e.g., cost vs. performance vs. reliability) where trade-offs exist among solutions for dynamic reconfiguration. Evolutionary search techniques, such as genetic algorithms, that rely on biological principles of parallel search provide one approach for the generation of candidate solutions. This paper provides insight into how the underlying solution's encoding may have unintended interactions with specific operators of evolutionary search that produce artificial barriers within the solution space.

In order to automate the generation of DAS configurations, a search-based technique can be embedded within the DAS that is capable of discovering optimal reconfiguration strategies at *run time*. One such evolutionary search-based technique, genetic algorithms (GAs) [5,6], explores a large number of solutions in parallel and uses stochasticity to avoid becoming trapped in suboptimal regions of the solution space. In previous work, we developed Plato [15], a GA-based reconfiguration tool used to evolve overlay networks [1] for a remote data mirroring (RDM) application. In the original Plato tool, user-specified weighting coefficients established a prioritization among the problem's dimensions. Upon further inspection, we found that with this approach, large regions of the solution space were unreachable by evolutionary search, and solutions were often misaligned with the user's weights and therefore ill-fit for their intended environment. To mitigate these issues, we developed Targeting Plato [2] that successfully returned solutions from regions previously unreachable by Plato by targeting desired, user-specified solution qualities. Without *a priori* knowledge of the solution space, two critical disadvantages for using weighting coefficients and target values to guide evolutionary search exist: (1) a trial-and-error approach may be necessary to identify the correct combination of search parameters yielding good solutions, and (2) multiple iterations of search may be necessary to obtain a diverse solution set.

This paper proposes an approach to incorporate multi-objective evolutionary algorithms (MOEAs) into the decision-making process of DASs to generate target reconfigurations at run-time in response to changing environments and requirements. Unlike the GA-based approaches previously mentioned, MOEAs do not require users to specify desired solution characteristics or a prioritization among the problem's dimensions (i.e., objectives) in order to guide evolutionary search. Instead, MOEAs are able to return a *diverse* suite of *Pareto-optimal* solutions whose quality along one particular dimension cannot be improved upon without sacrificing quality along another dimension. As a result, MOEAs can be harnessed for DAS applications to explore the solution space landscape and inform the end user where solution tradeoffs occur.

While leveraging a commonly-used MOEA named NSGA-II [1] for our industrial RDM application, we observed that this algorithm's search coverage was significantly limited when compared to the overall Pareto surface where additional novel solutions existed. In an empirical study, we investigated the potential causes that might prevent search from expanding into these novel regions by performing a set of experiments that targeted specific search operators. Our

[1] Non-dominated Sorting Genetic Algorithm [4].

results revealed that in a discrete optimization problem where the number of solution elements can freely evolve (i.e., variable-length genome), an artificial selective pressure is created that selects against valid regions of the objective space. Specifically, solutions with fewer elements are able to mutate farther distances in the solution space than solutions with a greater number of elements. As a result, solutions with fewer elements are prioritized by NSGA-II's crowding distance operator and search is biased into regions containing these solutions. Despite nearly 20 years of algorithm developmental studies in EMO, this aspect of limited search ability of an EMO due to interactions between varying genome size and its diversity preserving operator has not been studied in depth.

The remainder of this paper is organized as follows. In Section 2, we provide background regarding RDM systems as well as Plato and Targeting Plato. Section 3 discusses our initial observation of the underlying problem when NSGA-II was applied to the RDM problem, where its search coverage was compared to the entire Pareto surface. The series of experiments investigating the root cause of NSGA-II's restricted search performance is provided in Section 4, and its impact on similar application domains and evolutionary approaches is discussed in Section 5. Lastly, Section 6 overviews related work, and Section 7 summarizes our findings and outlines future directions for this work.

2 Background

This section overviews topics fundamental to the approach described in this paper. First, we describe the RDM application, necessitate the use of DASs within this domain, and describe the challenges in their design. Next, we provide an overview of genetic algorithms and discuss their utility within the original Plato dynamic reconfiguration tool for navigating vast, complex solution spaces.

