OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Heuristics for the Maximal Diversity Selection Problem

S.R. Subramanya
[1](*School of Engineering and Computing, National University, San Diego, USA*)

Abstract: *The problem of selecting k items from among a given set of N items such that the 'diversity' among the k items is maximum, is a classical problem with applications in many diverse areas such as forming committees, jury selection, product testing, surveys, plant breeding, ecological preservation, capital investment, etc. A suitably defined distance metric is used to determine the diversity. However, this is a hard problem, and the optimal solution is computationally intractable. In this paper we present the experimental evaluation of two approximation algorithms (heuristics) for the maximal diversity selection problem.*

Keywords: *Maximum diversity selection, Selection problem, Diverse subset, Heuristics*

I. INTRODUCTION

The problem of selecting a subset of elements from a given set, so that the "diversity" among the selected elements is the maximum possible, has been well studied. It arises in numerous applications across highly diverse areas such as in public administration, bio diversity measures, plant breeding, social problems, economics, experimental design, pollution control, ecological preservation, capital investment, curriculum design, the design of VLSI circuits etc. Some representative examples are, the selection of a governing body of an organization from among a given set of nominations, the selection of jury to hear a case in a city, the selection of people from a population to take part in a survey, the selection of persons for testing a product, the selection of companies stocks to invest for a diverse investment portfolio, etc. This is a computationally hard problem, and the optimum solution is practically impossible. Numerous approximation algorithms, probabilistic algorithms, and heuristics have been proposed. We present two heuristics for this problem as applied to an application using a part of the census data, and the experimental results of their implementation.

1.1 Problem statement

Informally, given a set of N elements, and a set of attributes and corresponding values for each of the elements, the problem is to select a subset M ($M < N$) such that the selected elements are as diverse as possible under a given diversity measure. The diversity measure between any two elements is usually a measure of distance defined over the set of attributes of the two elements.

Note that the total number of all possible selections of K items from among N items (which is the number of possible K -element subsets of N elements) is given by:

$$\frac{N!}{k!\,(N-k)!}$$

For each selection, the diversity of the elements needs to be computed, and the best subset (with the maximum diversity) is to be selected. Using the naïve exhaustive search and determining the optimal solution is not practical. Even for relatively moderate sizes of N, the time complexity is enormous. Therefore approximation algorithms, probabilistic algorithms, and heuristics are possible recourses which yield good enough solutions fairly quickly.

First, we will give the terminology and definitions used in the formal problem statement and in the Pseudocode used in the heuristics.

S: Original Set
D: Set of maximally diverse elements selected
N: Number of elements in the set = |S|
K: Number of elements to be selected = |D|
R: Number of relevant attributes of interest of each element
S_{ik}: State or value of attribute k of element i
d_{ij}: Diversity measure between elements i and j
δm_i : Sum of the distances between the median of cluster i and the medians of all the other clusters.

δ_{ik} : Sum of the distances between point *i* of cluster *k* and the medians of all the other clusters.
Δ : Sum of the pairwise distances between all pairs of the medians of the clusters

The diversity between two elements may be defined to be a normalized distance between them. There have been several distance measures defined, and the choice of a distance measure depends on the application. One of the most commonly used distance measure is the Euclidean distance. For any two elements *i* and *j*, with vectors of attribute states (S_{i1}, S_{i2}, …… S_{iR}) and (S_{j1}, S_{j2}, …… S_{jR}), respectively, the Euclidean distance between them d_{ij} is given by:

$$d_{ij} = \sqrt{\sum_{k=1}^{R} (S_{ik} - S_{jk})^2}$$

A measure of the diversity of a set of *N* elements could be the sum of the Euclidean distances between each distinct pair of elements, which is:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}$$

Letting x_i = 1 if element *i* is selected and 0 otherwise, the maximum diversity problem can then be formulated as the following quadratic zero–one integer program.
Maximize

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}\, x_i x_j$$

subject to

$$\sum_{i=1}^{N} x_i = K, \quad x_i \in \{0,1\}, 1 \leq i \leq N$$

The general maximum diversity problem is intractable [1]. Thus, the optimal solution for the problem, even for moderate problem sizes, is beyond reach, and it is unlikely to find an algorithm for the optimal solution computation in reasonable (polynomial) time. Therefore, numerous heuristics and approximation algorithms have been proposed in the literature.

In the next section we present some background and work related to maximum diversity selection problem. Section 3 presents the two proposed heuristics. Section 4 describes the results of the implementation of the heuristics and their performance. This is followed by conclusions.

