Managing Objective Archives for Solution Set Reduction in Many-Objective Optimization

Amy Peerlinck
Math and Computer Science Department
Western Colorado State University
Gunnison, CO 81230
apeerlinck@western.edu

John Sheppard
Gianforte School of Computing
Montana State University
Bozeman, MT 59717
john.sheppard@montana.edu

***Abstract*—As objectives increase in many-objective optimization (MaOO), often so do the number of non-dominated solutions, potentially resulting in solution sets with thousands of non-dominated solutions. Such a larger final solution set increases difficulty in visualization and decision-making. This raises the question: how can we reduce this large solution set to a more manageable size? In this paper, we present a new objective archive management (OAM) strategy that performs post-optimization solution set reduction to help the end-user make an informed decision without requiring expert knowledge of the field of MaOO. We create separate archives for each objective, selecting solutions based on their fitness as well as diversity criteria in both the objective and variable space. We can then look for solutions that belong to more than one archive to create a reduced final solution set. We apply OAM to NSGA-II and compare our approach to environmental selection finding that the obtained solution set has better hypervolume and spread. Furthermore, we compare results found by OAM-NSGA-II to NSGA-III and get competitive results. Additionally, we apply OAM to reduce the solutions found by NSGA-III and find that the selected solutions perform well in terms of overall fitness, successfully reducing the number of solutions.**

***Index Terms*—many-objective optimization, solution set reduction, evolutionary algorithms**

I. Introduction

Many-objective optimization (MaOO) focuses on solving optimization problems with more than three competing objectives [1]. Such problems are becoming more prominent in real-world applications (e.g., search-based software engineering [2], hybrid car controlling [3], and automotive engine calibration [4].) MaOO comes with added difficulties as compared to multi-objective optimization (MOO). Some of the identified problems of interest are visualization of the solution set, the number of non-dominated solutions found, and diversification of the solutions throughout the search process [5].

To reduce the number of non-dominated solutions as the objective space increases, Multi-Objective Evolutionary Algorithms (MOEAs) are often used; more specifically, decomposition-based approaches such as MOEA/D [6] and NSGA-III [7] are widely used. However, these approaches rely on pre-defined reference vectors to guide the search and adjust the search throughout the optimization process, i.e., they do not offer a way to reduce the solution set post-optimization. Furthermore, such approaches come with their own set of issues, the most prominent being the decrease in diversity and the need to determine the appropriate weight vectors [5]. Several adjustments to these decomposition based methods have been proposed to address these issues [8], [9].

To better address diversity loss in MaOO, methods adjusting the selection criteria have been proposed. This is accomplished by changing the Pareto dominance relationship or creating a specialized fitness function, where the adjustments focus on achieving a good balance between diversity and convergence. This has been accomplished through methods such as α-dominance [10], dominance-ratio adjustment [11], objective reduction based on dominance relations [12], maximum-vector-angle-first principle[13], generalized Pareto optimality [14], clustering of the solutions [15], and adjusted distribution estimation [16]. Similarly, using a performance indicator to evaluate solutions can be an effective strategy. Hypervolume-based evolutionary algorithms are the most common approach [17], [18], but the hypervolume calculation has two serious drawbacks: its dependence on a reference point and the high computational cost [19].

Archive maintenance tactics offer a different kind of solution to the problems found in MaOO. In this approach, the focus lies on an external archive that maintains the set of found non-dominated solutions. Archive management strategies often use ideas from the aforementioned methods, for example, using the hypervolume indicator [20], reference-point based archive management [21], and two-archive based methods where one archive focuses on diversity (indicator-based) and the other on convergence (Pareto-based) [22], [23].