2.1 Remote Data Mirroring

In the RDM application [9] provided by an industrial collaborator, the objective is to *copy* critical data residing at primary sites and *remotely store* (mirror) this data on one or more secondary sites across a network in order to mitigate the presence of site/link failure and ensure file synchronization [10,11]. Two critical design decisions for an RDM solution, referred to as an **overlay network** [1], are (1) the subset of network links to include from the underlying base network and (2) which of two RDM networking protocol types should be used on each active link. A *synchronous* protocol requires that each critical data item is received and applied at all secondary site(s) before proceeding at the primary site. In contrast, *asynchronous* protocols coalesce data items at the primary site that are later sent to secondary sites in batch form and applied atomically after a specified length of time. Table 1 presents the elapsed time between each batch, the amount of data at risk should a failure occur (in GB), and the proportion of bandwidth consumed for synchronous (P1) and asynchronous (P2-P7) protocols.

In our experiments, we evolve overlay network solutions for a fully-connected underlying network containing 26 remote data mirrors. The order of complexity for this problem encompasses $2^{n(n-1)/2}$ network constructions. For 26 remote data mirrors and 7 different RDM protocols, there are 7 * 2^{325} possible overlay network configurations. The RDM application contains multiple competing objectives where trade-offs must be made among solutions' operational cost, performance in bandwidth consumption, and reliability in the face of failure.

Table 1. Properties of synchronous and asynchronous RDM protocols [10]

Protocol Type	**Communication Protocol**	**Interval**	**Data at Risk (GB)**	**Bandwidth**
Synchronous	P1	0 minutes	0.0	1.0
Asynchronous	P2	1 minute	0.35	0.9098
	P3	5 minutes	0.6989	0.8623
	P4	1 hour	1.7782	0.7271
	P5	4 hours	2.3802	0.5732
	P6	12 hours	2.8573	0.4380
	P7	24 hours	3.1584	0.3967

2.2 Plato

Plato [15] is a genetic algorithm-based tool to support RDM reconfiguration at run time according to high-level, user-specified objectives. Within Plato, an evolved overlay network solution is encoded as a vector where each element maps to a specific connection (link) in the base network and stores (1) a boolean flag for whether the connection is used in the solution, and (2) the specific RDM protocol used by the active connection.

Three competing objectives are targeted during optimization: Cost (f_{cost}), Performance (f_{perf}), and Reliability (f_{reliab}). The aggregate formulas for determining these objective values are given in Equations (1)-(7) and were derived from studies for optimizing data recovery systems [10]. In these equations, a candidate solution vector ($\mathbf{x}$) contains N total links from the underlying base network. For each link ($\mathbf{x}_i$), the $\mathbf{x}_i^{flag}$ indicates that the link is active (1) or inactive (0) and $\mathbf{x}_i^{risk}$ and $\mathbf{x}_i^{bandwidth}$ correspond to the data at risk and bandwidth consumed by the link's RDM protocol, respectively. The operational expense of an underlying network link is denoted as C_i, while properties of a particular RDM protocol, such as the bandwidth consumed (ex. $P1^{bandwidth}$), refer to the values in Table 1. To avoid biasing search along objectives with larger ranges of values, each objective is normalized between 0.0 and 1.0.

In the original Plato tool, a user's high-level goals were incorporated into a linear-weighted sum (e.g., $\alpha_{cost}f_{cost} + \alpha_{perf}f_{perf} + \alpha_{reliab}f_{reliab}$) to guide evolutionary search towards regions of desired solutions. As environmental conditions and/or requirements change at run time, the system responds by automatically updating these coefficients to evolve new network reconfigurations. Upon closer inspection, we found that the "surface" containing valid solutions is non-convex and cannot be detected [3] by the linear-weighted sum approach used by Plato. As a result, Plato's evolved solutions were often misaligned with the user's weights and therefore ill-fit for their intended environment.

$$\text{Minimize} \quad (f_{cost}, f_{perf}, f_{reliab}) \tag{1}$$

$$f_{cost}(\mathbf{x}) = \frac{\sum_{i=0}^{N} C_i x_i^{flag}}{\sum_{i=0}^{N} C_i} \tag{2}$$

$$f_{perf}(\mathbf{x}) = \frac{f_{efficiency}(\mathbf{x}) - P1^{bandwidth}}{P1^{bandwidth} - P7^{bandwidth}} \tag{3}$$

$$f_{efficiency}(\mathbf{x}) = \frac{\sum_{i=0}^{N} x_i^{bandwidth} x_i^{flag}}{\sum_{i=0}^{N} P1^{bandwidth} x_i^{flag}} \tag{4}$$