II. BACKGROUND AND RELATED WORK

The maximal diversity problem (MDP) was introduced by [2]. Considerable work has been done in the solution of the maximum diversity selection (MDS) and related problems, many of them being approximation algorithms and heuristics. We give below a few representative ones.

The formulation of the maximum diversity selection problem as a quadratic zero-one model is given in [3]. It also presents situations and their handling where some attributes take precedence over others in the measurement of diversity. Five heuristics to find groups of students with the most diverse characteristics such as nationality, age, and graduation level, their implementation and testing on real data is presented in [4].

An iterative technique, called a greedy, randomized, adaptive search procedure (GRASP) has been proposed in [5]. Each iteration in the process consists of two phases namely, construction and local search. A feasible solution is built in the construction phase, and exploration of the neighborhood is performed by the local search. The application of this technique to the maximal diversity selection problem has been the subject of numerous studies, with implementations, and some examples are [6, 7, 8]. A Tabu search-based solution to the MDP is presented in [9].

Conceptual issues such as qualitative versus quantitative diversity and choice of an index suitable for describing the degree of inhomogeneity or diversity in a group of people or computer agents is discussed in [10]. It also addresses an inverse problem, namely, given a pre–specified degree of inhomogeneity, to determine the set of numbers having the desired degree of inhomogeneity.

A practical application in the manufacturing domain occurs where the number of different configurations of a given part far exceeds the acceptable number on the assembly line. In these situations, an optimal number of configurations is to be selected for manufacturing such that any required non-produced configuration can be replaced by the cheapest produced one that is compatible with it. This is referred to as the

Optimal Diversity Management Problem (ODMP) which is given in [11]. In [12], the use of Lagrangean relaxation to reduce the size of the problem is presented in order to be able to solve it to optimality via classical integer optimization.

An exact algorithm using a network flow approach for the maximum diversity selection problem as applied to finding diverse working groups for a graduate course is presented in [13]. A variant called Reactive GRASP, for which a parameter used in the construction phase is self-adjusted for each iteration is proposed in [14], where its application to TDMA traffic assignment is presented.

A review of all the heuristics and metaheuristics for finding near-optimal solutions for the maximum diversity problem (MDP) is given in [15]. It also gives a new benchmark library MDPLIB, which includes most instances previously used for this problem, as well as new ones. It presents comparison of extensive computational experiments of 10 heuristics and 20 metaheuristics for the MDP. Non-parametric statistical tests are reported in their study to draw significant conclusions.

Location of facilities according to distance, accessibility or impacts is given in [16]. Diversity maximization which is crucial for establishing viable ecological systems is presented in [17]. A metaheuristic framework called scatter search that explores solution spaces by evolving a set of reference points is presented in [18]. It is shown to outperform the best approximation procedures reported in [8, 9].

For ambiguous or broad queries over large data sets, it is desirable to achieve "diversification" of query results. The queries might have different contexts, domains, and applications, and without diversification of results, results that could be of use might be missed out. The diversification also reduces redundancy by showing more information, thus increasing both the effectiveness of the query as well as user satisfaction across a wide variety of users. In this context, the several approximation algorithms have been impractical on large data sets. A streaming-based approach which processes items incrementally, maintaining a near-optimal diverse set at any point in time is proposed in [19]. This approach has been shown to have linear computation and constant memory complexity with respect to input set size, and effective for streaming applications. A general machine learning approach to predicting diverse subsets, which is shown to make predictions in linear time and training time that scales linearly in the number of queries is presented in [20].

In drug discovery process, a subset selection scheme that minimizes the overall compound similarity which ensures a wider coverage in terms of compound diversity has been used. A subset can be selected on the basis of target-specific, gene family specific or chemical diversity for wider coverage of chemical series. A metric that can be used as a measure of the diversity of subsets in the drug discovery domain is defined in [21]. It also presents a combinatorial optimization algorithm for selecting or partitioning the collections of subsets.

Algorithms for the diversity problem in the design-variable space of a multi-objective problem (maximum design diversity) are presented in [22]. It is applicable in product design, where it is competitively advantageous to develop a comprehensive list of the most diverse architectures spanning the design space, and to be able to evaluate them to find those with the greatest real value. It also presents a real-world mixed-integer non-linear programming model with two objectives, dozens of variables and hundreds of constraints taken from the domain of aircraft-engine design.