All of the aforementioned approaches adjust the optimization process to address the diversity and solution set size issues. There is no guarantee the resulting solution set will be of a "reasonable" size. In psychology and consumer research, the choice overload hypothesis refers to the fact that "an increase in the number of options to choose from may lead to adverse consequences such as a decrease in the motivation to choose or the satisfaction with the finally chosen solution [24]." Furthermore, research has found that certain factors can exacerbate the effects of choice overload, including difficulty of the task, complexity of the choice set, preference uncertainty, and decision goals [25]. If we consider MaOO to be a such a difficult task that produces a complex solution set, a large number of final solutions (> 26) is more likely to lead to choice overload for the end user. In this paper, we focus on managing the number of non-dominated solutions post-optimization to mitigate the choice overload effect. We propose a multi-archive approach to reduce the non-dominated solution set found by any MOEA to facilitate decision making for the end user, without the need for expert knowledge in the field of evolutionary optimization.

Our archive management strategy creates separate archives for each objective based on the non-dominated solutions produced by an algorithm, where each archive focuses on maintaining the "best" solutions for the relevant objective while introducing diversity. We update the objective archives throughout the generations, and after the final generation, we find the solutions that belong to multiple archives to create a small final solution set to present to the end user.

II. Solution Selection

MaOO with M objectives can be represented as follows, assuming minimization:

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\},$$

where $f_i \in F^M$ represents the objective space, $M > 3$, $\mathcal{X} \in \mathbb{R}^n$ is the solution space, and $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ denotes the decision variables. With an increase in competing objectives, the number of non-dominated solutions in the Pareto front often increases as well, complicating the search process and resulting in large Pareto fronts. When dealing with such large objective spaces, three main problems have been identified [5]: 1) Convergence and diversity are compromised; 2) The curse of dimensionality arises in the objective space; 3) Visualization of solutions becomes more difficult, as does making a final solution choice. The first two problem areas have been addressed in many different ways, mostly focusing on adjusting algorithms to increase diversity or by adjusting the selection procedure (e.g., indicator-based selection instead of Pareto-dominance) [26]. Each of the approaches presented in Section I offers ways to balance convergence and diversity as the objectives increase but do not address the issue of large non-dominated solution sets. Most research focuses on helping the decision maker in their choice for a final solution focuses on dimensionality reduction to aid in visualization [19] or by incorporating preferences directly into the search processes [27]. However, dimensionality reduction comes at the cost of information loss in the objective space, and preferences are highly domain-specific.

There has been research in selecting a subset of solutions after the final non-dominated solution set has been generated, but most research in this area has focused on using the hypervolume metric to find the best solution subset [28], [18]. However, as previously mentioned, the hypervolume indicator comes with two major drawbacks [19]. A more promising approach was presented by Takagi *et al.*, where they perform environmental selection based on an MOEA's chosen selection procedure, for example, crowding distance as used by NSGA-II [29]. We identified two potential downsides to this approach.

Algorithm 1 Objective Archive Management

Input: Number of objectives M, ND archive $\mathcal{N}$, selection parameter k, diversity parameter ℓ

$\mathcal{F} \leftarrow \{\}$
$k \leftarrow \lceil k \times |\mathcal{N}| \rceil$
for all $i = 1$ to M **do**
  $\mathcal{F}_i \leftarrow \{\}$
  $\mathcal{N}' \leftarrow$ sort($\mathcal{N}$, i)
  $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \mathcal{N}'[: k]$
  $\mathcal{N}'' \leftarrow$ diversify_archive($\mathcal{N}'[k : 2k]$, ℓ) // Algorithm 2
  $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \mathcal{N}''$
end for
return $\mathcal{F}$

First, it requires a pre-defined solution set size, and second, it depends upon a specific algorithm to be selected to determine the type of environmental selection to be applied. The former means the end user needs to know how many solutions they want to keep, and the latter means that expert knowledge is required to make an appropriate choice [30]. Our research tries to address the post-optimization solution selection issue using the proposed Objective Archive Management (OAM) approach. In other words, we aim to reduce the amount of non-dominated solutions as generated by any MOEA without reducing the number of objectives, defining a fixed size of the final solution set, or the need for expert knowledge (either to determine reference vectors or for algorithm selection).

