# Optimizing the Descendant-Aware Clustering Parameters

Sukhwan Jung
*Department of Computer Science*
*University of South Alabama*
Mobile, USA
shjung@southalabama.edu

Aviv Segev
*Department of Computer Science*
*University of South Alabama*
Mobile, USA
segev@southalabama.edu

***Abstract*—Topic evolution is a recently introduced field of research as a substitute for a more traditional text-based topic evolution, allowing the tracking of more complex evolutionary events with the use of network structures. Network-based topic evolution showed that the neighborhood characteristics of newly introduced topics can be utilized to determine when a topic would emerge in a given domain. Predicting emerging topics requires a method for generating pseudo-neighbors of previously unseen topics as the neighborhood for an emerging topic is not known before its appearance. The authors proposed the Descendant-Aware Clustering algorithm to generate a set of neighborhood candidates for future emerging topics, surpassing existing algorithms in both performances and computation times. Optimizing the algorithm parameters enhances the performance even further. Significant performance improvements were observed when NSGA-III multi-objective algorithm was applied to over 100 research domains. A set of enhanced default values are introduced to the proposed algorithm removing the necessity for dataset-specific optimization, cementing the position of the Descendant-Aware Clustering as the best clustering algorithm for detecting ancestors of future emerging topics.**

***Keywords—topic evolution, topic prediction, topic emergence prediction, multi-objective optimization, scientometrics***

## I. Introduction

Researchers around the world work tirelessly to introduce new knowledge to their scientific communities, gradually expanding the domain knowledge with cascading contributions. The expansion can be illustrated by an ant colony, where the collective input from individual members leads to an expansion of the community. Materials detrimental to the expansion such as rocks are avoided, while beneficial resources such as water are sought after. Research contributions follow similar principles, focusing efforts on popular research topics with high projected impacts as opposed to the topics with diminishing popularity. It is therefore crucial that researchers know the states of existing research topics before being able to make higher impact contributions. Topic evolution can aid researchers in discovering the current state of research topics by detecting their evolutionary status such as survivability, maturity, and interactions [1].

Topic evolution detects semantical and relational changes between topics over time by analyzing time-specific topic models, which are traditionally text-based models generated from unstructured documents. Topics are extracted as word vectors or embeddings, representing statistical distributions of word or topic co-occurrences. While being successful in capturing how the semantics of a given topic evolves over time, the traditional topic models showed less compatibility when evolution between multiple topics are concerned [2]. This is because the identity of a topic is directly tied to its semantic while a semantic similarity measure is used to *link* topics in different timeslots; one-to-one connection over time tracking a single topic can be done, but it fails to deal with many-to-many connections over time. The amalgamation of identity and semantics also generates another problem that it hampers the *prediction* of new topics emerging in the future, by definition having unique semantics previously unseen in the given dataset. Identities of such topics remain unknown before the related documents are provided to extract their semantics.

A network-based topic model has been proposed to overcome such limitations, detaching the topics' identities from their semantics to enable topic correlation analysis as well as topic emergence predictions [3, 4]. Topic identity is represented by individual nodes within an evolving topic network independent of their semantics, where emerging topics can be viewed as new nodes introduced to the network. As topics rarely are truly novel, it can be assumed that they are not isolated. Newly emerging topics follow the same principle, allowing them to be represented by their neighborhood topics (or *ancestor* topics in previous timeslots). Previous research showed that such representation accurately reflects the identity of emerging topics and *ancestors* of emerging topics in the future can be distinguished from ancestors of existing topics in topic networks [5].

It is infeasible to test all possible subgraph combinations in large networks for their likelihood of being ancestors in the future. Clustering algorithms such as the Advanced Clique Percolation Method (*ACPM*) [6] were proposed for network-based topic emergence prediction in order to generate a set of clusters resembling *ancestors* of emerging topics in the future. Such algorithms reduce the computational intensity of the process by proposing a small set of acceptable candidate groups

without having direct information on their neighborhood membership.

The authors have proposed the Descendant-Aware Clustering (*DAC*) [7] as an enhanced method of detecting such groups by generating overlapping, non-exhaustive clusters specifically tailored to reflect the characteristics of *ancestors* of emerging topics. Comparison between the two algorithms in a wide range of research domains showed that the DAC outperformed not only the general purpose clustering algorithms in the task, but showed improvement over task-specific algorithms such as ACPM as well. The proposed Descendent-Aware Clustering algorithm showed higher positive matches, maintained its performance longer with increasing match threshold, and was shown to be faster and more memory efficient. The performance differences were significant in more than 100 evolving topic networks each generated for a specific research domain, on average with 48k topics and four million links over the ten-year period.

While the Descendant-Aware Clustering showed superior performances, many of the DAC's parameters were not justified in the previous publication with mathematical proofs or empirical evidence and were imported from other related publications. *Structural similarity threshold* and *cluster expansion path length* came from the Loop Edge Delete algorithm [8] where the DAC algorithm was inspired from, for example. Clustering postprocessing parameters such as *maximum number of edges per cluster* and *cluster merging threshold* were fixed to the same values as were used by the ACPM publication [6] to match the experiment conditions as well. This paper is written under the assumption that the initial parameters should be optimal and further performance boost can be expected through a process of optimization. The cluster match precision and recall were the two main performance measures used in the previous research and were optimized. Two measures are often conflicting therefore a multi-objective optimization algorithm is implemented to detect multiple parameter combinations with similar results in combined measures. Experiments were conducted on 103 topic network datasets with distinct histories and research behaviors, which were used in the original research for proposing the Descendant-Aware Clustering. The results showed that the performance improvements by optimized parameters are universal in more than 100 of them. Parameters individually optimized per dataset also led to the discovery of *enhanced default* parameter values, which showed significant performance increase in nearly all datasets.

Section II reviews the related work on network-based topic emergence prediction and multi-objective optimizations. Section III details the optimization problem and the DAC algorithm, and the experiment results are shown in Section IV.