A generic formulation of the core problem of optimal biodiversity preservation under a budget constraint, called the Noah's Ark Problem is presented in [23]. It addresses the main question of how to determine basic priorities for maintaining or increasing diversity. It develops a cost-effectiveness criterion which can be used to rank priorities among biodiversity-preserving projects. Two restricted, but realistic scenarios of the Noah's Ark Problem (allowing for variable conservation costs and uncertain survival of the taxa) for which a greedy algorithm is guaranteed to produce optimal solutions are presented in [24].

III. PROPOSED HEURISTICS

In this section, we present a description of the two heuristics, the Pseudocodes, and the time complexity analyses. Both of the heuristics are based on iterative improvement of the solution. In the descriptions, the terms "elements" and "points" are used synonymously. In the first heuristic, an initial set of K elements which are (approximately) farthest from one another is derived from the original set. Then, the elements of the set are iteratively improved. The improvement continues until no more significant improvement is reported by the heuristic. In the second heuristic, the well–known K–means algorithm is applied to the initial set of points (elements) to derive K clusters. It then proceeds to find iteratively an element in each of the K clusters such that the sum of the distances between those elements is as large as possible. When no more improvement is possible, the K elements are taken as the diverse set of elements (points).

3.1 Heuristics-I

In this heuristic, an initial set of K (approximately) farthest elements from out of the initial set S of N elements is derived, and the process is shown in the flowchart in Fig. 1. First, the element which is farthest from all of the remaining $(N - 1)$ elements of S is determined. It is then removed and placed in the set D, the set of maximally diverse elements (which is initially empty). Subsequently, the element in S whose total distance from the elements of D is the maximum, is removed from S and placed in D. This process continues until the number of elements in D reaches K. This is taken as the initial set of the most diverse K elements.

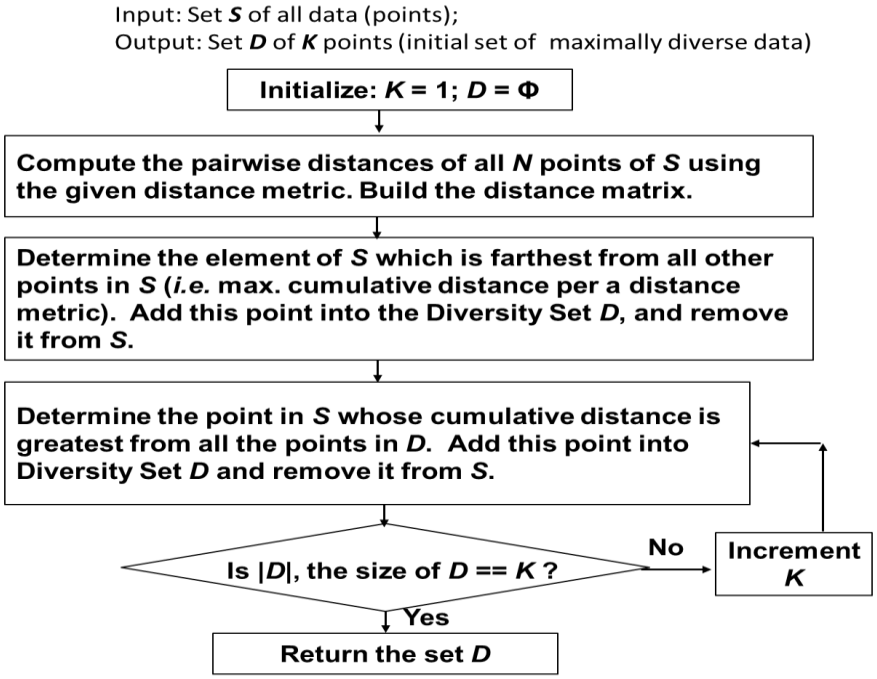

Figure 1. Deriving the initial set of K (approximately) farthest elements

The next stage of the heuristic iteratively improves the elements in the set D. The outline of this iterative improvement process is shown in Fig. 2. The iterative improvement is necessary since the initial set D is may not contain the maximally diverse elements. This is because, while selecting the element in S to be included in D, the element with the largest of the sum of its distances to the elements in D is selected. Thus, it is not considering the it's sum of the distances to other elements of S, and there are possibilities that points which are farther apart in S than are points in D may be missed out to be selected for inclusion in D.