III. Objective Archive Management

Since we are organizing a group of non-dominated solutions S into subgroups S_i for each objective, we call our algorithm "objective" archive management (OAM). Our approach is as follows. For each objective M_i, we sort S according to M_i. The first $k\%$ of the sorted solution set is added to S_i. Then the second $k\%$ of the sorted solution set is selected, from this second $k\%$, $\ell\%$ diversity solutions are chosen; half of which are diverse in the objective space, and half of which are diverse in the variable space, where both spaces are normalized (Algorithm 1). The collection of archives is referred to as the Objective Archive (OA). Diversity is determined by creating a dissimilarity or distance matrix $\mathbf{M}_d$ for the solutions' variables ($\mathbf{M}_{d_{var}}$) and fitness scores ($\mathbf{M}_{d_{fit}}$) separately (Algorithm 2). By checking both objective and variable diversity, we aim to account for biased problems (as defined in [31]). In our experiments, we use the cosine similarity metric to measure diversity due to its useful qualities in high dimensional spaces. Specifically, cosine similarity distinguishes different solutions from a directional perspective, making it a good choice to diversify the solution and variable space [32]. Note, however, that any distance metric can be used. Selecting for diversity in this way ensures that the chosen diversity solutions are still good solutions for objective M_i. However, we do not wish to select solely based on diversity. Since diversity is calculated based on all objective values and all decision variables, the objective we are considering for our OA does not influence the diversity of the solutions. As a result, if we were to choose solutions solely based on diversity without taking their ranking

	Sol1	Sol2	Sol3	Sol4	Sol5
Obj1	5	2	4	1	5
Obj2	2	2	4	3	6
Obj3	3	6	4	5	2

TABLE I: Example set of five solutions for three objectives.

k	Obj #	Selected solutions		
40%	1	sol4	**sol2**	
	2	**sol2**	**sol1**	
	3	sol5	**sol1**	
60%	1	**sol4**	**sol2**	**sol3**
	2	**sol2**	**sol1**	**sol4**
	3	sol5	**sol1**	**sol3**

TABLE II: Example of selected solutions for each objective based on parameter k. Bold solutions are those selected for the final archive based on $oc = 2$.

Algorithm 2 Diversify Archive

Input: Solution set $\mathcal{S}$, diversity parameter ℓ

$\ell \leftarrow \lceil \ell \times |\mathcal{S}| \rceil$
$S' \leftarrow \{\}$
$S_{var} \leftarrow \{X_0, \dots, X_{|\mathcal{S}|}\}$
$S_{fit} \leftarrow \{F_0, \dots, F_{|\mathcal{S}|}\}$
$\mathbf{M}_{d_{var}} \leftarrow$ cosine_distance(S_{var})
$S'_{var} \leftarrow$ sort($\mathbf{M}_{d_{var}}$)
$S' \leftarrow S' \cup S'_{var}[: \ell/2]$
$\mathbf{M}_{d_{fit}} \leftarrow$ cosine_distance(S_{fit})
$S'_{fit} \leftarrow$ sort($\mathbf{M}_{d_{fit}}$)
$S' \leftarrow S' \cup S'_{fit}[: \ell/2]$
return S'

for the relevant objective into account, the same diversity solutions would be selected for each archive. Consequently, the selected diversity solutions would not be diverse.

To illustrate the intuition behind the design choice of the k parameter and the use of overlap to find the final non-dominated solution set, we created a toy example with five non-dominated solutions and three objectives. Table I shows the objective values for each solution, and Table II shows which solutions are selected for each OA. In our example we do not take diversity into account to showcase the influence of k. When $k = 60\%$, the three best solutions for each objective are chosen to be added to the objective archive.[1] As we can see, when k is set to be a larger percentage, this means more solutions will be selected, resulting in more balanced solutions being added to each archive. However, even if k is small, the overlap count oc ensures the solutions being chosen perform well on at least oc number of objectives, thus avoiding solutions that only perform well on a single objective.

