Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Cluster Detection with the PYRAMID Algorithm

Samir Tout[1], William Sverdlik[2], and Junping Sun[3]

***Abstract*—As databases continue to grow in size, efficient and effective clustering algorithms play a paramount role in data mining applications. Practical clustering faces several challenges including: identifying clusters of arbitrary shapes, sensitivity to the order of input, dynamic determination of the number of clusters, outlier handling, processing speed of massive data sets, handling higher dimensions, and dependence on user-supplied parameters. Many studies have addressed one or more of these challenges. PYRAMID, or parallel hybrid clustering using genetic programming and multi-objective fitness with density, is an algorithm that we introduced in a previous research, which addresses some of the above challenges. While leaving significant challenges for future work, such as handling higher dimensions, PYRAMID employs a combination of data parallelism, a form of genetic programming, and a multi-objective density-based fitness function in the context of clustering. This study adds to our previous research by exploring the detection capability of PYRAMID against a challenging dataset and evaluating its independence on user supplied parameters.**

Keywords: Data Mining, Clustering, Genetic Programming, Parallelism, Density.

I. INTRODUCTION

CLUSTERING is a technique used to unravel data distributions and patterns in a dataset by means of a supervised [3] or unsupervised [6] classification of those patterns. Example clustering applications include multimedia analysis and retrieval [7], pattern recognition [10], and bioinformatics [3].

Research continues in the field of clustering, involving numerous disciplines. Many existing algorithms make certain assumptions about the underlying hypothesis space (i.e. bias): such assumptions, once placed into practice, may prove erroneous. For example, traditional centroid based clustering algorithms yield solutions that are a collection of convex sets. Such algorithms fail on non-convex boundaries. Other algorithms require user input parameters to initiate processing, as in k-means clustering. These observations motivate avenues for further research and introduce new challenges that become core issues to be addressed by clustering algorithms.

In [18], we introduced PYRAMID: *Parallel hYbrid clusteRing using genetic progrAmming and Multi-objective fItness with Density*. In its present form, PYRAMID employs a combination of data parallelism, genetic programming (GP), special operators, and multi-objective density-based fitness function in the context of clustering to resolve most of the above challenges. The data space is divided into cells that become the target of clustering thus eliminating dependence on the order of data input. Data parallelism is used to achieve speedup. The algorithm divides the data set onto multiple processors each of which executes a genetic program that uses a flexible individual representation that can represent arbitrary shaped clusters. The genetic program also utilizes a density-based fitness function that helps avoid outliers. The experiments in [18] have shown positive results. The datasets used therein were characterized by various sizes and irregular cluster shapes. They were used to compare cluster and outlier detection between PYRAMID and existing renowned algorithms such as BIRCH [20], CURE [5], DBSCAN [4], and NOCEA [13]. This paper is a continuation of [18]. It conducts experiments using another dataset that was employed in [15] featuring different challenges, as described in Section 5. It also explores the independence of PYRAMID on the user supplied parameters.

The rest of this paper is organized as follows. Most sections borrow from [18]. Section 2 enumerates some related work. Section 3 elicits some key concepts that play an essential role in this study. Section 4 elaborates on the PYRAMID approach. Section 5 provides details about the experiments from [18] as well as this study. Finally, Section 6 states the conclusion of this research and future directions.

II. RELATED WORK

Many algorithms have been developed previously to address some of the above mentioned issues, but each of those concentrated on a specific aspect.

[1] Keane, Inc., 24901 Northwestern Hwy, Southfield, MI 48075 and Department of Computer Science, Eastern Michigan University, Ypsilanti, MI, 48197
[2] Department of Computer Science, Eastern Michigan University, Ypsilanti, MI, 48197
[3] Graduate School of Computer and Information Sciences, Nova Southeastern University, 3301 College Avenue, Fort Lauderdale, Florida 33314

BIRCH [20] focused on speed by means of data summarization but favored circular clusters [9]. CURE [5] concentrated on sampling and outlier handling. DBSCAN [4] yielded good detections by favoring dense neighborhoods. As reported in [8], CURE and DBSCAN did not always detect outliers and depended on user-supplied parameters. RBCGA [12] used a genetic algorithm that utilized rectangular shapes, called rules, each representing a cluster, thus only accommodating rectangular cluster shapes. NOCEA [13] improved over RBCGA by providing multiple rules per cluster, thus offering better detection than RBCGA but, as stated in [13], its crossover operator broke large rules and did not always detect sparse areas within those rules, thus resulting in coarse detections. In [18], we introduced PYRAMID, a clustering algorithm that has proven to provide value in cluster and outlier detection on existing renowned datasets. We borrow directly from [18] in the ensuing description of PYRAMID.