The iterative improvement process starts by computing DC, the sum of the pairwise distances of the K points in D. Then, each point j in S is selected, and it is substituted for a point m selected in D. Its sum of the pairwise distances with $(K - 1)$ points in D (all the original points in D except point m replaced by point j from S) is computed. This is done in K times, going round all the points in D. The maximum sum of the "new" pairwise distances obtained by replacing each point of D with the given point of S is compared with DC. If it is greater than DC, then the points m in D and j in S are swapped. Thus after this iteration, the sum of pairwise distances of the elements of D would be greater than before the iteration, and thus there is improvement. This is repeated for all points in S, keeping track of whether there has been a swap of points in D and S. If there has been no swap, then there has been no improvement and the set D is taken to be the maximally diverse set of elements.

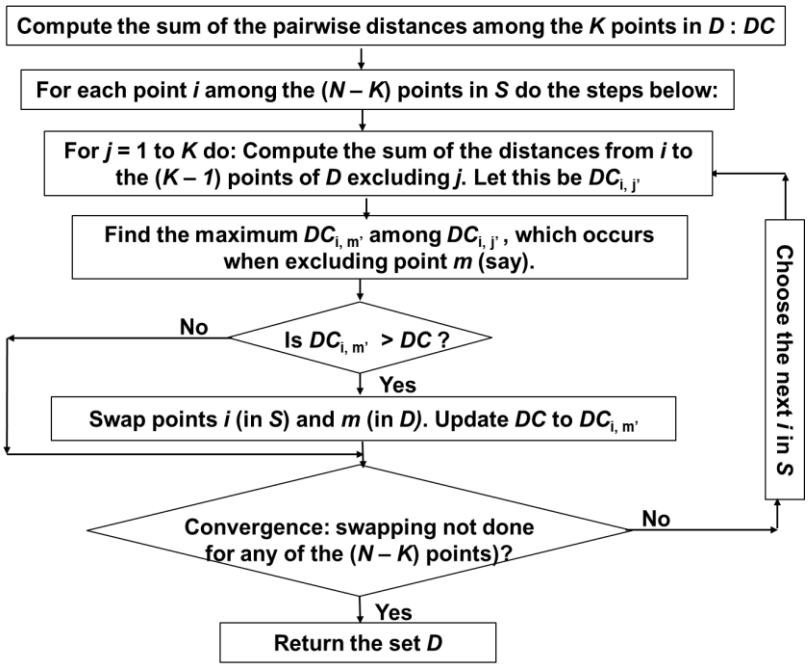

Figure 2. The iterative process of heuristics-I to improve the distances (diversity) of the elements of set *D*

3.1.1 Pseudocode

The Pseudocode for the derivation of the initial set *D* is given below. The starting point is the given set *S* of *N* elements and their pairwise distances, and the output is the initial set *D* of *K* (approximate) diverse elements.

Algorithm ***InitDiversitySet***
(Input: *S (set of N elements)*; output: *D* (*max. diversity subset of K elements of S*))
begin

1. Find the sum of distances of every element (point) of *S* from every other element of *S*.
2. Determine the element which is farthest from all the others (one with the largest sum of the distances). Add this element into the Diversity Set *D*, and remove it from *S*.
3. Identify the next element of *D* by calculating the element in *S* whose sum of the distances from the points currently in *D* is the greatest. Add this point into Diversity Set *D* and remove it from *S*.
4. Repeat step 2 until the size of *D* reaches *K* elements.

end

Algorithm ***IterativeImprovement***
(Input: *S (set of N – K elements), D(initial diversity subset of K elements)*; output: *D* (*improved diversity subset of K elements*))

begin

1. Compute *DC*, the pairwise distances among the *K* elements of *D*. distances of every element (point) of *S* from every other element of *S*.
2. For i = 1 to ($N – K$) over the elements of *S* do steps 3 – 6.
   2.1. For j = 1 to *K* over the elements of *D* do step 2.1.1.
      2.1.1. Compute the sum of the distances from i to all the elements of *D*, excluding j. Let this be $DC_{i,j'}$.
   2.2. Find the maximum of $DC_{i,j'}$ for all j. Let this be $DC_{i,m'}$, occurring when point m is excluded.
   2.3 If $DC_{i,m'} > DC$, swap i (in S) and m (in *D*). Replace *DC* by $DC_{i,m'}$.
3. Check if there was no swap done at all in the ($N – K$) iterations in step 2. If not go to step 2. Otherwise go to step 4.
4. There has been convergence. Return *D* as the diversity set.