We can now use the created OA to reduce the non-dominated solution set into a more manageable size. We do this by counting how many times each solution occurs in the

[1]Note that in practice, we would not set $k > 50\%$, since this would result in a large number of solutions being retained, which defeats the purpose of solution set reduction.

Algorithm 3 Find Overlapping Solutions

Input: Objective archive OA, overlap count oc
Init: Dictionary *count* $\leftarrow \{\}$, reduced solution set $\mathcal{S} \leftarrow \{\}$

arch $\leftarrow$ flatten(OA)
for all $X \in$ arch **do**
  count$[X] \leftarrow 0$
end for
for all $i = 1$ to M **do**
  for all $X \in OA_i$ **do**
    count$[X]$ = count$[X] + 1$
  end for
end for
for all $x, c \in$ count **do**
  if $c \geq oc$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
  end if
end for
return $\mathcal{S}$

M objective archives (Algorithm 3). The user can then choose how many archives a solution needs to belong to (overlap count oc) to be included in the final solution set.

The OAM approach can be applied in two different ways to reduce the non-dominated solution set: by keeping an external OA that is continuously updated at each generation of an MOEA, or by applying the OAM strategy a single time to the non-dominated solution set produced after an MOEA has finished running. We refer to the former as E-OAM (external OAM), and the latter as S-OAM (single OAM). In both cases, the final OAM can be used to find overlapping solutions to reduce the number of non-dominated solutions (Algorithm 3).

IV. Experiments

In preliminary studies, we applied NSGA-II, MOEA/D, and SPEA2 to the DTLZ [33] and WFG [31] benchmark suites using different variable grouping strategies [34]. We found that NSGA-II performed well on MaOO problems regardless of variable grouping (as compared to MOEA/D and SPEA2). As a result, we decided to use NSGA-II as our base algorithm; however, as previously stated, the OAM approach could be applied to any algorithm. We compared OAM-NSGA-II to the environmental selection approach in [29] and NSGA-III [7], lastly, we apply OAM to the results obtained by NSGA-III. Each algorithm was run with a population of 1000, for 100 generations. Through preliminary studies, we found that solving DTLZ5, DTLZ6, WFG3, and WFG7 resulted in large non-dominated solution sets (> 500). We used these four functions for our experiments, each with 5 and 10 objectives and 100 decision variables [31]. We performed 30 independent iterations of the algorithms on each problem and report Hypervolume (HV) [35] and Spread (S) [36]. The Wilcoxon rank-sum test with $\alpha = 0.05$ was performed to assess statistical significance for all results.

TABLE III: Chosen parameter combinations (k and ℓ).

Problem	DTLZ5		DTLZ6		WFG3		WFG7	
M	5	10	5	10	5	10	5	10
k	0.40	0.40	0.50	0.50	0.40	0.50	0.50	0.50
ℓ	0.30	0.50	0.40	0.40	0.40	0.40	0.20	0.40

A. Convergence vs. Diversity

Before presenting the results of the comparative analysis, we empirically examined the influence of the convergence and diversity parameters k and ℓ on solution quality when using the E-OAM strategy. As explained in Section III, we do not expect to find statistically significant differences in the k and ℓ parameters. We ran experiments for all combinations of $k = \{0.25, 0.4, 0.5\}$ and $\ell = \{0.2, 0.3, 0.4, 0.5\}$. We evaluated the archive at each generation using HV and spread. [2]

Following statistical hypothesis testing, we found that there was no significant difference between the different ℓ parameter settings for HV and spread on problems DTLZ5, DTLZ6, and WFG3. The same does not hold true for WFG7, where we do find statistically significant differences. However, there is little to no convergence for either HV or spread for WFG7, regardless of the chosen k and ℓ parameters. Additionally, it is interesting to note that spread decreases for DTLZ5, which indicates more similar solutions are being found. We believe the problem lies with the performance of NSGA-II; since the OA is only updated with solutions found by the underlying optimization algorithm, it is directly influenced by that algorithm's performance. In other words, if the underlying algorithm has trouble finding good solutions, the OA does not improve these solutions, it simply selects a subset from the solutions. Table III shows the parameter combinations we used throughout the rest of the paper based on our results.