III. Definitions

This section briefly introduces terms and concepts that are pertinent to the PYRAMID algorithm. The reader is encouraged to refer to [18] for further details. In all definitions, n symbolizes the number of points in a data set and d is the number of dimensions.

Definition 3.1: A Minimum Bounding Hyper-Rectangle (MBHR) is the smallest hyper-rectangular area in the data space that contains all the d-dimensional data points in a given data set [11].

Definition 3.2: Binning of dimension m within the MBHR where $m = 1, \ldots, d$, is the division of the m-axis into t_m non-overlapping segments, called *bins*. All bins within a dimension m have the same bin width.

Definition 3.3: The intersections of the bin lines construct a d-dimensional grid that divides the MBHR into contiguous non-overlapping d-dimensional **cells**, denoted *quantization*. Furthermore, cells have the following property:

$\forall$ cell c, width of c with respect to dimension $m = w_m$.

Definition 3.4: Rule r is the hyper-rectangular sub-region of the MBHR that contains one or more cells, denoted *constituent cells* of r. A rule r is said to overlap with another rule r' if they share at least one common constituent cell. This study does not allow overlapping rules within the same solution.

Definition 3.5: Individual I is the region in the MBHR that is a union of rules, called I's *constituent rules*. A list of the constituent rules of an individual I and their constituent cells, comma-separated, is called the individual *profile*, or profile(I). The *size* of a an individual *size*(I) is the number of rules in I.

The cardinality, volume, and density of cells, rules, and individuals are the same as outlined in [18].

Definition 3.6 Geometric Division is an algorithm that divides the data space into quadrants. A quadrant encompasses a data subset formed by the data points that belong to its constituent cells. The details of this algorithm are outlined in [18] and exemplified in Fig. 1 in a three-dimensional data space.

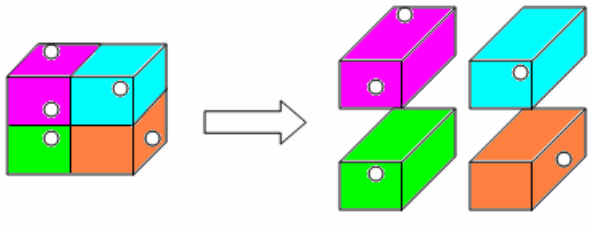

Fig. 1. Sample geometric division.

IV. The Approach

PYRAMID is a multi-step hybrid approach that is based on several components that utilize the above concepts. For the sake of simplicity, the rest of this study focuses on two-dimensional data sets as in [18] and leaves higher dimensions for future research. The PYRAMID algorithm is summarized in Fig. 2.

A. Data Transfer from Master to Slaves

The geometric division algorithm forms quadrants as groups of cells. The master processor sends each quadrant's data subset to a separate slave processor where another quantization is performed and a genetic program is executed.

```
Master Processor
1. Conduct binning.
2. Perform geometric division.
3. Send each subset to a different slave.
4. Receive p resulting subsets of discovered
   data points from p slaves. Determine
   cells that contain returned points.
5. Merge returned cells into global solution
   that labels every cell with a cluster.
Slave Processor
1. Receive a data subset P from master
   processor. Perform quantization on local
   data.
2. Run genetic program on the local data
   points in P (on current slave processor).
3. After algorithm finishes, send points in
   discovered cells to master processor.
```

Fig. 2. Master and slave roles in PYRAMID.