end

3.1.2 Analysis

Consider first the derivation of the initial set given in the algorithm `InitDiversitySet`. It is easy to see that step 1 takes $N(N-1)/2 = O(N^2)$ time. Step 2 is essentially finding the maximum among N values, which takes $O(N)$ time. In steps 3 and 4, at any step i, the sum of the distances of each of the $(N-i)$ elements of S with the i elements of D is computed. Steps 3 and 4 are performed $(K-1)$ times. Thus, the time taken by steps 3 and 4 is: $(N-1) + 2(N-2) + 3(N-3) + \ldots\ldots (K-1)(N-K+1)$

$= [N + 2N + 3N + \ldots\ldots (K-1)N] - [1 + 2^2 + 3^2 + \ldots\ldots + (K-1)^2]$

$= N[1 + 2 + 3 + \ldots\ldots + (K-1)] - [1 + 2^2 + 3^2 + \ldots\ldots + (K-1)^2]$

$= N K(K-1)/2 - K(K-1)(2K-1)/6$

which is $O(NK^2 - K^3)$.

Thus, the complexity of `InitDiversitySet` is $O(N^2) + O(N) + O(NK^2 - K^3)$. Assuming that the pairwise distances of the N elements are given, the $O(N^2)$ computations are not necessary, and the overall complexity would be $O(NK^2 - K^3)$.

For the `IterativeImprovement`, step 1 takes $K(K-1)/2 = O(K^2)$ time. Step 2 is essentially a loop going over $(N-K)$ times. The inner loop at step 2.1 goes over K times, and step 2.1.1 within the inner loop takes time $(K-1)$. Thus the steps 2 – 2.1.1 take $(N-K)K(K-1) = O(NK^2)$ time. Step 2.2 takes K units of time. Thus the overall complexity of one iteration of the `IterativeImprovement` is $O(K^2) + O(NK^2) + O(K) = O(NK^2)$. Suppose there are I iterations. Therefore, the complexity of `IterativeImprovement` is $O(NK^2I)$.

3.2 Heuristics-II

An outline of this heuristics is shown in Fig. 3. In this heuristics, the well–known K–means clustering algorithm is applied on the initial set S of N elements to derive an initial set of K clusters. Then, the elements of the set are iteratively improved. The improvement continues until no more significant improvement is reported by the heuristic.

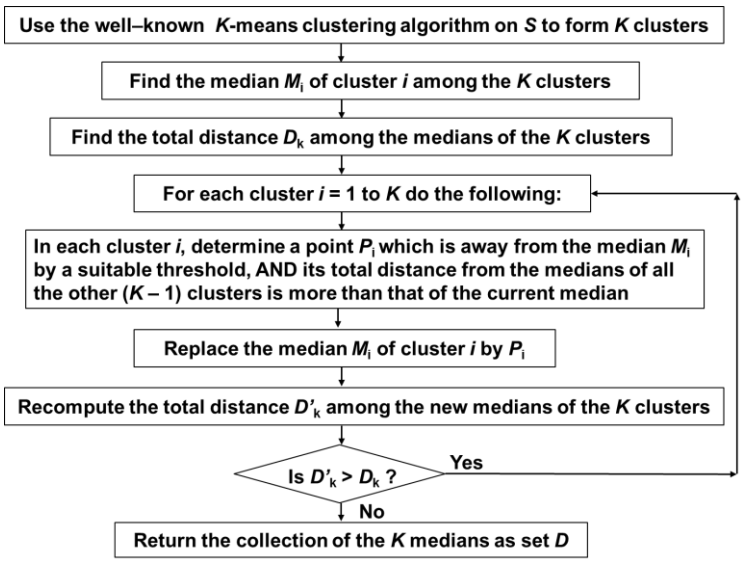

Figure 3. Outline of heuristics-II and iterative improvement based on K-means clustering

The iterative improvement consists of the following steps. The medians (centroids) of the clusters are determined, and the sum of the pairwise distances among the medians D_k is computed. Then, for each cluster i, a point P_j which is sufficiently away from the median (beyond a threshold), is chosen, and the sum of the pairwise distances between all the other medians and the selected point is computed. If this distance is more than the sum of the pairwise distances of the current median M_i with the rest of the medians, then the median M_i is replaced with the point P_j. This is repeated for all clusters. Then the sum of the pairwise distances of the (possibly) new set of medians of the clusters D'_k is computed. If D'_k is greater than D_k, it indicates that there has been an improvement. The above process is repeated. When D'_k is no longer greater than D_k, it indicates further

improvements are highly unlikely. The heuristic stops and returns the medians of the K clusters as the diverse set of K elements.