B. Environmental Selection Results

In this section, we consider the solution set reduction aspect of OAM compared to Environmental Selection (ES) [29]. In these experiments, we performed reduction using both OAM and ES on the same non-dominated solution set generated by NSGA-II. We used the generated OA to find overlapping solutions to determine the reduced solution set. We looked at the number of solutions generated by both E-OAM and S-OAM with overlap equal to 60% and 80% of the number of objectives. The resulting solution set sizes, as well NSGA-II's solution set size, are shown in Table IV. Using 80% overlap results in empty solution sets for some of the problems, we recognize that this is an important flaw in our method and discuss future work to address this issue in our conclusion. In this paper, we decided to use 60% overlap to generate the final non-dominated solution sets to avoid empty solution sets.

We applied ES to both the solutions found in the complete OA generated by E-OAM before finding overlap and to the non-dominated solution set generated by NSGA-II. We set the number of the selected solutions to be the same as the number created by the OAM overlap. This means the number of solutions selected from the OA was the same as the number of solutions generated by the E-OAM overlap, and the number of solutions selected from NSGA-II's solution set was the same as those from S-OAM. Therefore, we denote the two different ES-based selections as ES-E and ES-S respectively. The choice of parameters k and ℓ can be found in Table III. We report HV and S for the original non-dominated solution set from NSGA-II as well as for the different implementations of OAM and ES. We compare the quality of the selected solutions through HV (Table V), S (Table VI), and solution visualization through radar plots. [2]

E-OAM not only reduces the solution set to a more manageable size but improves HV and S for most problems as compared to NSGA-II's non-dominated solution set. Considering classic NSGA-II does not use any archive management strategy, this makes sense since we are keeping track of all non-dominated solutions found when using the E-OAM approach. The interesting part is that the reduced solution sets still have an improved HV and S for most problems, compared to NSGA-II, which only has significantly better results for spread on the 5-objective WFG3. When comparing

[2]Due to space limitations, we were not able to include all of our results; therefore, we provide supplementary materials at the following link https://github.com/AmyLinck/OAM-supplementary .

TABLE IV: Average size for NSGA-II, E-OAM, and S-OAM with different overlap sizes (indicated by the percentages).

		NSGA-II	E-OAM		S-OAM	
Problem	M		60%	80%	60%	80%
DTLZ5	5	931	46	0	144	35
	10	887	100	31	354	111
DTLZ6	5	654	140	3	381	129
	10	776	123	53	291	127
WFG3	5	643	36	0	49	49
	10	837	47	33	478	297
WFG7	5	995	61	2	326	75
	10	1000	474	119	285	251

TABLE V: HV for NSGA-II, OAM, and ES. Bold indicates statistical significance with $\alpha = 0.05$.

Problem	M	NSGA-II	E-OAM	ES-E	S-OAM	ES-S
DTLZ5	5	0.988	**0.997**	0.985	0.987	0.985
	10	0.985	**0.998**	0.980	0.982	0.980
DTLZ6	5	0.912	**0.968**	0.879	0.912	0.880
	10	0.915	0.920	0.880	0.914	0.881
WFG3	5	0.757	0.758	0.756	0.757	0.756
	10	0.066	0.066	0.060	0.066	0.060
WFG7	5	0.106	**0.201**	0.093	0.105	0.093
	10	0.062	**0.190**	**0.144**	0.062	0.049

TABLE VI: S for NSGA-II, OAM, and ES. Bold indicates statistical significance with $\alpha = 0.05$.

