A Genetic Algorithm Approach to Cluster Analysis

Marc C. Cowgill and Robert J. Harvey
Department of Psychology
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0436 U.S.A.

Layne T. Watson
Departments of Computer Science and Mathematics
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0106 U.S.A.

Abstract—A common problem in the social and agricultural sciences is to find clusters in experimental data; the standard attack is a deterministic search terminating in a locally optimal clustering. We propose here a genetic algorithm (GA) for performing cluster analysis. GAs have been used profitably in a variety of contexts in which it is either impractical or impossible to directly solve for a globally optimal solution to complex numerical problems. In the present case, our GA clustering technique attempted to maximize a variance-ratio (VR) based goodness-of-fit criterion defined in terms of external cluster isolation and internal cluster homogeneity. Although our GA-based clustering algorithm cannot guarantee to recover the cluster solution that exhibits the global maximum of this fitness function, it does explicitly work toward this goal (in marked contrast to existing clustering algorithms, especially hierarchical agglomerative ones such as Ward's method). Using both constrained and unconstrained simulated datasets, Monte Carlo results showed that in some conditions the genetic clustering algorithm did indeed surpass the performance of conventional clustering techniques (Ward's and K-means) in terms of an internal (VR) criterion. Suggestions for future refinement and study are offered.

1. INTRODUCTION

There is certainly no shortage of either clustering algorithms (e.g., Anderberg [1]) or statistics that attempt to quantify the quality or goodness-of-fit of obtained cluster solutions (e.g., Milligan [2]). However, remarkably little consensus exists with respect to the question of which clustering algorithm performs the best under given circumstances, or, for that matter, which goodness-of-fit criterion should be used to answer this question. Consequently, we find continued research interest in both the development and evaluation of new clustering algorithms (e.g., [3], [4]), as well as in evaluating the performance of the various cluster algorithms and fit indices (e.g., [5], [6]).

In many situations it remains conceptually desirable to define the "best" cluster assignment as being the one that achieves a globally optimal value for internal cluster cohesion and external cluster isolation, for example, when one has reason to believe that the true latent cluster membership pattern is one that demonstrates *independent-cluster* structure. Unfortunately, although the goals of within-cluster cohesion and external cluster isolation have long been viewed as desirable ones, the computational algorithms used by most existing clustering techniques tend *not* to be ones that seek a globally optimal solution with regard to these properties. This is especially the case for the hierarchical agglomerative algorithms (e.g., Ward's [7]) that lock clustered entities into the cluster (or descendents of that cluster) into which they were initially assigned. Thus, although at any given step of the hierarchy the clusters may be formed to produce the best possible isolation and cohesion possible in view of the existing cluster memberships, this will *not* necessarily produce an assignment that achieves the best possible cluster separation and cohesion for the given number of clusters. As a practical matter, the extremely large number of possible cluster assignments that exists for even moderate-sized problems (e.g., with 50 entities to be clustered, there are the Stirling number of the second kind $\left\{ \begin{matrix} 50 \\ 5 \end{matrix} \right\} \approx 10^{32}$ possible 5-cluster assignments and $\left\{ \begin{matrix} 50 \\ 10 \end{matrix} \right\} \approx 10^{43}$ possible 10-cluster assignments; with 100 entities, there are $\left\{ \begin{matrix} 100 \\ 5 \end{matrix} \right\} \approx 10^{67}$ 5-cluster assignments and $\left\{ \begin{matrix} 100 \\ 10 \end{matrix} \right\} \approx 10^{93}$ 10-cluster assignments) makes it highly impractical to attempt a brute-force search of all possible assignments to find the one that maximizes cluster isolation and cohesion.

To address this problem, we developed a clustering algorithm that explicitly seeks to optimize a fit criterion defined in terms of within-cluster cohesion and between-cluster isolation. The algorithm appraised in this study, COWCLUS, uses what is termed a *genetic* algorithmic approach in an attempt to recover cluster structure under diverse dataset conditions. Genetic algorithms (GAs) can be differentiated from hill-climbing, deterministic algorithms such as Ward's [7] in that they (a) employ probabilistic search procedures (e.g., use random number generation), (b) search from a population of points, not a single point, and (c) use evolutionary principles, such as selection, mating, and mutation.

Genetic algorithms have been successfully employed in a wide array of scenarios, including composite material design [8], [9], [10], [11], aircraft design [12], [13], computer multitasking [14], noise suppression [15], and even in generation of models of consumer choice [16]; for general references, see Belew and Booker [17], Buckley and Petry [18], Goldberg [19], and Holland [20]. GAs, relative to hill-climbing methods, have performed particularly well under noisy dataset conditions (e.g., [21], [22], [19], [23]).