3.2.1 Pseudocode

The Pseudocode for the heuristics using the K–means clustering algorithm is given below. At the start the K–means clustering algorithm is applied to the given set S of N elements (points) to derive K clusters. Then an iterative improvement process is applied on the K clusters to derive one element from each cluster such that the collection of the K elements is expected to be as diverse as possible.

Algorithm ***K-Means-Based-Diversity-Set***
(Input: *S (set of N elements)*; output: *D* (*max. diversity subset of K elements of S*))

begin

1. Apply the K–means clustering algorithm on the given set S of N elements.
2. Determine the median element (point) M_i for each of the clusters $1 \leq I \leq K$.
3. Compute Δ, the sum of pairwise distances between the medians of the K clusters.
4. For each cluster $i = 1$ to K do the following:
   4.1. Determine a point P_r which has not already been considered and away from the current median beyond a threshold.
   4.2. Compute δ_{ri}, the total distance between P_r (of cluster i) and the other (K–1) medians M_j,$1 \leq j \leq K$, $i \neq j$.
   4.3. If $\delta_{ri} > \delta m_i$, then replace the median of cluster i with the point P_r.
5. Compute Δ', the sum of pairwise distances between the 'new' medians of the K clusters.
6. If $\Delta' > \Delta$ (there has been improvement), go to step 4. Otherwise, stop.

end

3.2.2 Analysis

The K-means clustering takes O($NKID$) time, where N is the number of points, K is the number of clusters, I is the number of iterations, and D is the number of attributes. Step 2 works on each of the K clusters. Assuming that on the average, each cluster has N/K elements, finding the median element takes $(N/K)^2$. Therefore, step 2 requires $O(K\ (N/K)^2) = O(N^2/K)$ time. Step 3 takes $K(K-1)/2 = O(K^2)$ time. The loop in step 4 is done K times. Step 4.1 takes O(1) time, 4.2 takes $(K-1)$ time, and 4.3 takes O(1) time. Thus steps 4 – 4.3 take $O(K^2)$ time. Step 5 takes $K(K-1)/2 = O(K^2)$ time. Thus, the complexity of steps 2–5 is $O(N^2/K + K^2)$ time. Assuming I iterations of improvement, the overall complexity of `K-Means-Based-Diversity-Set` $O(I(N^2/K + K^2))$.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed heuristics I and II given Sections 3.1 and 3.2 have been implemented in Python, and leveraged two core Python modules for mathematics and scientific calculations: Numpy and Scipy. Specifically, the initial clustering for the K–means algorithm was done using the kmeans2 algorithm provided by the Scipy.cluster.vq module.

The data used in the experiments is a publicly available dataset from census.gov describing 2012 average poverty percentage rates and median incomes of 3141 US Counties. The objective was to determine 50 counties with the most diverse (a) median incomes and (b) average poverty percentages. The experiment used scipy.cluster.vq.kmeans2 to identify 50 cluster centroids for Heuristic–II, and then analyzed each cluster to locate the data point closest to this centroid.

4.1 Experimental results

The results of improvements in the diversity measures starting from an initial set and using the iterative improvements of the proposed heuristics on the census data of (a) median incomes (left) and (b) average poverty percentages (right) are shown in Fig. 4. It is seen that in both cases, the heuristics give improvements in the diversity values. The improvements are around 10% of the initial set of values, where the initial values themselves are derived using certain algorithms (as outlined in sections 3.1 and 3.2).