$$f_{reliab}(\mathbf{x}) = 0.5 \times f_{reliab1}(\mathbf{x}) + 0.5 \times f_{reliab2}(\mathbf{x}) \tag{5}$$

$$f_{reliab1}(\mathbf{x}) = 1.0 - \frac{\sum_{i=0}^{N} x_i^{flag}}{N} \tag{6}$$

$$f_{reliab2}(\mathbf{x}) = \frac{\sum_{i=0}^{N} x_i^{risk} x_i^{flag}}{\sum_{i=0}^{N} P7^{risk} x_i^{flag}} \tag{7}$$

To mitigate the aforementioned issues, we developed Targeting Plato [2] where a user specified the desired, *target values* of each objective to be optimized instead of specifying a relative prioritization via weighting coefficients. In this approach, candidate solutions were rewarded for their proximity to the ideal solution's target values. While Targeting Plato provided a more intuitive method for domain experts and expanded search coverage into regions previously unreachable using Plato, these techniques (1) required *a priori* knowledge of the solution space to ensure suboptimal solutions were not returned, (2) were highly dependent on the correct specification of user inputs (e.g., weights and target values), and (3) rewarded solutions for maximizing a combined objective function. As a result, these approaches often sacrificed search exploration for exploitation and returned solution sets often lacking in diversity and trade-offs made among the objectives.

3 Problem Definition

Domain experts often seek to optimize multiple competing (orthogonal) objectives simultaneously in order to assess where trade-offs exist among the problem dimensions. Multi-objective evolutionary algorithms (MOEAs) differ from traditional genetic algorithms in that competing objectives/dimensions are not collapsed into a single objective function but instead are treated individually in order to find a *diverse* set of *Pareto-optimal* solutions.

3.1 Original NSGA-II

For this work, we use the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [4] whose design is particularly well-suited to mitigate the drawbacks of both

Plato and Targeting Plato through the incorporation of two main operators: (1) *non-dominated sorting* and (2) a *crowding-distance* operator. Non-dominated sorting mitigates the concern of suboptimal solutions and ensures that solutions approach true Pareto-optimality by giving priority to solutions whose objective measures dominate (i.e., improve upon) the objective measures of other solutions in the population. NSGA-II's use of a crowding distance operator mitigates the lack of diversity problem by giving priority to non-dominated solutions located in less crowded (i.e., novel) regions of the objective space. In addition, NSGA-II does not rely upon user-specified weighting coefficients or target values.

Using the original implementation of NSGA-II [4], we performed a series of runs to assess its ability to evolve overlay network solutions comparable to the experimental solutions found in previous work [15]. Each run contained a population of 500 candidate solutions employing tournament selection ($k = 5$), two-point crossover, and a 5% mutation rate for a total of 1,000 generations, equating to roughly 3 minutes of wall-clock time. To provide adequate statistical significance, 30 replicate runs were evaluated with each run using a unique random seed.

Using the three objective measures (Cost, Performance, and Reliability), we plotted a three-dimensional point for each solution returned by NSGA-II (colored red in Figure 1). We observed that solutions were clustered around three distinct extrema in the objective space and that, despite the use of NSGA-II's crowding-distance operator, solutions were predominantly located near one another. Moreover, a high Cost measure correlates to more active network links and therefore, a broader range of evolvable Performance levels should exist since each link can support one of seven different RDM protocols. As shown in Figure 1, however, NSGA-II was unable to return solutions with a diverse set of Performance measures as the majority of networks with high Cost only have a Performance level near 0.70. These observations suggested that NSGA-II was not returning a solution set representative of the entire Pareto-optimal surface.

3.2 Epsilon-Constrained NSGA-II (Pareto Surface)

To assess the true shape of the underlying Pareto-optimal surface, we performed a series of additional runs using the epsilon-constraint method for NSGA-II. Using this method, search continues to seek Pareto-optimal solutions that minimize all three objective values described in Equation (1), however, subject to the constraint that they are located within a set of user-specified boundaries. By performing an exhaustive sweep of the objective space using interval sizes of 0.01 along both the Cost and Performance dimensions, we obtained a fine-grained sampling of 10,000 regions across the Pareto-optimal surface.