## II. Related Works

### A. Network-based Topic Emergence Detection Methods

Topic evolution is a field of research that aims to automatically track topical changes in a given document collection over time, either with uniformly or irregularly divided timeslots [2]. For each timeslot, a set of topics governing documents within that time period are generated through topic modeling methods [9]. Temporal topic chains are then formed over consecutive timeslots by connecting similar topic models. Depending on the variations in size and interactions, various evolutionary events were detected in the early days of topic evolution [10, 11]. While simpler events such as enlarge and shrink can be detected by analyzing size changes of a single topic chain, more complex evolutionary events such as merge and split involve multiple chains, distinguishing evolution within a single topic and evolution involving multiple topics [12].

The separation of a single-topic and multi-topic evolution led to a computational limitation in topic evolution, as the traditional text-based topic models were not well suited to distinguish between them. Traditional topic models Latent Dirichlet Allocation (LDA) [13] and its variants extract latent semantics from a document set in the form of word-popularity sets based on the word co-occurrence distributions. Topic models are therefore represented as distinct word distributions, which are identified for each document [14]. Topics in LDA-like models are compared by the similarities in word distributions. A more recent approach, word embedding [15], assigns numerical context to words instead of having topics as word distributions and topic similarities are measured in terms of vector similarities [16]. In either case, the topic's identity is directly linked to its semantics. This is problematic for multi-topic evolution detection as semantically *similar* topics are considered to *share* an identity. Tracking the topic chain allows effective detection of *content transition* of a single topic, but cannot effectively track other topics merging into, or split from, that single chain as such semantic similarity is translated as sharing the same topic identity [3]. There are multiple attempts to overcome such limitations, including a two-tiered topic evolution where single-topic and multi-topic are identified in each tier [1]. Time-spanning global topics are retrieved from the whole corpus, representing a set of topics that are present over the whole document collection. Local topics, on the other hand, represent time-specific topics and are extracted from the yearly divided collections instead. The static global topics are matched to a series of dynamic local topics at each timeslot having cosine membership similarities above a given threshold. The number and sizes of matched local topics dictate the evolutionary event of the topic chain represented by the global topic; decreased and increased numbers of local topics connected to a global topic respectively represent the merging and splitting of the topic. While having limited success, such approaches were not applicable in a wider range of datasets due to their inability to adapt to the overall shifts in the topics over time [4].

Network-based topic evolution is proposed to overcome such limitations by separating the topical identity and their semantics and providing a general foundation for a more advanced topic evolution research with topic emergence identification using underlying network patterns [3, 4]. The topics are defined by node structures within a word network [17]. This is based on the assumption that *inventions* and subsequent *innovations* [18] have causal relationships to the components used for the invention and their combinations [19]. Node prediction based on preferential attachment link prediction is proposed to classify whether the nodes in citation networks have a connection to a new node in the future [20], labeling the

new nodes by utilizing the metadata of their neighboring nodes [21]. Similar approaches were also made with multi-layer bibliographic networks for enhancing the performances [3], [22]. A more recent approach attempted topic *emergence* detection by utilizing the structural features of topic ancestors with machine learning models [17]. The emergence of a topic is defined as a binary classifier whether a given topic is newly introduced to the target research field or not, and topics are classified based on observed structural features of their ancestors in previous years. This approach showed high generalizability with high accuracy [4]; however it had limited success on predicting the emerging topics due to the fact that ancestors of a topic can only be discovered *after* a topic is introduced to the network. Such information is unavailable when predicting the future, and use of random subgraphs as candidates is infeasible as the exponential number of possible candidates in larger graphs.

The use of traditional clustering algorithms was largely unsuccessful at detecting accurate *ancestor* candidates, as they are not bound by the traditional cluster characteristics. The main hub of an ancestor group is a single common successor therefore they are not guaranteed to be connected to each other. Some topics could be popular enough to contribute towards multiple emerging topics, so ancestors need to be overlapping, while also non-exhaustive as not all topics contribute towards new topics. Overlapping clustering algorithms such as the Loop Edge Delete (*LED*) can be used for the task, as it is designed to work on large-scale datasets with linear time complexity [8]. Edge removal based on a structural similarity threshold is iteratively applied to divide clusters at each loop, starting from the whole network as a single cluster. Removed edges are then looped through to dictate cluster overlaps. Such algorithms, however, are not designed to identify topic ancestors. The Advanced Clique Percolation Method (*ACPM*) classification algorithm was proposed to identify surging topic correlations by generating clusters with characteristics similar to their ancestors [6]. A novel topic at its embryonic stage is found in the semantic-enhanced topic evolutionary networks, which is represented by its core publications and related author information. Topic clusters with notable recent collaborations are regarded as the ancestors of such novel topics [23]. The pseudo-clique definition of clusters in the original clique percolation method [24] resulted in cluster size disparity problems, and the advanced algorithm tried to overcome this with additional processes including intensity-based clique filtering and neighborhood extraction with local maxima.

### *B. Multi-Objective Optimization*

There are growing interests in the multi-objective optimization (MOO) methods, treating it as a successor to single-objective optimization which can better solve real-world problems which often come with more than one objective at times. Optimization is a prevalent problem in any field where multiple variables are involved in solving an issue; it is essential that optimal solutions are found algorithmically or mathematically for practical purposes [27]. An optimization problem can be formalized as a process for detecting the best fit value $x$ minimizing, or maximizing a given objective function $f(x)$ [28]. The objective function is coupled with a set of predefined input parameters, each with a set minimum and maximum possible values. Inequality or equality constraints can supplement the problem, providing binary conditions within the problem parameters that must be satisfied. There were several attempts to solve multi-objective problems with single-objective optimizations. One approach is to solve each objective as a discrete optimization problem, merging the results using self-adaptive crossovers on the differential evolution algorithm [29]. Another approach merges the objective functions together into a single objective first. The issue in either approach is that they perform poorly when objectives are not positively correlated, showing a conflicting relationship instead [30].

The MOO methods are designed to solve problems with conflicting objectives, producing multiple best solutions if necessary, as more than one combination of input values could result in outcomes of the same quality. This is achieved by determining the Pareto-optimal solutions over the given input space. Constraints in conflicting objectives result in constraint functions having non-linear and non-convex characteristics, therefore more flexible evolutionary algorithms are desired. The nature of conflicting objectives results in non-linear, non-convex constraint functions, leading to the focused efforts in using the evolutionary algorithm for its flexible, derivative-free nature [31]. State-of-the-art algorithms such as the Pareto Archived Evolution Strategy [32], Strength Pareto Evolutionary Algorithm (SPEA2) [33], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [34], and Non-dominated Sorting Genetic Algorithm (NSGA-III) [35] share the similar approaches.