B. Genetic Program

A genetic program typically represents a solution as a tree-based individual [10]. In this study, each individual is encoded as a combination of blocks (rules) to form a genetic programming tree with leaf nodes symbolizing these constituent rules. This

representation offers more flexibility than genetic algorithm-based bit-strings [10]. A sample is shown in Fig. 3, which demonstrates the tree representation for individual I_1 from Fig. 4. As in standard genetic programming, the internal nodes represent the functions that apply to the leaf nodes [10]. In this study, the only function employed is union, which symbolizes that the individual is formed as a combination of its constituent rules (leaf nodes).

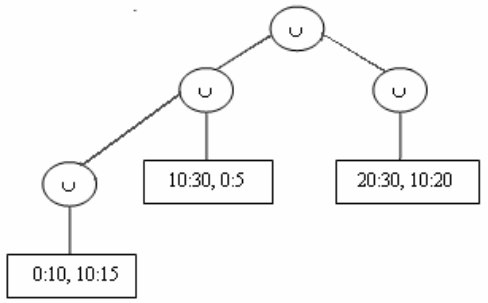

Fig. 3. Individual I_1 tree representation.

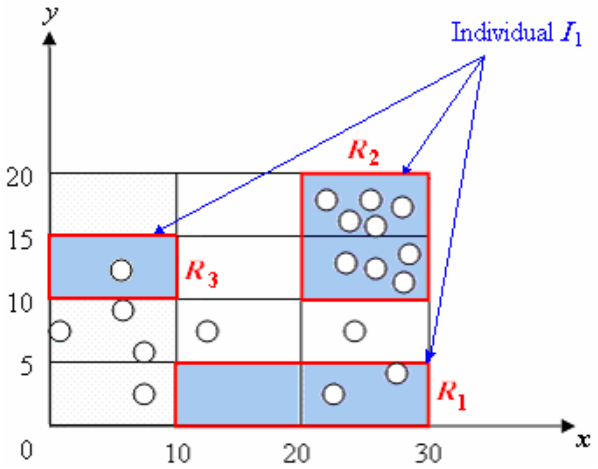

Fig. 4. Rules for individual in Fig. 3.

1) Genetic Operators

The main genetic operators used by PYRAMID are crossover, smart mutation, architecture altering (also called structural), and repair. The reader is encouraged to refer to [18] for further details about these operators. In summary, crossover occurs at the rule level by swapping rules between individuals thus producing two new individuals. Smart mutation has two flavors: enlarge mutation, which moves towards denser neighboring cells while shrinking mutation diminishes a rule by one bin with respect to a certain dimension m. Mutation always produces one new individual. Architecture altering adds a new rule or deletes an existing one. Finally, the repair operator the repair operator reforms overlapping rules into new ones that align better with the distribution of the data points. The repair sample is in Fig. 5 where the frame depicts the area covered by the original rule.

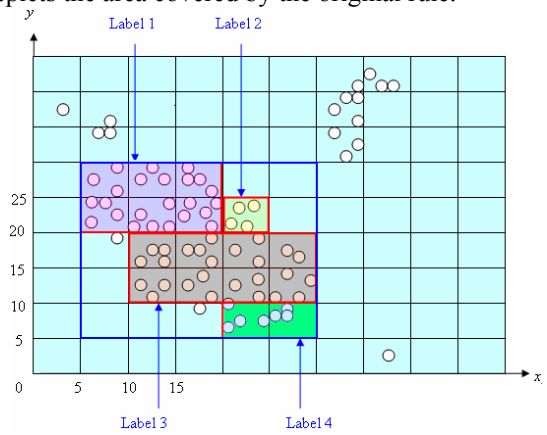

Fig. 5. PYRAMID repair operator.

2) Fitness Function

PYRAMID focuses on three main factors to achieve good solutions. It attempts to find a solution that achieves high coverage. It also tries to identify gatherings of dense areas in the MBHR by looking for solutions in the form of dense individuals composed of dense rules that contain dense cells. Finally, it attempts to avoid complex individuals by having a bias in favor of those having a smaller number of member rules, thus exercising parsimony pressure. Therefore, PYRAMID attempts to identify better solutions, or individuals, by incorporating, in its fitness function, the following three main objectives [18] shown in (1).

$$Fitness(I) = \frac{F_{coverage}(I) \times F_{density}(I)}{F_{size}(I)} \qquad (1)$$

3) Selection Operator and Elitism

The selection operator is based on tournament selection with a tour size of three [2]. We adopt one-individual elitism, whereby in every iteration, the best performer is preserved for the next generation [1].