In Figure 1, we plotted solutions (colored grey) returned by epsilon-constrained NSGA-II and confirmed that the original NSGA-II returned only a limited subset of the solutions on the true Pareto-optimal surface. This observation was troubling for several reasons. First, NSGA-II was unable to return solutions from a large region of the search space where Pareto-optimal solutions were demonstrated to exist, suggesting that hidden factors may be hindering search. Second, despite NSGA-II's use of a crowding-distance operator designed to coerce solutions into unoccupied, novel regions of the objective space, the returned solutions are clustered closely together. Third, the overall shape of the returned solution set tapers towards three distinct regions within the objective space, indicating the potential presence of an artificial selective pressure biasing solutions toward the extremes of each objective.

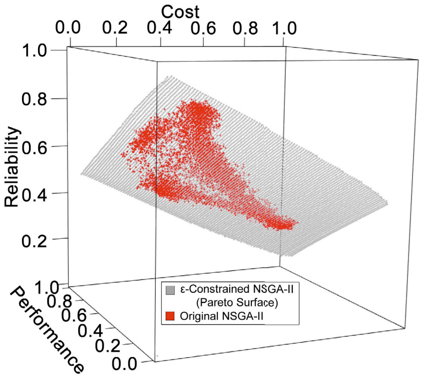

Fig. 1. Original NSGA-II solutions (red) compared against solutions on Pareto surface (grey)

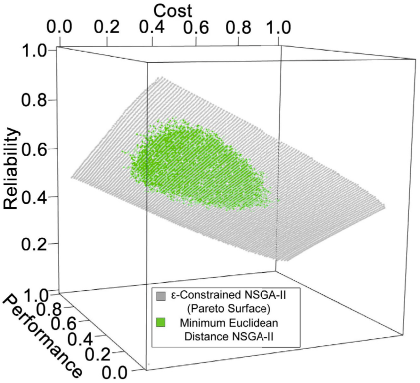

Fig. 2. NSGA-II-MinEuc solutions (green) compared against solutions on Pareto surface (grey)

4 Experimental Design and Results

Next, we describe a series of experiments that were performed to determine the leading causes of (i) artificial basins-of-attraction that solutions evolve towards, (ii) high solution crowding/clustering, and (iii) large regions of undiscovered non-dominated solutions. Each of the experimental treatments comprise 30 replicate runs using the same experimental parameters discussed in Section 3, unless stated otherwise.

4.1 NSGA-II (Minimum Euclidean Crowding-Distance)

Problem/Motivation: Upon analyzing the number of unique Pareto fronts maintained for each generation, we observed that the original NSGA-II rapidly converged to a *single* Pareto front. Consequently, the distinguishing selection factor among solutions becomes the crowding distance operator. In the original NSGA-II [4], crowding distance is assigned by sorting each Pareto front by an objective measure and either (1) awarding positive infinity to "boundary solutions" possessing the minimum/maximum objective values or (2) summing the distance between adjacent solutions' objective values for non-boundary solutions.

While priority is given to solutions maximizing their crowding distance, this implementation may become noisy and overestimate how crowded a solution truly is in the objective space. Adjacent solutions within a *single* objective may be quite distant when their *additional* objective values are taken into consideration. As a result, the original crowding distance operator does not store the distance to the nearest individual solution but rather stores the shortest distances to *any* solution within each objective. In addition, boundary solutions receive the maximum achievable crowding distance thereby producing artificial advantageous regions of the objective space.

Hypothesis 1: By assigning a crowding distance value of positive infinity to boundary solutions, artificial basins of attraction are created that bias search in the original NSGA-II.

Methods: To provide a more accurate distance measure and avoid producing false optima, a new implementation (NSGA-II-MinEuc) was used to replace the original crowding distance with the minimum Euclidean distance between solutions as a diversity-preserving mechanism. This implementation uses *every* objective value of a solution during its distance calculation while also removing the positive infinity assignment bias.

Results: In Figure 2, we observed that the artificial basins-of-attraction anomaly is no longer present with the removal of the positive infinity assignment, and the returned solution set is more evenly distributed in the objective space. Despite similar high levels of solution clustering witnessed previously, NSGA-II-MinEuc is able to *dynamically* respond to boundary solutions in novel areas without an explicit reward/bias. Therefore, in our remaining experiments, we leverage the minimum Euclidean crowding distance operator as we address NSGA-II's restricted search coverage.