Past efforts to apply the genetic approach to classification problems include attempts to (a) find good ordered representations (permutations) of objects (Bhuyan, Raghavan, and Elayavalli [17]; Mishra and Raghavan [24]), (b) divide N numbers into K groups as to minimize differences among group sums (Jones and Beltramo [17]), (c) color the U.S. map so that no two bordering states are the same color (Jones and Beltramo [17]), (d) solve a set partitioning problem related to airline flight crew scheduling (Levine [25]), and (e) improve upon Wong and Lane's [26] Kth nearest neighbors classification algorithm [27]; for a general discussion, see Falkenauer [28].

COWCLUS, alternatively, is a general-purpose, independent-cluster-seeking algorithm. An independent-cluster-seeking algorithm is one in which each object may belong to only one cluster. Though COWCLUS can be modified to maximize almost any objective function, its current implementation attempts to maximize the Calinski and Harabasz [29] Variance Ratio Criterion (VRC), a measure often associated with determining the correct number of clusters in a dataset. VRC is defined as:

$$\mathrm{VRC} = \frac{\text{trace } B/(k-1)}{\text{trace } W/(n-k)}, \tag{1}$$

where n and k are the total number of objects and the number of clusters in the partition, respectively; the B and W terms are the between-cluster and the pooled within-cluster sums of squares (covariance) matrices.

VRC was chosen to be the objective function due to its high intuitive appeal as to what constitutes "true" cluster structure. In general, researchers have posited the existence of clusters as distinct groups, possessing the qualities of internal cohesion and external isolation (e.g., [30], [31], [32], [33]). VRC appears to reflect such a description, more so than the objective function of most hierarchical clustering algorithms (among them, Ward's):

$$d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \lambda |d_{ki} - d_{kj}|. \tag{2}$$

In this formula $d_{k(ij)}$ represents the revised distance value between cluster k and the newly formed group (ij) which was formed by merging clusters i and j. The parameter values α_i, α_j, β, and λ determine the particular HCA routine [34], [35]. For Ward's technique, $\alpha_i = (n_k+n_i)/(n_k+n_{ij})$,

$\alpha_j = (n_k + n_j)/(n_k + n_{ij})$, $\beta = -n_k/(n_k + n_{ij})$, and $\lambda = 0$, where n refers to the number of objects in a cluster.

VRC was picked also due to its excellent performance against other internal criteria. In a study by Milligan [2], a very similar measure, McClain and Rao's [36] W/B statistic, fared extremely well in comparison with 29 other internal criteria. VRC also may be the most effective criterion for purposes of cluster number determination [37], [38]. Moreover, VRC shares an isomorphism with the F statistic, the linchpin of most inferential analyses in current behavioral research. Just as the F statistic gauges the size of differences between groups in the context of an ANOVA, VRC measures the degree of isolation between clusters.

2. DESIGN OF COWCLUS

Most of the clustering techniques used today are considered hill-climbing techniques, in that they obtain their solutions by passing through the data in the steepest permissible direction locally, as defined by an objective function value. Ward's algorithm, for example, begins with the computation of a distance matrix between the $n(n-1)/2$ pairs of objects, each object representing a cluster. Based on the objective function in Equation (2), clusters are then successively merged until only one cluster remains. Alternatively, COWCLUS searches through the data by (a) testing a population of *cluster assignments* (a.k.a. partitions, referred to as *members*), (b) retaining the best members, (c) combining or *mating* the surviving members in order to form novel members, and (d) mutating or altering a small percentage of members in order to ensure diversity in the population (see Figure 1). Parallel search is inherent throughout each COWCLUS run—the search path is not determined by the calculation of a single best value. An advantage of this strategy lies in its potential to *work* around local optima [19].

Perhaps the most ostensible difference between COWCLUS and other clustering methods is that COWCLUS searches through pre-generated assignments rather than building assignments up from inter-object distances. The initial population of COWCLUS members is generated from random cluster assignments (partitions). By using the results of previous analyses as a template, the potential search space is constrained to aid a guided search through the space of possible partitions. In this sense COWCLUS is a genetic/hill-climbing hybrid: first a nondeterministic genetic algorithm is used to find many good partitions, then deterministic hill-climbing methods are used to improve (if possible) these partitions producing the final best partition. This application of deterministic hill climbing to members of a population is known as *local improvement*; it is not done at each iteration because the cost would be prohibitive.