4) Main Algorithm

The GP that is run on each slave processor is summarized in Fig. 6. After each operator is applied, the fitness of resulting individuals is evaluated.

```
t = 0
Initialize population t
Evaluate population t
While (not termination condition)
 Begin
  t = t + 1
  s = selection from population t-1
  c = crossover 2 individuals in t
  m = smart mutation
  a = architecture-altering
  e = elitism
  Evaluate(fitness) population t
 End
```

Fig. 6. Serial GP algorithm.

C. *The Merge Phase*

After the discovered points are sent back to the master, it traverses these discovered cells; assigning them cluster labels based on their neighborhoods. The merge algorithm was discussed in details in [18].

V. EXPERIMENTS

Several experiments were performed in [18] to test the ability of PYRAMID to identify clusters of arbitrary shapes, dynamic determination of the number of clusters, speedup using parallelism, independence of the order of input, and outlier handling. For these experiments, PYRAMID was run over existing two-dimensional data sets that were used by other algorithms such as NOCEA, CURE, DBSCAN, and RBCGA. Table 1 shows a list of those data sets. In addition, DS5 is included in this table, which will be utilized in a later experiment.

TABLE 1. DATA SETS USED IN PYRAMID EXPERIMENTS.

DATA SETS	DATA OBJECTS	CLUSTERS
DS1	8,000	6
DS2	10,000	9
DS3	100,000	6
DS4	1,120	3
DS5	100,000	100

Fig. 7 demonstrates the results that were produced in [18] after running PYRAMID on the first three data sets while Fig. 8 shows their detection by NOCEA. Fig. 9 depicts the detection of DS1 and DS2 by CURE and that of DS3 by DBSCAN. The left side of Fig. 10 shows how PYRAMID detected DS4 while the right side shows how RBCGA detected the same.

It is noteworthy from Fig. 7 how PYRAMID identified the correct number of clusters. With respect to the goodness of detection, it is evident in Fig. 10 and from a comparison between Fig. 7 and Fig. 8 that PYRAMID provided a smoother detection than RBCGA and NOCEA, respectively. Furthermore, Fig. 7 and Fig. 9 demonstrate how PYRAMID overcame outliers while CURE and DBSCAN did not [8].

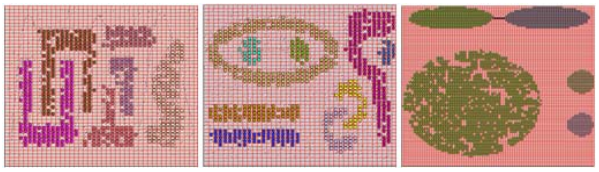

Fig. 7. PYRAMID cluster discovery.

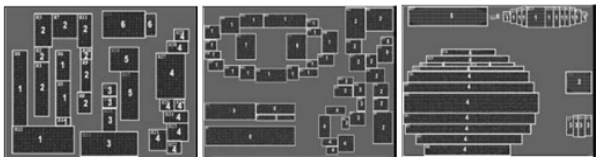

Fig. 8. NOCEA cluster discovery [13].

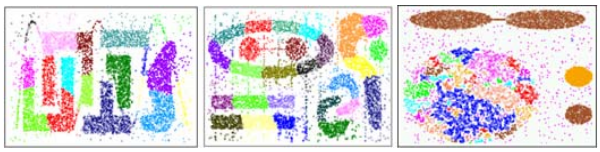

Fig. 9. Discovery of DS1, DS2 by CURE, DS3 by DBSCAN [8].

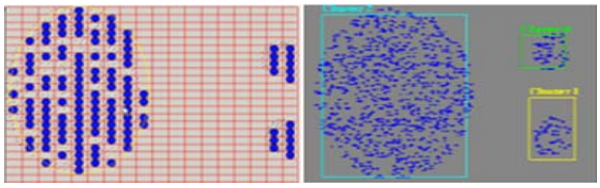

Fig. 10. DS4 by PYRAMID versus RBCGA [12].