4.2 Epsilon-Constrained NSGA-II (Offspring Distance)

Problem/Motivation: From our previous observations, we were able to conclude that the hidden factor restricting search coverage (1) affected the crowding distance values of solutions since NSGA-II quickly converged to a single Pareto front and (2) placed a negative selective pressure on large networks since search coverage tapered off as Cost increased. Taken together, these results suggested that an evolved overlay network's size (i.e. number of active links) might affect the crowding distance its offspring are able to attain.

Hypothesis 2: The number of active networks links of an evolved solution produces a difference in the parent-to-offspring crowding distance values across the objective space.

Methods: To determine whether this differential existed, we used epsilon-constrained NSGA-II to iterate across the objective space in increments of 0.01 in order to measure the average crowding distance between parent and offspring solutions. For each parent solution, we generated 50 offspring solutions and recorded the average Euclidean crowding distance to the parent. We required 100 parent solutions to be discovered within each increment to allow search to potentially discover different solutions with similar Cost and Performance values. In this experiment, we removed the crossover operator to measure two important aspects of search: (1) the average parent-to-offspring mutation distance and (2) the average distance solutions mutate away from where the crossover operator initially places them in the objective space. Therefore, these results indirectly measure how evolved solutions would be affected had crossover been included.

Results: After plotting the average parent-to-offspring distance across the Pareto surface in Figure 3 and applying a color scheme to visualize its topography, we observed that smaller networks with fewer active links (i.e. low Cost) are, on average, able to mutate *farther* from their parents than networks with more active links (i.e. high Cost). In addition, we plotted where solutions were returned by NSGA-II-MinEuc in Figure 4. These results confirmed our initial hypothesis and were a result of the objective functions listed in Equations (1)-(7) being formulated as ratios and normalized, a common approach for many optimization problems. For example, a mutation altering 3 links in a *small* network containing 10 links has a much greater impact (e.g., 30% change in the network's objective values) when compared to altering 3 links in a *large* network containing 100 or 500 links. The ratios of large networks experience greater "inertia," or *resistance to change*, in their objective measures and therefore, receive worse crowding distance values than networks with fewer active links that are capable of moving around the objective space more easily.

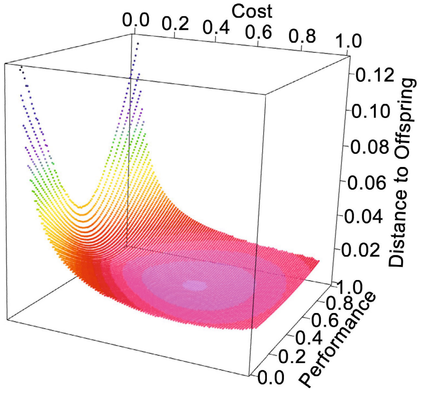

Fig. 3. Average parent-to-offspring Euclidean crowding distance

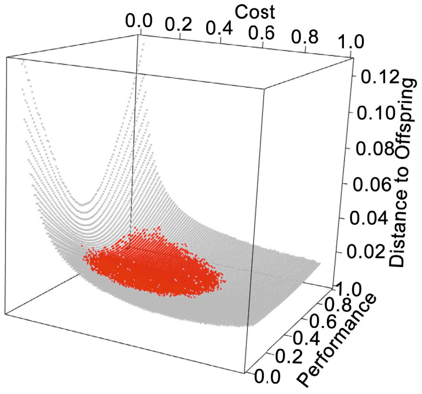

Fig. 4. Locations where NSGA-II-MinEuc solutions were returned

4.3 Epsilon-Constrained NSGA-II (Additional Factors)

Problem/Motivation: One remaining issue left to address is to determine whether NSGA-II is selecting for regions associated with large crowding distances between parent and offspring solutions; what additional factors are preventing NSGA-II's search from expanding into surrounding regions where even larger crowding distance values were shown to exist?

One factor that may prevent search from expanding into low Cost regions associated with larger parent-to-offspring crowding distance values is an increased risk of offspring solutions becoming disconnected. In our RDM application, it is critical that evolved networks remain *connected* meaning that, a path exists from any data mirror to all other data mirrors to ensure copies of critical data items can be distributed should a site failure occur. In our NSGA-II implementation, a disconnected overlay network is considered dominated by any connected solution, regardless of its objective measures.