After a predefined number of generations and the application of local improvement to the last generation, the member possessing the highest VRC is chosen as the COWCLUS solution. The number of generations is set by the user, and is based upon the number of objects to be clustered, the speed of the computing system, and several other context-specific considerations. A precise description of COWCLUS follows.

3. GENETIC ALGORITHM

In general, a genetic algorithm (GA) works with a population of b individuals, each representing in this case an assignment of n points to k clusters. The population of individuals (cluster assignments) goes though a selection process for breeding, whereby assignments with a higher merit function, the Variance Ratio Criterion (VRC) here, have a higher probability of being selected for breeding. Breeding is performed in such a manner that the child assignments maintain some likeness to the parent assignments. The next generation of assignments is comprised of b children created during the breeding process. Thus, the population size remains constant throughout the generations. The process continues for many generations, and terminates after a fixed number of generations, although there are many other reasonable stopping criteria which may be applied. The assignment with the best merit function (VRC) value in the end, after local improvement, is used. The details of breeding for a GA are described next.

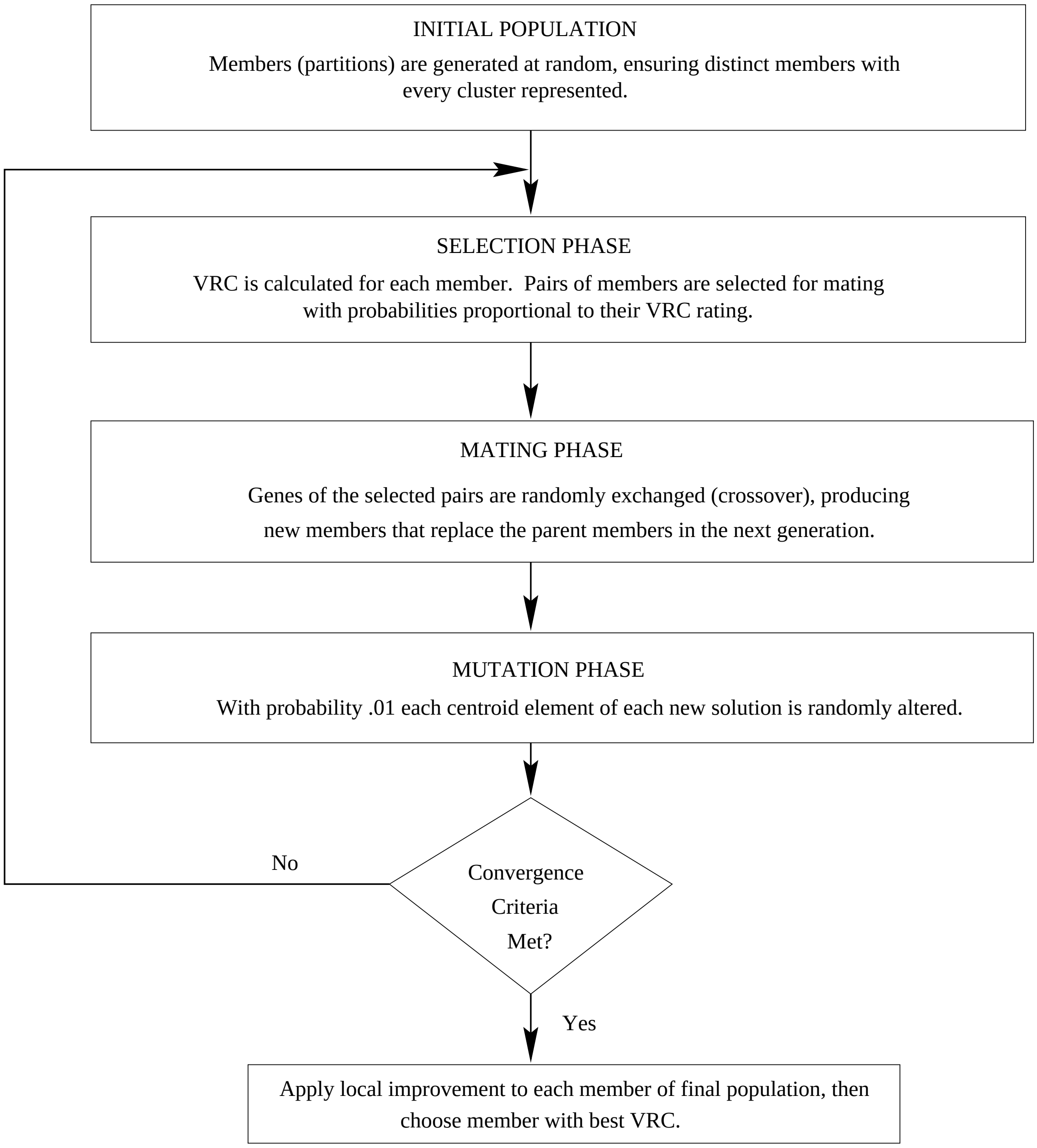

Figure 1. Outline of the COWCLUS algorithm.