NSGA-III algorithm is based on its predecessor NSGA-II, which is a modified genetic algorithm with changes in the mating and survival phase. In each generation, individuals that fail to surpass others in any of the objective functions are rejected to produce a list of non-dominated candidates. The survival of such individuals is then determined by the Manhattan distance in the objective space, allowing points on either end of the Pareto optimal curve to survive. The outcome is then analyzed to detect unrepresented reference direction, which is a unit simplex representing the partitioned directions each solution can take in the design space. A reference direction unrepresented by any of the surviving solutions is *filled* by allowing the closest discarded solution to survive instead; distance is measured in a line perpendicular to the target reference point crossing over the solution.

## III. Method and Experiments

While the ACPM showed higher performance compared to traditional clustering algorithms including overlapping clustering such as LED, clique-based calculations made the algorithm highly complex, which is not suitable for large networks which topic networks often are. The authors proposed the Descendant-Aware Clustering algorithm [7] based on the understanding that emerging topics can be defined as *emerging technologies* in the bibliometric domain, exhibiting high popularity and interactivity [25]. The algorithm tries to incorporate topic emergence indicators such as *prevalence*, *persistence*, *growth*, and *community* utilized in a recent study [26]. The algorithm works on a top-down approach; structurally dissimilar topics are first disconnected, resulting in a number of connected components each with highly adhesive intra-

connections. The components are then independently expanded and merged if necessary.

The DAC algorithm is run on evolving topic networks. Yearly topic co-occurrence networks are first generated with the *pace of collaboration* [23] as edge weight, where the pace for a given year *y* is calculated as the weighted mean of collaboration power over a set evolutionary window ω. This is to reflect the recent growth rates in each timeslot, which are often associated with emerging topics; harmonic mean is used to calculate collaboration strengths instead of arithmetic mean to reduce the impact of a few hub topics with extreme size and interactions. The pace of collaboration aims to detect recent surges in collaborations, penalizing the earlier half of the time window to produce regression outcomes. Once the topic network is generated, *structural similarities* of each connected topic are calculated based on a ratio of shared neighborhood topics. A binary parameter σ dictates the additional use of weighted edge information; if set to True, the structural similarity is modified by incorporating *weighted vector similarity* and *normalized weight to common neighbor size* as well. Any edge with structural similarity below the structural similarity threshold α is removed, which is calculated by the percentile parameter α'. The resulting list of connected components is expanded by maximizing the PageRank values, adding a neighboring node to a component if an average PageRank is increased by that. The expansion process is a depth-first algorithm with a set path length ε; neighbors of up to length ε can be added to a component as the intermediate topics were already added to it.

The resulting clusters undergo postprocessing to regulate their size and quality; a maximum number of edges per cluster *m* is introduced to represent each cluster with its top *m* edges with the highest weights, while clusters with cosine membership similarity above a cluster merging threshold τ are merged into one. Once the clusters are generated, they are compared against the answer set – actual ancestors for topics emerging *d* years in the future. A cluster and answer pair is set to be a positive match when their membership similarity measure is above a threshold θ. The answer sets as well as the topic networks have been uploaded as python object binary files to the Zenodo repository[1].

Some parameters were not used for optimizations for a variety of reasons. The cluster similarity threshold for positive matches during the optimization process θ' was predetermined to be 0.20 as all tested threshold values θ from 0.01 to 0.75 resulted in superior performances of the DAC algorithm. It is expected that optimized parameters will increase the performance over all threshold ranges. The use of additional weighted edge information yielded inferior performances compared to the unweighted version hence σ was set to False as well. The authors used fixed values on these parameters to reduce the calculation redundancy as their effect is already known. Only the year 2005 has been experimented for the same reason as the DAC algorithm's performances are not sensitive to a given year *y*. While the effect of evolutionary window length ω was not previously analyzed, its default value was unchanged in the experiments as the authors expect some degree of multi-year tracking is necessary for analyzing bibliography-based datasets as seen in other previous research [3, 4]. The year distance to the answer set *d* and the positive match threshold θ are only used after the clustering is finished and a range of values are tested. Table I lists the predetermined parameters.

TABLE I. PREDETERMINED PARAMETERS FOR THE EXPERIMENT.

| Param | Description | Value |
|---|---|---|
| $\theta'$ | Positive match threshold used during optimization | 0.20 |
| $\sigma$ | Use of weighted edges | False |
| $y$ | Year the clusters are found | 2005 |
| $\omega$ | Evolutionary window | 5 |
| $d$ | Year distance parameter | [0, 1, 2, 3] |
| $\theta$ | Positive match threshold | [0.01, 0.02, …, 0.50] |

DAC clusters are overlapping and non-exhaustive, which can be generated from large and dense networks requiring less computational resources compared to other related algorithms. Using the default parameters, previous research [7] showed that all variations of DAC resulted in higher prediction accuracy compared to clusters from Clauset-Newman-Moore, LED, and ACPM with identical cluster postprocessing. It consistently outperformed other algorithms over 100 different datasets, showing that it is not only resource efficient and accurate but also applicable to various research domains without manual interventions. A larger portion of the actual ancestors was matched with DAC's clusters with higher similarities. Predictions were possible for non-consecutive years as well, allowing a possibility of multi-year predictions with three-year predictions showing an average F1 of 0.4685.