Concerning the dependence on the order of input, the left part of Fig. 11 shows the result of running PYRAMID on DS1 with its original. The right part depicts the detection after DS1 was shuffled in a completely different order. It is obvious that both detections are fairly similar, thus demonstrating the independence of PYRAMID on the order of input.

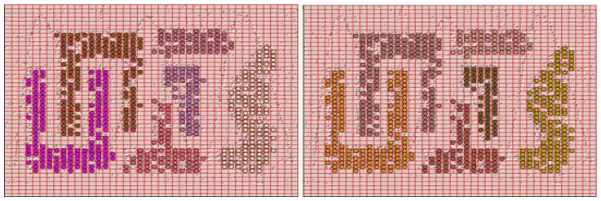

Fig. 11. Detection with different data order.

Other experiments were conducted in [18] to evaluate the improvements in speed that PYRAMID achieved from serial to parallel with four and sixteen slave processors, for data sets DS1, DS2, and DS3. The results showed considerable speedup improvements that ranged from 1.8 to 6.43. The reader is encouraged to refer to [18] for further details.

Furthermore, as suggested in the Conclusion and Future Work section of [18], further experiments were conducted in this research to evaluate two additional aspects: the performance of PYRAMID with a more challenging dataset, referred to as DS5 and the evaluation of PYRAMID's independence on user-supplied parameters.

Evaluation of PYRAMID with DS5:

DS5 contains one hundred clusters that are closely intermingled with a considerable amount of outliers, which makes it a very challenging dataset for various clustering algorithms. In [15], this dataset was processed by an algorithm, named WaveCluster, which transforms the original feature space by applying wavelet transform and then finding dense regions in the new space. This produces different clusters at different resolutions and scales, which can be chosen based on users needs [15]. Fig. 12 below compares the

WaveCluster detection of DS5 (middle) with that of PYRAMID (right). The circular shapes point to clusters that were erroneously grouped together under the same color, therefore representing the same cluster whereas they should not. It is noticeable that PYRAMID did erroneously combine some clusters. However, it is also visible that the number of such errors is lower than that in WaveCluster, as reported in [15].

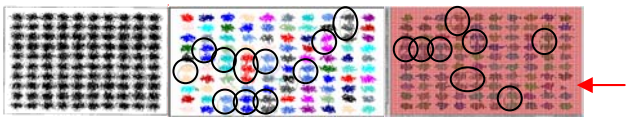

Fig. 12. DS5 and its discovery by WaveCluster and PYRAMID.

It is also worth noting that PYRAMID had a partial detection of one of the hundred clusters, as indicated by the arrow in Fig. 12.

Independence of PYRAMID on user parameters: Further experiments were conducted to evaluate the impact of modifying different parameters on the outcome of the PYRAMID algorithm. To accomplish that, the following parameters were modified:

- Population size was reduced from 15 to 10.
- Number of rules per individual was reduced from 10 to 5.
- Crossover percentage was changed from 20 to 45.
- Mutation percentage was changed from 35 to 20.
- Structural percentage was changed from 45 to 35.

As demonstrated in Fig. 13, PYRAMID was still able to identify nine clusters correctly for data set DS2. Although the quality of detection may look slightly different from its previous counterparts, it still provided good detection. The results have shown that detection remained fairly similar even when crucial parameters such as the genetic algorithm population size, number of rules per individual, and the genetic operator percentages. This is demonstrated in Fig. 13 where these parameters were changed and the results are compared to the original PYRAMID run for DS2 shown in Fig. 7.

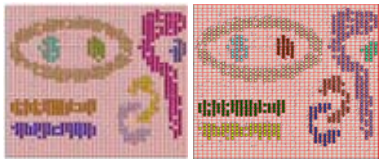

Fig. 13. PYRAMID detection with different parameters on DS2.