Problem	M	NSGA-II	E-OAM	ES-E	S-OAM	ES-S
DTLZ5	5	0.005	**0.0151**	0.006	0.003	0.001
	10	0.006	**0.034**	0.007	0.004	0.000
DTLZ6	5	0.058	**0.117**	0.019	0.058	0.014
	10	0.049	0.048	0.009	0.048	0.002
WFG3	5	**0.011**	0.006	0.007	0.006	0.000
	10	0.006	**0.270**	**0.190**	0.006	0.001
WFG7	5	0.047	**0.346**	**0.212**	0.046	0.001
	10	0.077	**0.623**	**0.450**	0.077	0.002

TABLE VII: AC and size for NSGA-III and E-OAM-NSGA-II.

		NSGA-III		E-OAM	
	M	AC	Size	AC	Size
DTLZ5	5	60.80%	25	39.20%	38
	10	77%	266	23%	90
DTLZ6	5	85.50%	70	14.50%	130
	10	83%	714	17%	160
WFG3	5	100%	80	0%	34
	10	99%	188	1%	48
WFG7	5	100%	70	0%	64
	10	100%	715	0%	400

TABLE VIII: HV and S for NSGA-III and S-OAM NSGA-III. We found no statistical significance with $\alpha = 0.05$.

		NSGA-III		S-OAM	
	M	HV	S	HV	S
DTLZ5	5	0.999	0.025	0.999	0.016
	10	0.999	0.056	0.999	0.040
DTLZ6	5	0.999	0.143	0.999	0.142
	10	0.999	0.231	0.999	0.224
WFG3	5	0.871	0.522	0.871	0.219
	10	0.188	0.888	0.187	0.390
WFG7	5	0.613	1.861	0.512	1.844
	10	0.468	3.847	0.322	3.355

ES to OAM, we see that for both single and external use, OAM has better performance than ES on all problems.

C. NSGA-III Results

The reduced solution sets created when using NSGA-II are still considered large sets (> 26). Therefore, we investigated the effect of applying OAM to NSGA-III, an algorithm adjusted to improve performance on MaOO problems. Our final experiments considered two different aspects of the algorithm as compared to and applied to NSGA-III:

1) We compared the final results from NSGA-III to the final archive found by E-OAM with NSGA-II.
2) We applied S-OAM to the final results set found by NSGA-III.

Since NSGA-III relies on reference vectors to guide the algorithm through the search space, we ran experiments with different numbers of partitions using the Das-Dennis approach to generate different-sized sets of reference vectors [37]. We tested 3, 4, 6, and 9 reference vectors. [2] Based on these results, we use four partitions to generate the reference vectors.

1) External Archive Solutions: This section compares the non-dominated solution set found through overlap from OAM-NSGA-II to the non-dominated solutions found by NSGA-III. In addition to HV and spread (Table VIII), we calculated adjusted coverage (AC), which combines the solutions from NSGA-III and E-OAM-NSGA-II into a new non-dominated front, and calculated how many solutions from each solution set remain in the new front [38]. As mentioned previously, OAM does not influence algorithm performance. Given that NSGA-III has been shown to improve performance for MaOO, it came as no surprise that NSGA-III covered most or all of the non-dominated solutions found by NSGA-II (Table VII) or that NSGA-III had significantly better spread for all problems [19]. However, given this information, it is interesting to note that E-OAM-NSGA-II performed reasonably well on the DTLZ problems. According to Huband *et al.*, DTLZ5 and DTLZ6 represent degenerate Pareto fronts [31], but this attribute no longer holds true when $M > 4$. This could explain why a more complex algorithm no longer has as much benefit.

2) Direct Solution Set Reduction: In addition to the comparative analysis, we applied S-OAM to the results found by NSGA-III to show its general applicability and to further investigate the type of results that are selected when applying S-OAM. We investigated the influence of the k, l, and oc parameters on the solution set size. [2] As expected, the number of chosen solutions gradually decreased with smaller k and l parameters, with bigger jumps occurring when the k parameter changed. Based on our results, we set parameters $k = 0.4$, $\ell = 0.5$, and $oc = 60\%$ for 5 objectives, selecting $15-20\%$ of the total solutions, and $k = 0.4$, $\ell = 0.4$, and , and $oc = 80\%$ for 10 objectives, selecting $7 - 15\%$ of the total solutions.