TABLE II. DESIGN SPACE PARAMETERS FOR OPTIMIZATION WITH THEIR MINIMUM AND MAXIMUM VALUES.

| Param | Description | Min | Max | Default |
|---|---|---|---|---|
| $\alpha'$ | Structural similarity threshold percentile | 0.80 | 0.99 | [0.90, 0.95] |
| $\varepsilon$ | Cluster expansion length | 1 | 8 | 3 |
| $m$ | Maximum edges per cluster | 5 | 50 | 15 |
| $\tau$ | Cluster merging threshold | 0.5 | 1.0 | 0.7 |

The NSGA-III algorithm is run to optimize the DAC's performance in individual datasets using an implementation from the PyMoo library [36]. The remaining four parameters are used as design parameters, with the minimum and maximum values shown in Table II along with the default value used in the previous research [7]. Precision and recall are selected to represent objective functions as they are two basic accuracy metrics and generally conflict with each other. The purpose of optimization is to identify a set of parameters maximizing both metrics at the same time, which can be used as a dataset-specific default parameter value. As an NSGA-III is implemented to deal with minimization problems, the objective functions are set as the following:

$$f_1 = 1 - precision, \text{ and } f_2 = 1 - recall, \text{ where}$$
$$precision = |matched\ clusters|/|clusters|, \text{ and}$$
$$recall = |matched\ answers|/|answers|. \quad (1)$$

[1] https://zenodo.org/record/5746108

TABLE III. LIST OF NSGA-III PARAMETERS USED IN THE EXPERIMENT.

| Parameters | Value | Description |
|---|---|---|
| ***dim*** | 2 | 2 objective functions |
| ***p*** | 75 | Gaps between consecutive points per objectives |
| ***pop*** | 100 | Populations during evolutionary algorithm |
| ***f_tol*** | 0.0025 | Termination tolerance for objective movements |
| ***nth_gen*** | 3 | Termination tolerance calculation frequency |
| ***n_last*** | 3 | Sliding window for determining termination |
| ***n_max_gen*** | 50 | Fixed generation for termination |

The 2-dimensional objective space is partitioned into 75 reference sections, uniformly generated using the Das and Dennis's structured approach [37]. A total of 76 reference points are selected as a result on the unit simplex ($C_p^{dim+p-1} = C_{75}^{76} = 76$). A population of points per generation is set to be 100 to account for possible variations within each section; a lower population size and number of generations would result in a faster optimization with lower optimization. Table III shows the list of parameters used for the NSGA-III algorithm, including a series of termination-related parameters; they are used to reduce the computational resources used during the experiment while retaining a certain degree of performance. The termination criterion is calculated every three generations using the changes made over them, which can go up to 50 generations as long as better solutions are identified at each stage. The experiment is conducted on a desktop computer with AMD Ryzen 7 2700X 8-core processor (16CPUs, ~3.7GHz) and 16Gb Memory in an attempt to show that the DAC algorithm is efficient enough that its optimization process does not require station-grade computing resources. Multiprocessing is done to fully utilize 16 CPUs, running 16 processes at a time. As the objective functions are not calculated in vectorized matrix operations, starmap interfaces are used to allow distributing solution evaluations[2].

The optimization algorithm is run on 103 topic network datasets, each representing timestamped topic co-occurrences related to a domain topic. The datasets were extracted from the Microsoft Academic Graph dataset as of February 2020, using 103 of the 292 high-level *fields of study* keywords set by Microsoft as domain topics. The resulting datasets had on average 48,467 topics and 3,965,339 topic co-occurrences, with large standard deviations of 18,259.48 and 1,217,796 [7].

## IV. RESULTS

The minimization problem for two objective functions is optimized using the NSGA-III algorithm, which results in multiple non-dominating solutions per each dataset it's run on. Considering all non-dominated solutions, $f_1$ showed lower average values than $f_2$. Fig. 1 shows the objective functions over the 103 datasets, ordered by the best solution with the lowest $f_2$. As the objective functions are inverse of the two actual objectives that were maximized, there are significant differences between the optimized *precision* and *recall* values. Higher *precision* of 0.7135 compared to the average *recall* of 0.6153 indicates that the DAC algorithm at the optimized state produces clusters that are more likely to be a positive match, while its high degree of filtration results in a lower ratio of the matched ancestor groups. The average ratio of the matched clusters remained similar over different predicted years with ANOVA showing insignificant differences in $f_1$ over $d$ with p = 0.7626. The ratio of the matched ancestors however significantly fluctuates with p < 10E-6, as the number of emerging topics and therefore its ancestors are not regularized over different datasets as well as over different timeslots. Their variances remain relatively small and consistent, indicating that the different population sizes between the DAC clusters and actual ancestors are the source of lower *recall* values overall.

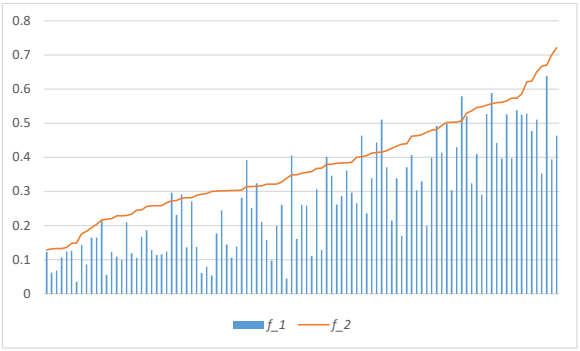
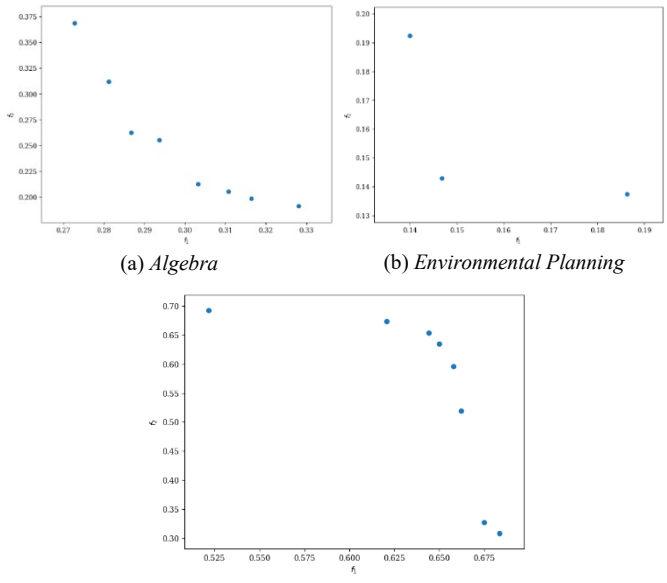



Fig. 1. Average optimized objective function values for 103 datasets, ordered by ascending $f_2$.