A second factor that may prevent search from expanding along the Performance dimension is a decreased probability of evolved networks adopting the same RDM protocol across an increasing number of their active links. With seven RDM protocols, the probability of mutation alone producing networks with optimal Performance is $(1/7)^N$. In our experiments, a 26-mirror base network requires *minimally* 25 active links to ensure a connected network resulting in a probability of $(1/7)^{25}$. Although selection and the crowding distance operator work to maintain these solutions, the disruptive effect of crossover and mutation counteract solutions gaining additional identical links. Moreover, each time a novel solution is selected, there is an increased likelihood that its offspring will mutate a shorter distance than its current nearest neighbor causing the region to become more crowded and disadvantageous in subsequent generations.

Hypothesis 3: An increased risk of offspring becoming disconnected inhibits search into regions with higher parent-to-offspring crowding distances.

Hypothesis 4: The net effect of mutation on an evolved solution's objectives opposes search from expanding into novel regions of the objective space.

Methods: Using epsilon-constrained NSGA-II to iterate across the objective space in increments of 0.01, we generated 50 offspring solutions from parent solutions found within each increment. For each offspring solution, we recorded (1) mutation's net effect on Cost, Performance, and Reliability compared to its parent as well as (2) the parent network's edge connectivity. Edge connectivity measures the minimum number of edges that must be removed to cause a network to become disconnected. We required 100 solutions to be found within each increment so that NSGA-II could discover different solutions with similar Cost and Performance measures.

Results: In Figure 5, we observed that the percentage of active links required to disconnect a network decreases with respect to network Cost, matching our expectations since fewer active links increase the likelihood that a critical link is removed in mutation. In addition, we observed a sudden drop in edge connectivity where minimum spanning tree networks have achieved the lowest possible

Cost since the removal of *any* network link disconnects the network. More importantly, Figure 5 demonstrates that NSGA-II-MinEuc returned solutions up until this boundary, thus providing a key insight for why search did not expand further along the Cost dimension.

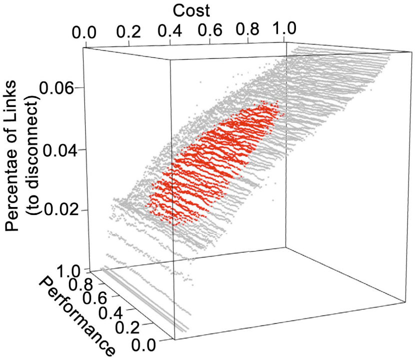

Fig. 5. Percentage of active links to remove and disconnect networks

In addition, we confirmed that mutation opposes the expansion of search into novel regions along the Performance and Reliability dimension as solutions with low objective values experience a net *increase* after mutation is applied and similarly, solutions with high objective values experience a net *decrease*. As a result, these evolutionary operators produce a "funneling" effect as they work to return solutions to a zero net effect point where there is equal probability of activating/deactivating a network link and substituting among RDM protocols.

4.4 Increased Mutation Probability

Problem/Motivation: In previous experiments, we determined that network size affected the parent-to-offspring mutation distance as well as the distance similar networks are able to mutate from one another. While the original NSGA-II appears to have responded to this differential, we have not formally tested whether this factor affected search.

Hypothesis 5: NSGA-II's search forgoes expanding into novel regions of the objective space in favor of regions associated with greater mutation distances

Methods: To test our hypothesis, we divided the objective space into two equal regions: (1) a *Control* region (Cost ≤ 0.50) where we maintained the original mutation rate of 5% and (2) a *Treatment* region (Cost > 0.50) where we increased the mutation rate solutions were exposed to. If our hypothesis is correct, we expect a majority of the final NSGA-II solutions to reside within the Treatment region where an evolved network is more *likely* (not guaranteed) to experience a greater number of mutations and therefore a greater change in their objectives and crowding distance, than networks found in the Control region.

Results: In Figure 6, the 5% mutation rate treatment provides a baseline measure (5.94% ± 1.33%) of solutions in the Treatment region for the original NSGA-II when both regions' mutation rates are equal and thus no difference is present. We observed a significant increase ($p \ll 0.01$) in the percentage of solutions found in the Treatment region as its mutation rate was increased, resulting in the number of solutions found within the Control region dropping from 94.06% in the original NSGA-II to as low as 17.24%. These results confirm our hypothesis that NSGA-II's search forgoes expanding into novel regions of the objective space in favor of regions where higher crowding distances are achievable.