VI. Conclusion and Future Work

A novel approach to clustering large data sets, called PYRAMID, was introduced in [18] which leverages some of the concepts used in NOCEA [7]. PYRAMID improved over NOCEA by employing a hybrid combination of GP's global search and strong representational capabilities along with a powerful density-aware multi-objective fitness function. PYRAMID also employed data parallelism to achieve speedup. The experiments in [18] demonstrated that PYRAMID detects clusters of arbitrary shapes, is immune to outliers, and independent of the order of input. In addition, it does not require prior knowledge of the number of clusters, and its inherent data parallelism allows it to improve performance. This paper added to [18] by exercising the ability of PYRAMID to detect a more challenging dataset, DS5, which was employed in previous well known clustering research [15]. The results have demonstrated that, despite the challenges inherent within DS5, PYRAMID has exhibited relatively better detection than WaveCluster [15]. Another experiment also attested to the independence of PYRAMID on user-supplied parameters.

One possible avenue for future research is to revisit the PYRAMID algorithm and explore the performance measurements through speedup with higher dimensions. Other avenues include: performing additional experiments to assess various aspects of cluster detection such as exploring the use of rules with variable shapes, not strictly rectangular, and using other data sets as well as other forms of parallelism.

References

[1] Berkhin, P. (2002). Survey of clustering data mining techniques. *Accrue Software*. Retrieved February 28, 2005 from http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf

[2] Davis, L. (1991). *Handbook of Genetic Algorithms*. New York, NY: Van Nostrand Reinhold.

[3] Dettling, M. & Bühlmann, P. (2002). Supervised clustering of genes. *Genome Biology*, *3*(12), 39-50.

[4] Ester, M., Kriegel, H.P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon,* 226-231.

[5] Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, WA,* 73-84.

[6] Han, J. & Kamber, M. (2001). *Data Mining, Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann.

[7] Hinneburg, A. & Keim, D.A. (1998). An efficient approach to clustering in large multimedia databases with noise. *Proceedings of the Fourth International Conference on Knowledge Discovery in Databases, New York, NY,* 58-65.

[8] Karypis, G., Han, S., & Kumar, V. (1999). Chameleon: A hierarchical clustering using dynamic modeling. *IEEE Computer: Special Issue on Data Analysis and Mining*, *32*(8), 68-75.

[9] Kolatch, E. (2001). *Clustering Algorithms for Spatial Databases: A Survey* (Technical Report No. CMSC 725). Department of Computer Science, University of Maryland, College Park, 1-22.

[10] Koza, J.R. (1991). Evolving a computer program to generate random numbers using the genetic programming paradigm. *Proceedings of the Fourth International Conference on Genetic Algorithms, La Jolla, CA,* 37-44.

[11] Ohsawa, Y. & Nagashima, A. (2001). A spatio-temporal geographic information system based on implicit topology description:STIMS. *Proceedings of the Third International Society for Photogrammetry and Remote Sensing* (*ISPRS*) *Workshop on Dynamic and Multi-Dimensional Geographic Information System, Thailand,* 218-223.

[12] Sarafis, I., Zalzala, A., & Trinder, P. (2002). A genetic rule-based data clustering toolkit. *Proceedings of the 2002 World Congress on Evolutionary Computation, Honolulu, USA,* 1238-1243.

[13] Sarafis, I., Zalzala, A., & Trinder, P. (2003). Mining comprehensive clustering rules with an evolutionary algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, USA,* 1-12.

[14] Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice and Visualization.* New York, NY: John Wiley and Sons.

[15] Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). WaveCluster: a multi-resolution clustering approach for very large spatial databases. *Proceedings of the 24th Intl. Conf. on Very Large Data Bases, New York,* NY, 428-439.

[16] Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis.* London, UK: Chapman and Hall.

[17] Sturges, H. (1926). The choice of a class-interval. *Journal of the American Statistical Association*, *21*(1), 65–66.

[18] Tout, S., Sverdlik, W., & Sun, J. (2006). Parallel hybrid clustering using genetic programming and multi-objective fitness with density (PYRAMID). *Proceedings of the 2006 International Conference on Data Mining* (*DMIN'06*), *Las Vegas, NV, USA,* 197-203.

[19] Wang, W., Yang, J., & Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. *Proceedings of the 1997 International Conference on Very Large Data Bases, Athens, Greece,* 186-195.

[20] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada,* 103-114.

[21] Zitzler, E., Laumanns, M., and Thiele, L. (2001). *Spea2: Improving the Strength Pareto Evolutionary Algorithm* (Technical Report No.103). Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, 1-19.