The intent of the GA is to select the partition of n points into k clusters that maximizes the VRC, the measure of cluster quality. Any partition of the n distinct points into k clusters is called a candidate. To start the algorithm, an initial population of b candidates is created by randomly creating b cluster assignments, ensuring that the assignments are distinct and that each cluster is represented in an assignment. A candidate is represented by a sequence of n integers, each between 1 and k, where the jth component (called a *gene*) of the sequence indicates to which cluster the jth point is assigned. Each cluster must have at least one point assigned, so each integer between 1 and k must appear at least once in the genetic sequence.

The candidates are selected as parents for breeding based on the fitness VRC. Candidates with a higher fitness will be selected more often than candidates with lower fitness. The rank fitness is defined as $b + 1 - r$, where r is the rank of the candidate in the population in terms of the value of the VRC. The probability of selection as a parent is proportional to the rank fitness, so that the probability of the rth ranked assignment being selected as a parent is

$$p_r = \frac{(b+1-r)}{b(b+1)/2}.$$

The selection process for parenting is completed by generating a uniformly distributed random number x between zero and one, and selecting the rth ranked assignment satisfying $P_r \leq x < P_{r+1}$, where

$$P_r = \sum_{i=1}^{r-1} p_i.$$

(A crude variant of this selection scheme is to simply consider each candidate in turn, starting with the highest ranked, generate a random number x between zero and one, and select the rth ranked candidate as a parent if $x \leq 1 - r/b$, continuing to cycle through the candidates until two parents have been selected. Since the mechanics of selection are not significant for the overall performance of a genetic algorithm, COWCLUS uses the cruder scheme.) After two parents have been selected the breeding process begins. A random integer j between one and $n-1$ (the number of genes -1) is generated. The child is constructed from the first j genes of the first parent and the last $n-j$ genes from the second parent. This process is called the *crossover* rule in GA parlance. (Technically this is known as one-point crossover. Two-point crossover exchanges a subsequence in the middle of the genetic sequence, and uniform crossover takes each gene of the child with probability .5 from the corresponding gene in each of the parents. COWCLUS uses uniform crossover in the early generations and two-point crossover in later generations.) Once the child is generated it goes through a mutation process. Each gene has a small chance (probability .01) of mutating, decided by generating a random number. If a gene is selected for mutation, the gene is replaced with a gene coming randomly from the set of allowable values for that gene. Finally the child is checked for validity—each cluster must be represented in the genetic sequence—and uniqueness, and destroyed if it is invalid or a duplicate.

A variant of this algorithm, known as an "elitist" strategy, is that after $b-1$ children are created, the parent generation is replaced, retaining the best parent for the next generation. After some stopping criterion (say, a fixed number of generations) has been satisfied, as a practical matter, local improvement (any deterministic, local hill-climbing method such as K-means) is applied to the best few individuals in this final generation, and the candidate with the highest VRC is used. There is no mathematical guarantee that the output of the genetic algorithm is truly a global optimum for the VRC; the justification for the GA approach is that it typically beats deterministic or random search methods.

4. METHOD

To evaluate the performance of COWCLUS, the Monte Carlo approach was chosen in accordance with the conventions established and maintained in previous clustering technique validation studies (e.g., [39], [40]). First, artificial datasets with known cluster structure were generated. Next, the constructed datasets were analyzed using the following clustering algorithms: COWCLUS, Ward's, convergent K-means with Ward's centroids as starting vectors (K-means[W]; see Anderberg [1]), and convergent K-means with random starts (K-means[R]; see Anderberg [1]). Finally, the performance of each method was assessed by examining both the degree to which each method recovered the known structure (an external criterion), as well as using an internal criterion that was sensitive to the degree of internal cohesion and external isolation of the clusters (VCR). Although the information produced using the external criterion arguably provides the "bottom line".