Each of the 103 datasets over four $d$ had on average seven solutions, with a variance of 13.25. One iteration of the optimization algorithm took 2026.60 seconds on average, ranging from 77.33 seconds for *environmental ethics* with *d=2* to 9370.54 seconds for *statistics* with *d=1*. The time differences mainly were derived from the size and complexity of different datasets; the time difference over different prediction years $d$ within each dataset was statistically insignificant with p=0.8321. More time doesn't mean higher improvement as the time spent showed weak negative correlations with the optimized model's F1 score (*correl=-0.3605*).



(a) *Algebra* (b) *Environmental Planning*

(c) *Particle physics*

Fig. 2. Three examples of optimized objective spaces when d=0, showing each solution as blue dots.

[2] https://pymoo.org/problems/parallelization.html

Fig. 2 showcases three possible objective spaces with multiple solutions, with an optimal example shown in Fig. 2(a). The *algebra* dataset resulted in a curve conforming to the normal Pareto-front shape, indicating the conflicting nature of two objective functions; an increase in one function decreases another covering similar surface areas. The *environmental planning* domain in Fig. 2(b) pictures a more optimized case with only three solutions; one intermediate solution with two extreme cases. The average of the three solutions resulted in the dataset being the 10th highest F1 score out of 103. A lower number of solutions in the Pareto-optimal shape means the solutions are more clearly presented. On the other hand, several bad outcomes were also observed as showcased in Fig. 2(c). In *particle physics* dataset, significant conflicts were only observed near the extreme values of either objective function, drawing a curve opposite to the normal Pareto-front curve; a small decrease in one function led to a greater increase in another, suggesting that the assumption of conflicting functions are much less obvious in this dataset. Many solutions are therefore not competitive with their rival solutions, especially when the optimization performance is calculated by the F1 score involving multiplication of *precision* and *recall*. This is validated by the fact that while it is ranked 99th for the performance of average solutions, it is the 3rd in terms of performance improvements when the best solution is selected instead of the average of all solutions as shown in Fig. 3. Selecting the best solution based on their F1 score improves the performance of other optimized models over different $d$ as well, further minimizing the objective function values in all but two out of 103 datasets.

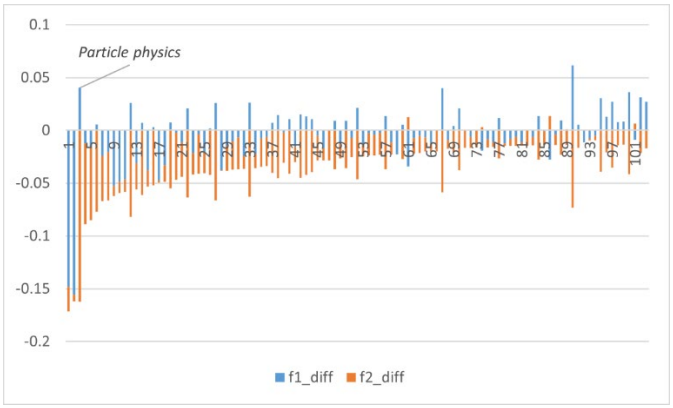



Fig. 3. Differences in $f_1$ and $f_2$ when a solution with the highest F1 score is selected per optimization instead of using an average of all solutions, ordered by ascending sum of the differences.

Using the best solutions with minimum objective functions, the optimized DAC algorithms were able to generate clusters more similar to the ancestors of emerging topics. Fig. 4 shows the performance increase made with the DAC parameter optimization over the 103 datasets with diminishing order of improvements in F1 scores when $d=0$. Precision and recall values are shown as stacked columns, showing that more improvements are made in precision with an average improvement of 0.2357 compared to an average of 0.1418 improvements in the recall. The optimized DAC clusters showed lower $f_1$ in general while more improvements were made in $f_2$ by selecting the best solutions. Combined with these findings, the disparity in the two measures indicates that the optimization process in general favors better true positive detection, and differences between non-dominating solutions are more dependent on the ratio of false positives. The F1 score showed a significant net improvement of 0.1948 on average, which is followed by a lower yet statistically similar ($p=0.7940$) improvement for detecting ancestors of topics emerging up to three years in the future. This performance improvement was the result of optimization using a positive match threshold of $\theta$=0.20, therefore is maximized at that threshold.

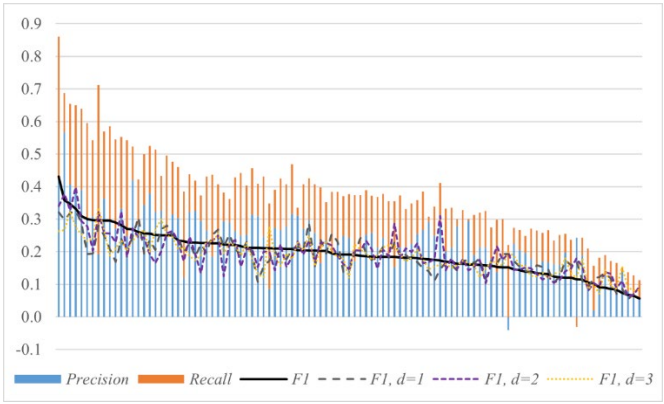



Fig. 4. Improvements in the DAC's performance with the dataset-specific optimized parameters measured by the increases in precision, recall, and F1 scores when d=0. F1 scores for different future years are shown in dotted lines.

The improvements can also be observed over a wide range of threshold values $\theta$ as shown in Fig. 5. Most of the optimized DAC clusters found over 103 datasets showed higher F1 scores compared to the default ones, making a normal distribution shape centered around the $\theta$=0.20 axis. While utilizing one value results in net improvements for most of the cases, the presence of outliers with decreased performances such as *environmental chemistry, cartography,* and *phychoanalysis* indicate that it is important to utilize different values of $\theta$ instead of choosing a single value. Worsening F1 scores indicate that in some cases the clusters found with different level of similarity threshold have distinct properties, and parameters optimized for the best clusters in one $\theta$ is not guaranteed to produce the best clusters in different $\theta$ especially when they are amply distant.