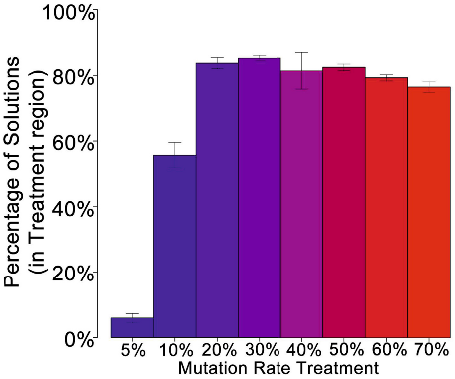

Fig. 6. Percentage of solutions found in the Treatment region

5 Discussion

The experiments in this paper have demonstrated the presence of a hidden interaction between the underlying genome representation and the crowding distance operator found in MOEA approaches such as NSGA-II. Two factors of our application caused this interaction to occur, namely, (1) a *variable-length genome* and (2) objective functions formulated as *ratios*. By allowing the number of solution elements to evolve freely, offspring generated from solutions with fewer elements are more likely to experience a larger change to their objective measures. As a result, offspring having a similar number of elements to their parents but that produce larger crowding distances are likely to be created and accepted by the NSGA-II algorithm. Therefore, NSGA-II's search is unable to locate novel, undiscovered regions of the Pareto front where differently structured solutions reside since these solutions cannot simultaneously achieve similar crowding distance values compared to the current parent members of the population.

The implications of this hidden interaction are significant since, depending on the impact of this interaction, the set of Pareto-optimal solutions returned by NSGA-II may only encompass an extremely limited region of the overall Pareto surface for a given application. More importantly, without knowledge of the underlying Pareto surface, this hidden interaction would be difficult to detect, and the original returned solution set might easily have been accepted as sufficient. In various application domains, the number of solution elements is often a free variable evolved in order to explore different designs and their associated trade-offs. Similarly, a common and often necessary task when normalizing an objective is to formulate the objective as a ratio. As such, this interaction becomes more difficult to detect and also more likely to occur as the dimensionality of the problem increases.

6 Related Work

To the best of the authors' knowledge, the interaction of variable-length genomes and the crowding distance operator as well as its effect on search coverage has not been documented in the literature. Ishibuchi et al. [7,8] examined how discrete objective functions with two different *granularities* (width of intervals within an objective) affected search when using popular multi-objective optimization algorithms including, NSGA-II, SPEA2, MOEA/D, and SMS-EMOA. A two-objective 0/1 knapsack problem [16] with integer profit values for each knapsack item was used in their experiments. By applying rounding factors of different sizes (e.g., round by 10, 100, etc.) to each objective, they explored the effect of different granularity combinations on search. Their results demonstrated that when the granularities of the problem's dimensions vary, search is biased towards particular regions of the objective space.

While these results strongly corroborate our findings and explore a similar problem, several key differences distinguish our work. First, the granularities of each objective were established by applying *ad hoc* rounding factors and were not attributes of the original application. As a result, both of their objectives had *uniform granularity* whereby the interval width was constant within each objective. Second, the granularity of each objective in [7,8] was *static* during search meaning that the interval width did not change from one generation to the next. In contrast, the granularities of the objectives in our work were not established but rather they *emerged* from our application thus concealing the hidden interaction and making it more difficult to detect. Third, the granularity of objectives was *non-uniform* since the interval width was dependent on the number of links within an evolved solution. Smaller networks experienced larger granularities during mutation and larger networks experienced smaller granularities. Lastly, the granularity of objectives was *dynamic* and evolved with respect to network size over time. For example, two networks (network A = 10 expensive links, network B = 30 inexpensive links) with the *same* Cost value of 0.30 will experience different granularities in their Cost objective.