To assess the performance of the new COWCLUS algorithm across a wide range of possible cluster structures, two kinds of datasets were analyzed: (a) error-free datasets, exhibiting considerable separation among clusters, and (b) unconstrained datasets with varying levels of noise, manifesting considerable cluster overlap. The procedure of dataset generation generally followed the conventions of previous Monte Carlo clustering studies (e.g., [39], [41], [40]).

4.1. Constrained Datasets

Multivariate mixtures where generated with an algorithm similar to that described in Milligan [32], [33] and frequently employed in past constrained-dataset studies (e.g., [39], [32], [33], [42], [43]). The clusters were well-separated, truncated multivariate normal mixtures (sets of distinct hyperellipsoids), embedded in a ten dimensional space. The dimensions were uncorrelated. The 324 datasets reflected a factorial arrangement of (a) number of clusters (3, 6, or 9), (b) sample size (50, 150, or 500 objects) and (c) evenness of cluster size (objects spread evenly across clusters, 10% of the objects in one cluster, 60% of the objects in one cluster). There were 12 datasets in each of the 27 cells. No noise was added to the data to ensure a low potential for cluster overlap.

4.2. Unconstrained Datasets

These datasets conformed to those generated in previous studies employing unconstrained datasets (e.g., [41], [40]). The clusters overlapped, in that the 10 variable means for each cluster were chosen randomly from a uniform distribution ranging from 20 to 40, with standard deviations chosen from a uniform distribution ranging from 5 to 10. Additionally, the correlation structure among variables of each cluster was specified in the manner of Blashfield [44] and Scheibler and Schneider [40]: The eigenstructures of the correlation matrices were such that the number of eigenvalues > 1.0 was determined by randomly selecting an integer from a uniform distribution from 1 to 5. The correlation matrix with this eigenstructure was then randomly selected from the population of such matrices using the algorithm given by Lin and Bender (see [45] for more details).

As with the constrained data, there were 324 datasets and the factorial design included (a) number of clusters, (b) sample size, and (c) cluster size evenness. A 3-level noise factor was also included: In the first condition no noise was present. In condition 2, 25% of the data values were replaced with uniform random noise (ranging from 20 to 40). In condition 3, 50% of the values were replaced with random noise.

The factorial design was therefore: 3 (number of clusters) $\times$ 3 (sample size) $\times$ 3 (cluster size evenness) $\times$ 3 (noise). To equal the total number of constrained datasets generated, 4 datasets were generated for each of the 81 cells of the unconstrained design. The numbers of constrained and unconstrained datasets were balanced so the genetic algorithm comparisons between the two would be fair.

4.3. Performance Criteria

The internal criterion employed was the aforementioned Calinski and Harabasz [29] Variance Ratio Criterion. With regard to COWCLUS, each of the 648 datasets was allowed approximately 25 minutes of dedicated processing time on an IBM RS/6000 model 355 workstation.

Constrained data. For each method, the VRC was recorded. The *Wins* statistic described below was not recorded due to the high number ($\sim$100%) of ties.

Unconstrained data: Wins. The number of times each method achieved the highest criterion value for a given dataset was recorded. Ties (instances where two or more methods yielded values within 0.1 percent) were excluded.

5. RESULTS

The constrained column in Table 1 illustrates how well COWCLUS, Ward's, and K-means(W) perform under ideal conditions. Each of these methods achieved perfect recovery in almost every

Table 1.
Constrained (clearly separated) and unconstrained (overlapping) mixtures: mean clustering performance as a function of clustering method.

	Variance Ratio Criterion		
Method	Constrained	Unconstrained	Wins
COWCLUS	1042.4653	31.2484	316 (∼100%)†
Ward's	1042.4268	25.8296*	0 (0%)
K-means(W)	1042.4623	29.7802	0 (0%)
K-means(R)	284.7385	27.6351	1 (0%)
Total (N)	324	324	317‡

* $p<.05$ Duncan's MRT, † $p<.001$, ‡ ties have not been included.

instance where the dataset reflected high cluster separation and contiguity. Only K-means(R) shows relatively poor performance. A Duncan's Multiple Range Test (MRT) reveals significant differences ($p < .05$) between the performance of K-means(R) and each of the other methods.

The results from the unconstrained dataset clustering (Table 1) require considerable scrutiny. The VRC results reflect an area of superiority that COWCLUS solutions enjoy over Ward's solutions. Specifically, the MRT shows (a) no significant differences among COWCLUS, K-means(W), and K-means(R), and (b) significantly worse performance for Ward's method.