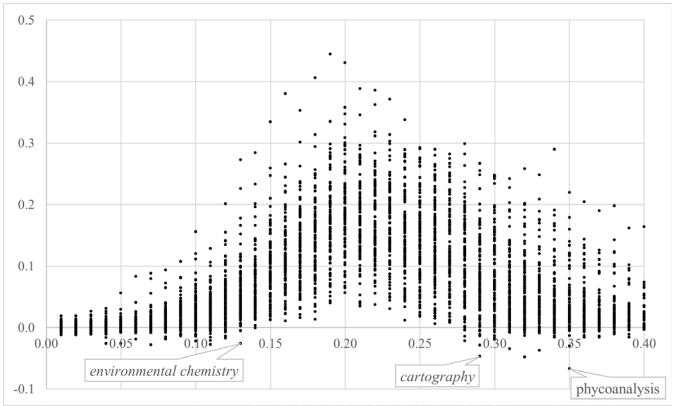



Fig. 5. Improvements in the DAC's F1 scores measured by running MOO on 103 datasets over a range of positive match threshold θ from 0.01 to 0.40, with d=0.

TABLE IV. MEANS AND STANDARD DEVIATIONS OF FOUR DESIGN SPACE PARAMETERS OVER 103 DATASETS.

| *d* | *α′* | | *ε* | | *m* | | *τ* | |
|---|---|---|---|---|---|---|---|---|
| *0* | 0.9534 | 0.03 | 5.09 | 1.82 | 43.40 | 6.85 | 0.7427 | 0.19 |
| *1* | 0.9519 | 0.02 | 5.10 | 1.84 | 44.49 | 6.64 | 0.7138 | 0.18 |
| *2* | 0.9498 | 0.03 | 5.23 | 1.82 | 43.72 | 6.65 | 0.7286 | 0.17 |
| *3* | 0.9550 | 0.02 | 5.41 | 1.88 | 43.95 | 6.40 | 0.7159 | 0.18 |

Analysis of the optimized design spaces revealed commonalities between the best solutions found over 103 different datasets as shown in Table IV. The structural similarity threshold percentile $\alpha'$ is averaged at 0.95, showing the lowest coefficient of variation (CV) with the standard deviation (SD) less than 3% of the mean value, suggesting that 95% of the topic co-occurrences in a topic network do not have essential contributions towards creation of new topics. The maximum number of edges per cluster $m$ and cluster merging thresholds $\tau$ showed larger SD relative to their average with respective CVs of 0.15 and 0.25. Their standard deviations, while having larger variances, are still within a quarter of their average values and are considerably consistent over the datasets. $m$ showed the largest changes from the default value used in the previous experiment, indicating that the original 15 edges per cluster are not fit to represent the ancestors in heavily connected topic networks. The optimized $\tau$ of 0.73 is similar to the default value of 0.7 as the optimized $\alpha'$ matched one of its default values. The length of cluster expansion paths $\varepsilon$ had the largest relative SD with CV=0.35, which is inflated by its integer format and lower average values. Up to five cluster expansion iterations were encouraged by the optimized models.

Optimization of multiple parameters is justified by analyzing correlations between the parameters and the performance metrics. No direct correlation between the three parameters and the performance measures, with $\alpha'$ and $\varepsilon$ having on average less than 0.01 correlation coefficients to *precision*, *recall,* or F1 scores. $m$ showed a more significant negative relationship to F1 with *correl=-0.2324* when *d=0,* while the coefficient diminished down to 0.0292 in the next year. $\tau$ was the only parameter with a consistent correlation coefficient, showing a weak positive correlation to all three measures with coefficients ranging from 0.33 to 0.47 for *d=0,1,2*. It is also positively correlated to the amounts of improvements made by optimizations, albeit with lower coefficients, indicating that the final stage of the DAC is the most crucial.

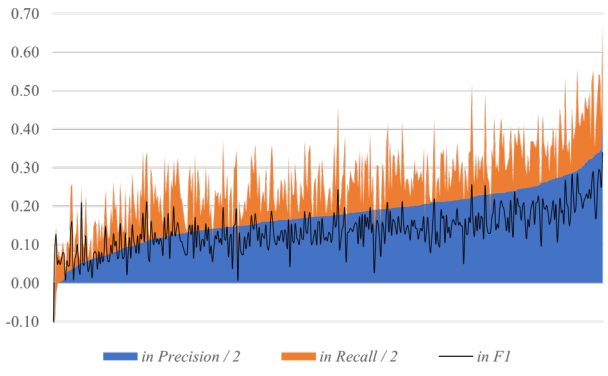



Fig. 6. Improvements in the DAC's precision, recall, and F1 scores by using the enhanced default parameter values instead of the original default values. Precision and recall are stacked, shown in half scales

The experiment on multiple datasets showed that running the multi-object optimization algorithm lead to improvements in the classification performances by capturing the dataset-specific parameters. Based on this result, it is assumed that there would be default DAC parameter values improving performances across the datasets. The mean parameter values in Table IV are therefore considered the enhanced default values, specific to distances to the predicted year which has shown considerable variations. Improvements made by the enhanced default values are shown in Fig. 6, where results over 103 datasets with four *d*s are ranked by improvements in precision. Improvements in precision and recall are shown as a stacked line with half the values for the matching axis with the F1 score. The DAC algorithm showed steady F1 score improvements in all but one iteration (411/412); only *computational chemistry* dataset in *d=1* showed a 0.0993 decrease in the F1 score. As the dataset is ranked 29th on the original results in terms of performance measures, this result can likely be attributed to the randomness of using fixed values over different datasets. Three more results for *astrophysics* (*d=1*), *crystallography* (*d=0*), and *traditional medicine* (*d=3*) showed negative improvements in precision, while still showing net positive F1 score improvements due to their recall values.

The improved results showed an extremely high correlation to the original results with correlation coefficients ranging from 0.9647 to 0.9810, indicating that the new default values provide non-skewed performance improvements across the various datasets. The improved results showed only a moderate correlation (*corr=0.5218 to 0.5421*) to the individually optimized results, suggesting that the improvements are made in a more uniform fashion and the high degree of improvements made in a few datasets are shared by the others. The enhanced defaults, of course, are inferior to the parameters individually optimized for each dataset showing on average 0.06, 0.05, and 0.05 reduction in precision, recall, and F1 scores. This and the previously mentioned outliers were to be expected and deemed not significant enough to diminish the effectiveness of the new default parameters.