The generally similar HV scores for NSGA-III and S-OAM-NSGA-III (Table VIII) indicate that the found solution sets are similar in quality i.e., E-OAM-NSGA-II finds a subset of the solutions found by NSGA-III, which we assessed visually as well. [2] In other words, even though NSGA-III resulted in more diversity, this is likely because NSGA-III was keeping solutions with a relatively large increase in one objective score to gain a small decrease in another objective score. When applying S-OAM, it removes many of these solutions while maintaining the increased diversity in the NSGA-III solution set. This indicates that if the original algorithm generates a diverse set of solutions, OAM can reduce this solution set successfully to a smaller size, mitigating the choice overload problem, while maintaining diversity.

V. Conclusion and Future Work

As the number of objectives of a problem increases, the number of non-dominated solutions often does as well. This could lead to a phenomenon known as choice overload [24]. We introduced the Objective Archive Management (OAM) strategy to reduce the final non-dominated solution set size for MaOO to address the choice overload problem. The presented approach has several benefits compared to existing approaches: it requires no pre-defined reference vectors, it can be applied to any algorithm or any solution set, the end-user does not need MOEA-specific knowledge, and it is easy for a user to choose a preference for edge solutions or balanced solutions. Through empirical analysis, we found that the OAM approach selects diverse solutions with good overall fitness. Furthermore, we were able to reduce the solution sets to contain $5-25$ solutions, which are considered "small", and a small solution set reduces the chance of choice overload occurring [25]. Overall, we conclude that using OAM for solution set size reduction performs well regardless of which algorithm is used to create a non-dominated solution set.

In this paper we only applied the external OAM strategy to improve the quality of the reduced solution set; we believe the external OAM could also be used to re-inject solutions into the MOEA's optimization process to help guide the search and to increase diversity. For example, the reduced set of non-dominated solutions could replace solutions that are similar to other solutions in the population. As previously mentioned, it is possible an empty solution set is returned. We aim to address this issue by including an archive weighting technique and returning the solutions with the highest weight if no solutions are selected through overlap. This strategy could also be adapted to allow the end-user to select a specific number of solutions from the archive by selecting the k solutions with the highest overall weight. Furthermore, it also allows for the introduction of user preference through weight vectors if desired.

REFERENCES

[1] S. Chen, R. Wang, L. Ma, Z. Gu, X. Du, and Y. Shao, "A novel many-objective bacterial foraging optimizer based on multi-engine cooperation framework," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and Q. Tang, Eds. Cham: Springer International Publishing, 2018, pp. 520–529.

[2] A. Ramírez, J. R. Romero, and S. Ventura, "A survey of many-objective optimisation in search-based software engineering," *Journal of Systems and Software*, vol. 149, pp. 382–395, 2019.

[3] K. Narukawa and T. Rodemann, "Examining the performance of evolutionary many-objective optimization algorithms on a real-world application," in *2012 Sixth International Conference on Genetic and Evolutionary Computing*. IEEE, 2012, pp. 316–319.

[4] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 641–655.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[6] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.

[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part 1: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[8] R. Fan, L. Wei, X. Li, J. Zhang, and Z. Fan, "Self-adaptive weight vector adjustment strategy for decomposition-based multi-objective differential evolution algorithm," *Soft Computing*, vol. 24, no. 17, pp. 13 179–13 195, 2020.

[9] C. Dai, X. Lei, and X. He, "A decomposition-based evolutionary algorithm with adaptive weight adjustment for many-objective problems," *Soft Computing*, vol. 24, no. 14, pp. 10 597–10 609, 2020.

[10] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of pareto-based moeas: Does non-dominated really mean near to optimal?" in *IEEE Congress on Evolutionary Computation*, vol. 2, 2001, pp. 957–962.