The last column of Table 1 reports tallies of the Wins statistic for unconstrained data. For the VRC Wins data, no inferential analysis was necessary, because COWCLUS won 97.53% of the trials, and Ward's and K-means won zero trials. Most striking is the almost perfect consistency with which COWCLUS yielded superior VRC values. The wins results offer considerable evidence as to the superiority of COWCLUS performance in this study.

6. DISCUSSION

6.1. Is Yet Another Clustering Algorithm Needed?

A great many clustering algorithms are now available, and one could be forgiven for asking whether the field of cluster analysis really needs yet another addition to these existing methods. In response, we would note that although the field currently suffers from no shortage of clustering methods, many (if not most) suffer from potentially serious limitations and criticisms, perhaps one of the most significant being that they do not effectively seek a globally optimal solution with respect to a conceptually meaningful clustering criterion (e.g., maximized internal cohesion and cluster separation). Although COWCLUS or other GA-based methods cannot guarantee to find the global maximum of VRC or similar criteria, such techniques do possess the important advantage of at least trying to seek this objective at a solution-wide level of analysis. Older algorithms—especially those based on stepwise, agglomerative algorithms such as Ward's—cannot make a similar claim. Indeed, only in exceptionally clear-cut cases (typically involving simulated data with little or no noise) would we expect them to identify a solution that is globally optimal in this sense.

As a practical matter, we must stress that the performance of the COWCLUS procedure was consistently superior to K-means and Ward's with regard to the results from the unconstrained datasets, and essentially on a par with K-means(W) and Ward's for the "easier" constrained datasets. It is tempting to make the argument that the unconstrained datasets are more "realistic" in their makeup (in the sense of being more like what would be seen in non-simulated data), and the fact that COWCLUS was consistently superior to the two traditional methods.

6.2. Future Directions for the Genetic/Hill-climbing Hybrid

While issues concerning Monte Carlo validation are being examined, the work on COWCLUS design should continue. For example, the COWCLUS hill-climbing seeds should be derived from a wider variety of methods (density search, complete linkage, and others). In theory, greater diversity of starting assignments should enhance COWCLUS' ability to work around local optima.

The parameters for COWCLUS also deserve research attention. In the current exploratory study, the effects of the following were not investigated fully: (a) number of generations, (b) types of mating used to form new members, (c) ways of scaling objective function values when deciding which members survive into the next generation, and (d) population size.

Perhaps most important is whether VRC serves as the best objective function (genetic algorithm merit function) for COWCLUS clustering. As mentioned earlier, almost any kind of objective function can be employed, and other internal criteria should be evaluated for their efficacy or robustness in this role.

Finally, while this study dealt with independent-cluster-seeking algorithms, COWCLUS can be modified to yield overlapping clusters. In many, if not most, applied contexts, cluster overlap may be present and if so, should be represented in any cluster assignment. It remains to be seen whether a genetic/hill-climbing routine can yield superior assignments for these cases.

6.3. Conclusions

The genetic algorithm approach in COWCLUS holds promise for cluster analysis research and application. With regard to the Rand evaluation, COWCLUS appeared to perform as well or better than any of the other methods. When considering the VRC evaluation, COWCLUS consistently produced significantly better assignments than Ward's technique, one of the most popular clustering methods. To the extent that researchers value the production of cluster assignments that demonstrate external isolation and internal consistency (and hence, high VRC), they may prefer the COWCLUS algorithm over the Ward's and K-means algorithms.

Still in an early stage of development, COWCLUS is likely to perform better as various aspects of its operation are refined and the speed of computers increases. Increasing the length (with a given initial population) and number (with different random initial populations) of computer runs improves the quality of the genetic algorithm assignments, while the quality of the Ward's and K-means assignments can not be improved by investing more computing resources. An extremely significant advantage of COWCLUS and genetic algorithms in general is that they can fully utilize parallel (even massively parallel) computers—child creation via crossover and the merit computation for individuals is completely independent and can be done in parallel, even with distributed machines over a network. An alternative coarse grained approach is to have each processor in a parallel computer run COWCLUS with a different initial population. None of the known deterministic clustering algorithms can significantly benefit from parallel computing hardware, the de facto architecture for all current supercomputers.