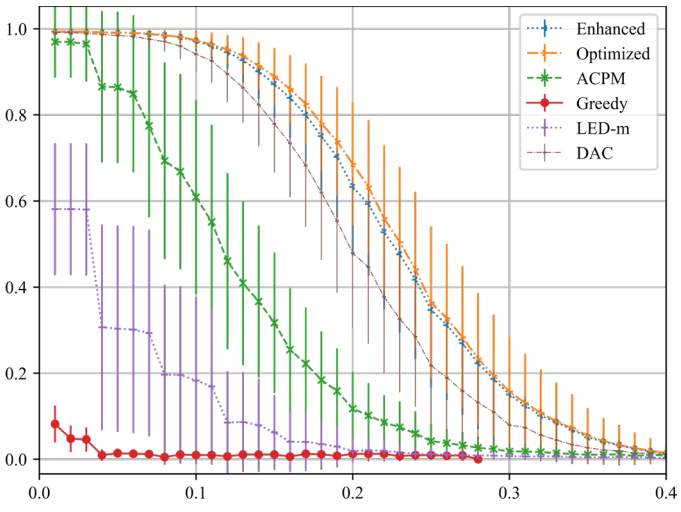



Fig. 7. The F1 score comparison between the DAC algorithm with the original default parameters, DAC with dataset-specific optimization, DAC with the enhanced default parameters, and other existing algorithms.

Fig. 7 visualizes the effects of multi-objective optimization on the DAC's performances in capturing clusters similar to the ancestors of emerging topics, which is compared to three other existing algorithms as implemented in the previous research [7], with the F1 score in the Y-axis and the positive match threshold $\theta$ in the X-axis. The dataset-specific *optimized* parameters and the averaged *enhanced* default parameters were used in comparison to the original *DAC*'s default parameters, both showing significant improvements over the original *DAC* result which already significantly outperformed existing clustering algorithms. They retained their high performance for larger $\theta$ as well, signifying the importance of parameter optimization as well as the general performance of the proposed DAC algorithm.

In summary, the DAC parameter optimization using the NSGA-III algorithm resulted in higher precision and recall across various research domains using the positive matching threshold of $\theta$=0.20. Precisions and recalls over the 103 datasets on average showed 0.2357 and 0.1418 higher values respectively, detecting more true positives and fewer false negatives when clusters sharing more than one-fifth of their members to the true ancestors are marked as positive matches. These results were not tied to the threshold $\theta$ as improvements over the two objectives were consistently observed over a range of $\theta$ as well. The optimization time varied by dataset size and complexity, ranging from 77.33 seconds up to a maximum of 9370.54 seconds. The *enhanced default* is proposed to remove the necessity of dataset-specific optimizations, with $\alpha'$*=0.95,* $\varepsilon$*=5,* $m$*=44,* and $\tau$*=0.7253* as opposed to the original default value of $\alpha'$*=0.95,* $\varepsilon$*=3,* $m$*=15,* and $\tau$*=0.70*.

## V. Conclusion

While the Descendant-Aware Clustering showed superior performance as well as computational simplicity across various domains, the default parameters previously used by the authors were not justified with comprehensive experiment results. The authors considered four parameters are utilized in their optimal capabilities as their default values were extracted from the previous related works. *Structural similarity threshold*, *cluster expansion path length, maximum number of edges per cluster,* and *cluster merging threshold* were optimized by solving a multi-objective optimization problem, maximizing both the precision and recall of the extracted clusters. While various topic network datasets with distinct histories and research behaviors resulted in a number of different solutions, improvements in both precision and recall were observed in nearly all of the 103 experimented datasets. The best solutions in each optimization problem showed consistently higher precision and recall when compared against the DAC with the unoptimized parameters. The F1 score showed a significant improvement of 0.1948 on average over all datasets, with statistically insignificant differences when predicting multiple years into the future. The parameters optimized with a set positive match threshold $\theta$ conserved their performances over a range of $\theta$ as well. Net positive improvements were observed in 88.82% of the instances with $\theta$ from 0.01 to 0.40, drawing a normal distribution centered around 0.20 where the optimization is taken place.

The optimization process was time-consuming when substantially large datasets were involved, as topic networks are heavily connected large networks. There are research domains with significantly larger number of topics and topic co-occurrences such as *computer science* or *medicine*, each making evolving topic networks with 180k/320k nodes and 130/170 million edges. Devices with higher computational resources would reduce the time spent, but the inherent complexity would still require significant investment in computing resources when the Descendent-Aware Clustering parameters are optimized on such large datasets. This led to the consideration of utilizing dataset-specific optimization results to introduce *enhanced default* as dataset-independent default parameter values, removing the necessity for dataset-specific optimization. Comparison between the original and enhanced default values revealed that the number of edges per clusters $m$ showed the largest changes. Relatively smaller differences in $\alpha'$ and $\tau$ suggested that machine-based optimization was the correct course of action as it would have been harder to theoretically distinguish such differences with manual calculations. The *enhanced default* parameters resulted in net F1 score improvement in 411 out of 412 iterations, validating the positive effect of using an optimized parameter set. While higher performances were observed when dataset-specific parameters are used instead, the differences were relatively smaller compared to the overall performance improvement over the original Descendant-Aware Clustering, or other existing clustering algorithms.

Future work would include further improvement on the Descendant-Aware Clustering algorithm as optimizing the parameter values alone resulted in significant changes. Case studies on datasets with deteriorated performances such as *computational chemistry* will be done to analyze the responsible domain characteristics, which can be incorporated into the DAC algorithm to implement a more generic algorithm applicable to a wider range of domains, or propose a behavior-specific algorithm for domains with distinct research behaviors. Multiple solutions could provide more insight into how objective space corresponds to design spaces and the parameters with extreme objective function values could be utilized to allow more detailed parameter configurations according to the user's interests.

## References


[1] B. Chen, S. Tsutsui, Y. Ding, and F. Ma, "Understanding the topic evolution in a scientific domain: An exploratory study for the field of information retrieval," *Journal of Informetrics*, vol. 11, no. 4, pp. 1175–1189, Nov. 2017, doi: 10.1016/j.joi.2017.10.003.