[11] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of moeas," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 5–20.

[12] D. Brockhoff and E. Zitzler, "Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods," in *IEEE Congress on Evolutionary Computation*, 2007, pp. 2086–2093.

[13] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.

[14] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of pareto-optimality for many-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 299–315, 2016.

[15] Q. Lin, S. Liu, K.-C. Wong, M. Gong, C. A. C. Coello, J. Chen, and J. Zhang, "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2018.

[16] F. Yang, L. Xu, X. Chu, and S. Wang, "A new dominance relation based on convergence indicators and niching for many-objective optimization," *Applied Intelligence*, vol. 51, no. 8, pp. 5525–5542, 2021.

[17] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[18] A. Menchaca-Mendez and C. A. Coello Coello, "An alternative hypervolume-based selection mechanism for multi-objective evolutionary algorithms," *Soft Computing*, vol. 21, no. 4, pp. 861–884, 2017.

[19] C. A. Coello Coello, S. González Brambila, J. Figueroa Gamboa, M. G. Castillo Tapia, and R. Hernández Gómez, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 221–236, 2020.

[20] J. D. Knowles, D. W. Corne, and M. Fleischer, "Bounded archiving using the lebesgue measure," in *IEEE Congress on Evolutionary Computation*, vol. 4, 2003, pp. 2490–2497.

[21] A. Britto and A. Pozo, "Using reference points to update the archive of mopso algorithms in many-objective optimization," *Neurocomputing*, vol. 127, pp. 78–87, 2014.

[22] L. Cai, S. Qu, and G. Cheng, "Two-archive method for aggregation-based many-objective optimization," *Information Sciences*, vol. 422, pp. 305–317, 2018.

[23] Q. Zhang, Y. Liu, H. Han, M. Yang, and X. Shu, "Multi-objective particle swarm optimization with multi-archiving strategy," *Scientific Programming*, vol. 2022, 2022.

[24] B. Scheibehenne, R. Greifeneder, and P. M. Todd, "Can There Ever Be Too Many Options? A Meta-Analytic Review of Choice Overload," *Journal of Consumer Research*, vol. 37, no. 3, pp. 409–425, 02 2010.

[25] A. Chernev, U. Böckenholt, and J. Goodman, "Choice overload: A conceptual review and meta-analysis," *Journal of Consumer Psychology*, vol. 25, no. 2, pp. 333–358, 2015.

[26] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, sep 2015.

[27] L. Li, H. Chen, J. Li, N. Jing, and M. Emmerich, "Preference-based evolutionary many-objective optimization for agile satellite mission planning," *IEEE Access*, vol. 6, pp. 40 963–40 978, 2018.

[28] K. Shang, H. Ishibuchi, L. M. Pang, and Y. Nan, "Reference point specification for greedy hypervolume subset selection," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2021, pp. 168–175.

[29] T. Takagi, K. Takadama, and H. Sato, "Non-dominated solution sampling using environmental selection in emo algorithms," in *IEEE Congress on Evolutionary Computation*, 2020, pp. 1–9.

[30] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[31] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[32] C. Xu, "A big-data oriented recommendation method based on multi-objective optimization," *Knowledge-Based Systems*, vol. 177, pp. 11–21, 2019.

[33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *IEEE Congress on Evolutionary Computation*, vol. 1, 2002, pp. 825–830.

[34] W. Chen and K. Tang, "Impact of problem decomposition on cooperative coevolution," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 733–740.

[35] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012.

[36] S. F. Adra and P. J. Fleming, "A diversity management operator for evolutionary many-objective optimisation," in *Evolutionary Multi-Criterion Optimization*. Springer, 2009, p. 81–94.

[37] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[38] A. Peerlinck and J. Sheppard, "Multi-objective factored evolutionary optimization and the multi-objective knapsack problem," in *IEEE Congress on Evolutionary Computation*, 2022, pp. 1–8.