REFERENCES

1. M. R. Anderberg, *Cluster analysis for applications*, New York: Academic Press, (1973).
2. G. W. Milligan, A Monte Carlo study of thirty internal criterion measures for cluster analysis, *Psychometrika* **46**, 187–199 (1981).
3. P. V. Balakrishnan, M. C. Cooper, V. S. Jacob, and P. A. Lewis, A study of the classification capabilities of neural networks using unsupervised learning: A comparison with K-means clustering, *Psychometrika* **59**, 509–525 (1994).
4. W. S. DeSarbo, R. L. Oliver, and A. Rangaswamy, A simulated annealing methodology for clusterwise linear regression, *Psychometrika* **54**, 707–736 (1989).
5. Cheng, R. and G. W. Milligan, Mapping influence regions in hierarchical clustering, *Multivariate Behavioral Research* **30**, 547–576 (1995).
6. J. R. Donoghue, The effects of within-group covariance structure on recovery in cluster analysis, *Multivariate Behavioral Research* **30**, 227–254 (1995).
7. J. H. Ward, Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association* **58**, 236–244 (1963).
8. N. Kogiso, L. T. Watson, Z. Gürdal, and R. T. Haftka, Genetic algorithms with local improvement for composite laminate design, *Structural Optim.* **7**, 207–218 (1994).
9. N. Kogiso, L. T. Watson, Z. Gürdal, R. T. Haftka, and S. Nagendra, Design of composite laminates by a genetic algorithm with memory, *Mech. Composite Materials Structures* **1**, 95–117 (1994).

10. S. Nagendra, R. T. Haftka, Z. Gürdal, and L. T. Watson, Derivative based approximation for predicting the effect of changes in laminate stacking sequence, *Structural Optim.* **11**, 235–243 (1996).
11. S. Nagendra, D. Jestin, Z. Gürdal, R. T. Haftka, and L. T. Watson, Improved genetic algorithms for the design of stiffened composite panels, *Comput and Structures* **58**, 543–555 (1996).
12. S. Burgee, A. A. Giunta, V. Balabanov, B. Grossman, W. H. Mason, R. Narducci, R. T. Haftka, and L. T. Watson, A coarse grained parallel variable-complexity multidisciplinary optimization paradigm, *Internat. J. Supercomputer Appl. High performance Comput.* **10**, 269–299 (1996).
13. M. Kaufman, V. Balabanov, S. L. Burgee, A. A. Giunta, B. Grossman, R. T. Haftka, W. H. Mason, and L. T. Watson, Variable-complexity response surface approximations for wing structural weight in HSCT design, *Comput. Mech.* **18**, 112–126 (1996).
14. G. Syswerda and J. Palmucci, The application of genetic algorithms to resource scheduling, *Proc. of the Fourth International Conf. on Genetic Algorithms*, (Edited by K. Belew and L. B. Booker), pp. 502–508, San Mateo, CA: Morgan Kaufman, (1991).
15. C. H. Chu, A genetic algorithm approach to the configuration of stack filters, *Proc. of the Third International Conf. on Genetic Algorithms*, (Edited by J. D. Schaffer), pp. 416–421, San Mateo, CA: Morgan Kaufmann, (1989).
16. D. P. Greene and S. F. Smith, A genetic system for learning models of consumer choice, *Proc. of the Second International Conf. on Genetic Algorithms*, (Edited by J. J. Grefenstette), pp. 217–223, Hillsdale, NJ: Lawrence Erlbaum, (1987).
17. R. K. Belew and L. B. Booker, (Eds.), *Proc. of the Fourth International Conf. on Genetic Algorithms*, San Mateo, CA: Morgan Kaufman, (1991).
18. B. P. Buckley and F. E. Petry, *Genetic algorithms*, Los Alamos, CA: IEEE Computer Society Press, (1992).
19. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, New York: Addison-Wesley, (1989).
20. J. Holland, *Adaptation in natural and artificial systems*, Ann Arbor: University of Michigan Press, (1975).
21. L. Davis, *Handbook of genetic algorithms*, New York: Van Nostrand Reinhold, (1991).
22. K. De Jong, Genetic algorithms: A 10 year perspective, *Proc. of the First International Conf. on Genetic Algorithms and their Applications*, (Edited by J. J. Grefenstette), pp. 169–177, Hillsdale, NJ: Lawrence Erlbaum, (1985).
23. S. W. Wilson, GA-easy does not imply steepest-ascent optimizable, *Proc. of the Fourth International Conf. on Genetic Algorithms*, (Edited by K. Belew and L. B. Booker), pp. 85–91, San Mateo, CA: Morgan Kaufman, (1991).
24. S. K. Mishra and V. V. Raghavan, *An empirical study of the performance of heuristic methods for clustering*, Unpublished doctoral dissertation, University of Southwestern Louisiana, Lafayette, LA, (1994).
25. D. Levine, *A parallel genetic algorithm for the set partitioning problem*, (Report No. ANL-94/23), Washington, DC: U.S. Department of Energy, (1994).
26. M. A. Wong and T. Lane, A kth nearest neighbor clustering procedure, *Journal of the Royal Statistical Society* **Series B, 45**, 362–368 (1983).
27. J. D. Kelly and L. Davis, Hybridizing the genetic algorithm and the K nearest neighbors classification algorithm, *Proc. of the Fourth International Conf. on Genetic Algorithms*, (Edited by K. Belew and L. B. Booker), pp. 377–383, San Mateo, CA: Morgan Kaufman, (1991).
28. E. Falkenauer, A new representation and operators for GAs applied to grouping problems, *Evolutionary Computation* (in press).
29. R. B. Calinski and J. Harabasz, A dendrite method for cluster analysis, *Communications in Statistics* **3**, 1–27 (1974).
30. R. M. Cormack, A review of classification, *Journal of the Royal Statistical Society* **134**, 321–367 (1971).
31. B. S. Everitt, *Cluster Analysis*, London: Halstead Press, (1974).
32. G. W. Milligan, An examination of the effect of six types of error perturbation on fifteen clustering algorithms, *Psychometrika* **45**, 325–342 (1980).
33. G. W. Milligan, An algorithm for generating artificial test clusters, *Psychometrika* **50**, 123–127 (1985).
34. G. N. Lance and W. T. Williams, A generalized sorting strategy for computing classification, *Nature* **212**, 218 (1966).
35. G. N. Lance and W. T. Williams, A general theory of classificatory sorting strategies: I. Hierarchical systems, *Computer Journal* **9**, 373–380 (1967).
36. J. O. McClain and V. R. Rao, CLUSTISZ: A program to test for the quality of clustering of a set of objects, *Journal of Marketing Research* **12**, 456–460 (1975).
37. S. Krokal-Schwerdt and T. Eckes, A graph theoretic criterion for determining the number of clusters in a data set, *Multivariate Behavioral Research* **27**, 541–565 (1992).
38. G. W. Milligan and M. C. Cooper, An examination of procedures for determining the number of clusters in a data set, *Psychometrika* **50**, 159–179 (1985).
39. L. Belbin, D. P. Faith, and G. W. Milligan, A comparison of two approaches to beta-flexible clustering, *Multivariate Behavioral Research* **27**, 417–433 (1992).
40. D. Scheibler and W. Schneider, Monte Carlo tests of the accuracy of cluster analysis algorithms–A comparison of hierarchical and nonhierarchical methods, *Multivariate Behavioral Research* **20**, 283–304 (1985).
41. J. N. Breckenridge, Replicating cluster analysis: Method, consistency, and validity, *Multivariate Behavioral Research* **24**, 147–161 (1989).