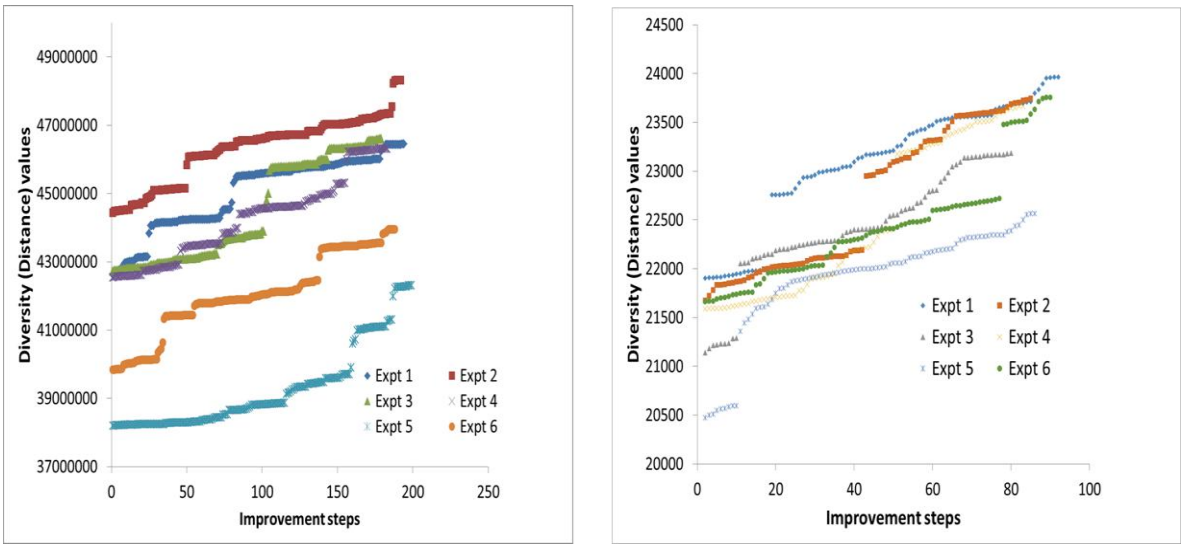

Figure 4. Improvements in the diversity measures using the iterative improvements of the proposed heuristics

Figure 5 shows a heat map of the 50 counties with (a) the most diverse median income (left) and (b) the most diverse poverty percentage (right).

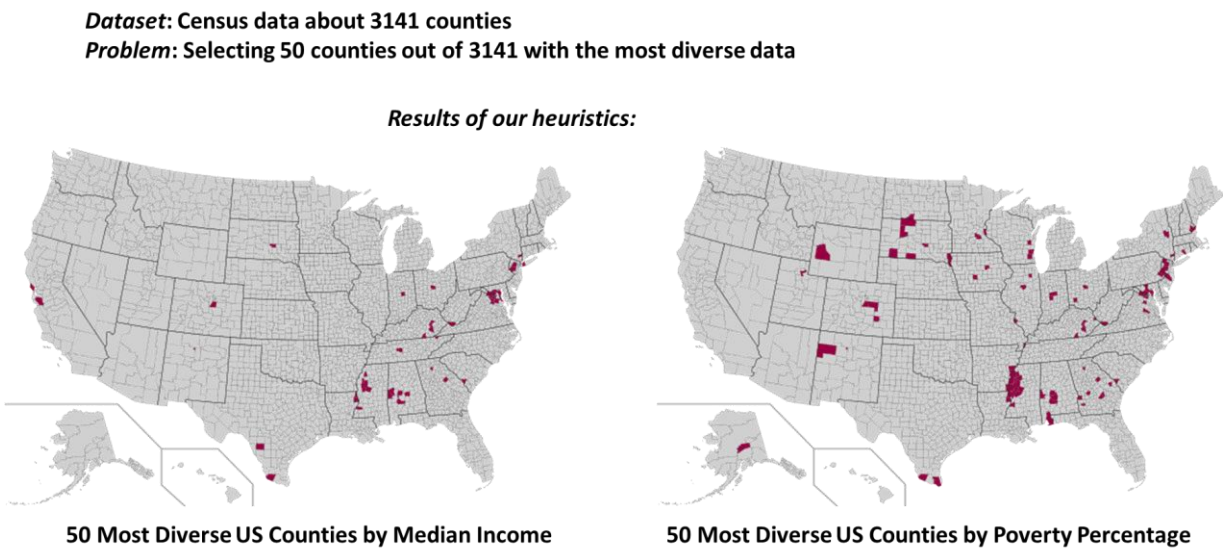

Figure 5. Heat maps of the results of selection of 50 out of 3141 counties with the most diverse of the specified attributes

V. CONCLUSION

There have been numerous studies related to the problem of selecting a subset of elements from a given set, so that the "diversity" among the selected elements is the maximum possible. It arises in numerous applications across highly diverse areas such as in public administration, bio diversity measures, plant breeding, social problems, economics, experimental design, pollution control, ecological preservation, capital investment, curriculum design, etc. The optimal solution to this problem is intractable, and approximation algorithms and heuristic are sought after. This paper proposed two heuristics and presented the results of implementing them on a part of the US census data related to median income and average poverty percentages in the counties. The complexities of the heuristics are acceptable and the running times are reasonable. The iterative improvement processes provide around 10% of improvement to the initial diversity values.

Acknowledgements

The author wishes to thank Mr. Timothy Wade, Sr. Security Integration Engineer at Leidos, for discussions of the heuristics, implementation of the heuristics, and running of the experiments.