These key differences between our two problems also lead to orthogonal results and conclusions. Ishibuchi et al. [7,8] observed that search was biased towards objectives with finer granularity whereas in this paper, solutions are biased towards the coarse granularity region of each objective. Also, this study found that discrete objectives with coarse granularities improve the search ability of NSGA-II with many dimensions (objectives), whereas the coarse granularity regions of the objective space in our work hinders NSGA-II's search coverage. The combined results of these two independent studies should enable the community to make more informed decisions about which MOEAs to use for problems with similar characteristics or at least should make the researcher more inquisitive of returned solutions.

7 Conclusions and Future Work

In this paper, we first explored the use of NSGA-II for an industrial remote data mirroring application. In this process, we observed the presence of a hidden interaction preventing search from reaching regions on the Pareto surface where optimal solutions were known to exist. Through a series of experiments, we determined that the root cause was the restricted search power of NSGA-II due to an unfavorable interaction between a variable-length genome representation and the crowding distance operator. Solutions with fewer elements experience greater changes to their objective values due to a more coarse-grained granularity and are able to achieve greater crowding distances. As a result, evolutionary search is hindered from exploring regions of the objective space where larger, Pareto-optimal solutions are known to exist. Recently [13], objective function granularity has been highlighted as an important future research area for determining its effects on search performance/dynamics as well as problem analysis. As a newly discovered interaction, we believe this phenomenon is limited to NSGA-II, but is potentially applicable to any EMO procedure, although further studies are needed to confirm this point. These results provide a key insight in this area and raise the level of awareness for researchers exploring multi-objective optimization for domains where the solution's size and/or number of features may evolve freely during search.

Future directions for this work include (1) finding methods, such as epsilon-dominance, that can be incorporated to handle this unintended interaction, (2) surveying other EMO approaches that utilize a different diversity-preserving mechanism and comparing their performance on this problem, and (3) designing a formal test problem to evaluate existing and new EMO algorithms on this rather unexplored interaction between search space properties and diversity preserving operators. Several variables considered for further study include the granularity level within each objective, variation in granularity across an objective, and dynamic changes made to granularity during search.

Acknowledgments. This work has been supported in part by NSF grants IIP-0700329, CCF-0820220, DBI-0939454, CNS-0854931, CNS-0915855. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other research sponsors.

References

1. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. SIGOPS Oper. Syst. Rev. **35**(5), 131–145 (2001)
2. Byers, C.M., Cheng, B.H.: Mitigating uncertainty within the dimensions of a remote data mirroring problem. Tech. Rep. MSU-CSE-14-10, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, September 2014
3. Das, I., Dennis, J.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Structural Optimization **14**(1), pp. 63–69 (1997)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Trans. Evol. Comp. **6**(2), 182–197 (2002)
5. Goldberg, D.E.: Genetic Algorithms in Search. Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)
6. Holland, J.H.: Genetic algorithms. Scientific American, July 1992
7. Ishibuchi, H., Yamane, M., Nojima, Y.: Effects of discrete objective functions with different granularities on the search behavior of emo algorithms. In: Soule, T., Moore, J.H. (eds.) GECCO, pp. 481–488. ACM (2012)
8. Ishibuchi, H., Yamane, M., Nojima, Y.: Difficulty in evolutionary multiobjective optimization of discrete objective functions with different granularities. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 230–245. Springer, Heidelberg (2013)
9. Ji, M., Veitch, A.C., Wilkes, J.: Seneca: remote mirroring done write. In: USENIX Annual Technical Conf., General Track, pp. 253–268. USENIX (2003)
10. Keeton, K., Santos, C., Beyer, D., Chase, J., Wilkes, J.: Designing for disasters. In: Proceedings of the 3rd USENIX Conf. on File and Storage Technologies, Berkeley, CA, USA, pp. 59–62 (2004)
11. Keeton, K., Wilkes, J.: Automatic design of dependable data storage systems (2003)
12. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
13. McClymont, K.: Recent advances in problem understanding: Changes in the landscape a year on. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2013 Companion, pp. 1071–1078, ACM, New York (2013)
14. McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C.: Composing adaptive software. Computer **37**(7), 56–64 (2004)
15. Ramirez, A.J., Knoester, D.B., Cheng, B.H., McKinley, P.K.: Applying genetic algorithms to decision making in autonomic computing systems. In: Proceedings of the 6th International Conference on Autonomic Computing, ICAC 2009, pp. 97–106. ACM, New York (2009)
16. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. Trans. Evol. Comp. **3**(4), 257–271 (1999)