42. G. W. Milligan, A study of the beta-flexible clustering method, *Multivariate Behavioral Research* **24**, 163–176 (1989).
43. G. W. Milligan, S. C. Soon, and Sokol L. M., The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure, *IEEE Transactions of Pattern Analysis and Machine Intelligence* **5**, 40–47 (1983).
44. R. K. Blashfield, Mixture model tests of cluster analysis: Accuracy of four agglomerative hierarchical methods, *Psychological Bulletin* **83**, 377–388 (1976).
45. M. Cowgill, Monte Carlo validation of two genetic clustering algorithms, *Dissertation Abstracts International* **54**, 2266-B. (University Microfilms No. 93-23-767) (1993).
46. C. Edelbrock, Mixture model tests of hierarchical clustering algorithms: The problem of classifying everybody, *Multivariate Behavioral Research* **14**, 367–384 (1979).
47. Hubert, L. and P. Arabie, Comparing partitions, *Journal of Classification* **2**, 193–218 (1985).
48. J. B. MacQueen, Some methods for classification and analysis of multivariate observations, *Proc. 5th Symposia on Mathematics, Statistics, and Probability*, pp. 281–297, Berkeley, CA, (1967).
49. L. J. Price, Identifying cluster overlap with NORMIX Population membership probabilities, *Multivariate Behavioral Research* **28 (2)**, 235–262 (1993).
50. V. V. Raghavan and B. Agarwal, Optimal determination of user-oriented clusters: An application for the reproductive plan, *Proc. of the Second International Conf. on Genetic Algorithms*, (Edited by J. J. Grefenstette), pp. 241–246, Hillsdale, NJ: Lawrence Erlbaum, (1987).