[2] A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou, "Topic Evolution in a Stream of Documents," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, 0 vols., Society for Industrial and Applied Mathematics, 2009, pp. 859–870. doi: 10.1137/1.9781611972795.74.

[3] S. Jung and W. C. Yoon, "An alternative topic model based on Common Interest Authors for topic evolution analysis," *Journal of Informetrics*, vol. 14, no. 3, p. 101040, Aug. 2020, doi: 10.1016/j.joi.2020.101040.

[4] S. Jung and A. Segev, "Analyzing the generalizability of the network-based topic emergence identification method (Accepted)," *Special Issue on Deep Learning and Knowledge Graphs in Semantic Web Journal*, 2021.

[5] S. Jung, R. Datta, and A. Segev, "Identification and Prediction of Emerging Topics through Their Relationships to Existing Topics," in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5078–5087. doi: 10.1109/BigData50022.2020.9378277.

[6] A. A. Salatino, F. Osborne, and E. Motta, "AUGUR: Forecasting the Emergence of New Research Topics," in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, New York, NY, USA, May 2018, pp. 303–312. doi: 10.1145/3197026.3197052.

[7] S. Jung and A. Segev, "DAC: Descendant-Aware Clustering Algorithm for Network-Based Topic Emergence Prediction," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 4043730, Feb. 2022. doi: 10.2139/ssrn.4043730.

[8] T. Ma *et al.*, "LED: A fast overlapping communities detection algorithm based on structural clustering," *Neurocomputing*, vol. 207, pp. 488–500, Sep. 2016, doi: 10.1016/j.neucom.2016.05.020.

[9] Y. Ding, "Community detection: Topological vs. topical," *Journal of Informetrics*, vol. 5, no. 4, pp. 498–514, Spring 2011, doi: 10.1016/j.joi.2011.02.006.

[10] D. M. Blei and J. D. Lafferty, "Dynamic Topic Models," in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, pp. 113–120. doi: 10.1145/1143844.1143859.

[11] A. L. Porter and M. J. Detampel, "Technology opportunities analysis," *Technological Forecasting and Social Change*, vol. 49, no. 3, pp. 237–255, Jul. 1995, doi: 10.1016/0040-1625(95)00022-3.

[12] Q. Mei and C. Zhai, "Discovering Evolutionary Theme Patterns from Text: An Exploration of Temporal Text Mining," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, 2005, pp. 198–207. doi: 10.1145/1081870.1081895.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[14] M. Steyvers and T. Griffiths, "Probabilistic topic models," in *Handbook of latent semantic analysis*, Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 2007, pp. 427–448.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model," p. 19.

[16] O. Levy and Y. Goldberg, "Neural Word Embedding as Implicit Matrix Factorization," p. 9.

[17] S. Jung, T. M. Lai, and A. Segev, "Analyzing Future Nodes in a Knowledge Network," in *2016 IEEE International Congress on Big Data (BigData Congress)*, Jun. 2016, pp. 357–360. doi: 10.1109/BigDataCongress.2016.57.

[18] J. A. Schumpeter, *Business cycles*, vol. 1. Mcgraw-hill New York, 1939.

[19] L. Fleming, "Recombinant Uncertainty in Technological Search," *Management Science*, vol. 47, no. 1, pp. 117–132, Jan. 2001, doi: 10.1287/mnsc.47.1.117.10671.

[20] S. Jung and A. Segev, "Analyzing future communities in growing citation networks," in *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM 2013) International Workshop on Mining Unstructured Big Data using Natural Language Processing*, New York, NY, USA, Oct. 2013, pp. 15–22. doi: 10.1145/2513549.2513553.

[21] S. Jung and A. Segev, "Analyzing future communities in growing citation networks," *Knowledge-Based Systems*, vol. 69, pp. 34–44, Oct. 2014, doi: 10.1016/j.knosys.2014.04.036.

[22] Y. Zhang, M. Wu, W. Miao, L. Huang, and J. Lu, "Bi-layer network analytics: A methodology for characterizing emerging general-purpose technologies," *Journal of Informetrics*, vol. 15, no. 4, p. 101202, Nov. 2021, doi: 10.1016/j.joi.2021.101202.

[23] A. A. Salatino, F. Osborne, and E. Motta, "How are topics born? Understanding the research dynamics preceding the emergence of new areas," *PeerJ Comput. Sci.*, vol. 3, p. e119, Jun. 2017, doi: 10.7717/peerj-cs.119.

[24] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, Art. no. 7043, Jun. 2005, doi: 10.1038/nature03607.

[25] H. Small, K. W. Boyack, and R. Klavans, "Identifying emerging topics in science and technology," *Research Policy*, vol. 43, no. 8, pp. 1450–1467, Oct. 2014, doi: 10.1016/j.respol.2014.02.005.

[26] J. Garner, S. Carley, A. L. Porter, and N. C. Newman, "Technological Emergence Indicators Using Emergence Scoring," in *2017 Portland International Conference on Management of Engineering and Technology (PICMET)*, Jul. 2017, pp. 1–12. doi: 10.23919/PICMET.2017.8125288.

[27] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, 2021.

[28] M. D. Intriligator, *Mathematical optimization and economic theory*. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[29] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1675–1689, Apr. 2013, doi: 10.1016/j.eswa.2012.09.014.

[30] 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, Apr. 2002, doi: 10.1109/4235.996017.

[31] R. Datta, K. Deb, and A. Segev, "A bi-objective hybrid constrained optimization (HyCon) method using a multi-objective and penalty function approach," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, San Sebastián, Spain, Jun. 2017, pp. 317–324. doi: 10.1109/CEC.2017.7969329.

[32] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, Jun. 2000, doi: 10.1162/106365600568167.

[33] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," 2001, doi: 10.3929/ETHZ-A-004284029.

[34] 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, doi: 10.1109/TEVC.2007.892759.

[35] K. Deb *et al.*, *Evolutionary Multi-Criterion Optimization: 10th International Conference, EMO 2019, East Lansing, MI, USA, March 10-13, 2019, Proceedings*. Springer, 2019.

[36] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[37] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998, doi: 10.1137/S1052623496307510.