REFERENCES

[1] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP–Completeness, W.H. Freeman, 1979.

[2] F. Glover, G. Hersh, and C. McMillan, Selecting Subsets of Maximum Diversity, MS/IS Report No. 77–9, University of Colorado–Boulder, 1977.

[3] C-C. Kuo, F. Glover, and K.S. Dhir, Analyzing and Modeling the Maximum Diversity Problem by Zero-One Programming, Decision Sciences, Vol. 24, No. 6, (1993), 1171–1185.

[4] R. Weitz and S. Lakshminarayanan, An Empirical Comparison of Heuristic Methods for Creating Maximal Diversity Groups, J. Operational Research Society, Vol. 49 (1998), 635–646.

[5] T.A. Feo and M.G.C. Resende, Greedy Randomized Adaptive Search Procedures, J. Global Optimization, Vol. 6, (1995), 109–133.

[6] P. M. D. Andrade, A. Plastino, L. S. Ochi, and S. L. Martins, GRASP for the Maximum Diversity Problem, Proceedings of the Fifth Metaheuristics International Conference, 2003.

[7] J.B. Ghosh, Computational aspects of the maximum diversity problem, Operations Research Letters archive, Vol. 19, Issue 4, October, 1996, 175–181.

[8] G. C. Silva, L. S. Ochi, and S. L. Martins. Experimental Comparison of Greedy Randomized Adaptive Search Procedures for the Maximum Diversity Problem, Lectures Notes on Computer Science, Vol. 3059, Springer, 2004, 498–512.

[9] A. Duarte and R. Martí, Tabu Search and GRASP for the Maximum Diversity Problem, European Journal of Operational Research, 178, (2007), 71–84.

[10] R. Warren, Designing Maximally, or Otherwise, Diverse Teams: Group-Diversity Indexes for Testing Computational Models of Cultural and Other Social-Group Dynamics, Association for the Advancement of Artificial Intelligence, 2009, 78–85.

[11] U. W. Thonemann and M. L. Brandeau, Optimal commonality in component design, Operations Research, Vol. 48, (2000), 1–19.

[12] O. Brient and E. Naddef, The Optimal Diversity Management Problem, Operations Research, Vol. 52, No. 4, July–August 2004, 515–526.

[13] J. Bhadury, E. Joy Mighty, and H. Damar, Maximizing Workforce Diversity in Project Teams: a Network Flow Approach, Omega, Vol. 28 (2000), 143–153.

[14] M. Prais and C.C. Ribeiro, Reactive GRASP: an application to a matrix decomposition problem in TDMA traffic assignment, INFORMS Journal on Computing, Vol. 12 (2000), 164–176.

[15] R. Marti, et. al., Heuristics and Metaheuristics for the Maximum Diversity Problem, Journal of Heuristics, Vol. 19, Issue 4, (August 2013), 591–615.

[16] E. Erkut and S. Neuman, Comparison of four models for dispersing facilities, Canadian Journal of Operational Research and Information Processing 29 (1991), 68–86.

[17] D. Pearce, Economics and genetic diversity". Future 19(6), (1987) 710–712.

[18] M. Gallego, A. Duarte, M. Laguna, and R. Martí, Hybrid Heuristics for the Maximum Diversity Problem, Computational Optimization and Applications, vol. 44, Springer (2009), 411–426.

[19] E. Minack, W. Siberski, and W. Nejdl, Incremental Diversification for Very Large Sets: a Streaming-based Approach, SIGIR'11, July 24–28, 2011, Beijing, China, 585–594.

[20] Y. Yue and T. Joachims, Predicting Diverse Subsets Using Structural SVMs, Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008.

[21] H. Zhu, J. Klug-McLeod, and G. A. Bakken, A Heuristic Algorithm for Plate Selection That Maximizes Compound Diversity, Croat. Chem. Acta 86 (4) (2013) 435–441.

[22] A. Zadorojniya, et.al., Algorithms for Finding Maximum Diversity of Design Variables in Multi-Objective Optimization, Procedia Computer Science 8 (2012) 171–176.

[23] M. L. Weitzman, The Noah's Ark Problem, Econometrica, Vol. 66, No. 6 (Nov. 1998), 1279–1298.

[24] K. Hartmann and M. Steel, Maximizing Phylogenetic Diversity in Biodiversity Conservation: Greedy Solutions to the Noah's Ark Problem, Systematic Biology 55(4), 2006, 644–651.