Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul, Korea • May 27-30, 2001

Parallel Clustering System Using the Methodologies of Evolutionary Computations

Reginald L. Walker
Computer Science Department
University of California at Los Angeles
Los Angeles, California 90095-1596
rwalker@cs.ucla.edu

Abstract- Several versions of the parallel clustering system were studied to improve performance of its initial implementation. The current versions were restricted to 1024 Web pages which, in turn, were used to create adaptive probe sets that were distributed to each indexer node. The probe sets were used to compute fitness measures associated with each indexer node used to create sub-species for the purpose of applying the new and traditional GA/GP operators. Speedup resulted from fitness-enhancing mechanisms that provided information results from previous fitness measurements of previous generations, such as the non-genetic transmission of cultural information. The clustering results are being used in the Tocorime Apicu project to develop a bioinformatic approach to the design and validation of an integrated, experimental search engine. This model provides a foundation for an evolutionary expansion of this computational model as World Wide Web (WWW) documents continue to grow. The clustering results were generated using message passing interface (MPI) on a network of SUN workstations.

1 Introduction

The methodologies of evolutionary computations (EC) [bfn96],[dqp00],[wal01] incorporate the following genetic operators: cross-over, mutation/editing, reproduction, and migration. A similar group of evolutionary operators associated with the biological model are: migration, swarming, and supersedure. The cross-over operator is similar to the reproduction operator for the chosen females associated with this biological model, but differs since the parent chromosomes cease to exist in genetic algorithms (GA)/genetic programming (GP) but persist in the true evolutionary sense. The children chromosomes replace their parents in standard GP. The migration operator in GP purges an existing population of the least desirable members, and in some cases the best member in some GP applications, as an attempt to avoid local optima. The emulated biological model for the Tocorime project provides a mechanism equivalent to *supersedure* in which only females of equal status compete in order to determine which female is allowed to mate. This aspect of the biological model has been added to this application in the form of an additional operator coupled with the traditional operators associated with evolutionary computations.

In the chosen biological system [wal00b], individuals migrate from one subpopulation to another because of crowding, changes in environmental conditions, limitations on colony activities, or because members become disoriented (see Tables 1 and 2). These external evolutionary factors have the benefit of providing diversity in the gene pool. The mating ritual associated with this biological model provides a built-in mechanism for incorporating a host of diverse genetic profiles via cross-species mating [sla98],[coe00] into existing and/or new colonies. The mating ritual for the chosen biological model occurs in designated mating areas referred to as *congregation areas*. This unique mating ritual has been simulated in a new genetic operator called *supersedure mating* which is used to maintain diversity within the simulated population of individuals. The evolutionary processes exhibited by the biological model not only provide a template for the storage, processing, and retrieval of valuable information, but also for Web scout probes/scouts/foragers that require an advanced communication system. The complexity of this project is not just in achieving a solution, but rather in evolving a neat, compact solution to the problem on a continuous basis.

2 Related Work

The parallel extension (PDBSCAN) [xjk99] of a sequential database system clustering algorithm (DBSCAN) was designed for knowledge discovery in spatial databases where the clusters may be of arbitrary shapes due to noisy databases. The authors provide a methodology that enables an efficient access to distributed data in a share-nothing architecture via the replication of indices by using a *data placement* strategy for the parallelization of clusters. This strategy assigns a Hilbert value to each data page according to its center of gravity and cluster pages based on the properties of the Hilbert curve. Chang et al. [ckss00] also incorporate the Hilbert value into methodology for retrieving, processing, and clustering multi-dimensional attribute spaces.

A cluster analysis methodology known as the Evolutionary Pre-Processor (EPrep) [dj97] used partitioning mechanisms to group individuals within a species (speciation) by automatically selecting features from a given data set. This methodology was used as a means of most efficiently avoiding local suboptimums resulting from uneven distribution of subpopulations across a domain. The process of speciation is accomplished via the sharing (fitness) function that computes the distance between two chromosome candidates/parents when the crossover operator is applied, thus ensuring mat-

0-7803-6657-3/01/$10.00 © 2001 IEEE

Table 1: Factors that determine evolutionary effect within the biological model.

	Causes of the evolutionary effect					
	Small nest cavity	**Crowding**	**Queen rearing**	**Unsatisfactory environmental conditions**	**Failure or death of queen**	**Colony activities hindered**
Migration	X	X		X		X
Swarming	X	X	X			X
Supersedure		X	X		X	X

Table 2: The biological model and its corresponding EC operators.

Cause	**Biological model**	**Evolutionary computation operators**	**Tocorime Apicu**
Individuals fly out	Migration/swarming	Migration	Speciation
Males/females mate	Reproduction/supersedure	Crossover/reproduction	Crossover/supersedure
Evolutionary changes	Mutation	Mutation/editing	Mutation/editing

ing between members of the same species – a process that produces a more efficient implementation. Here, the resulting GA operator was referred to as *mating restriction during evolution*, and the EPrep methodology was developed and applied to data sets with disparate rather than more singular measurements (such as an image or time series data sets).

The merits of a similarity metric for a distributed association mining application [po00] included its ability to assess the resulting partitions of a global dataset in order to compute the correlation of each subset. The computed correlation measures were used to compare the partitions in terms of how they are correlated – their adaptive *probe sets* (sets of the most common attributes) of the global dataset. This metric can be used to cluster similar datasets in order to perform meaningful distributed data mining. A GA feature selector [cq99] incorporated the concept of multiple correlation into the design of a classifier/indexer. The multiple correlation model was used to represent a multi-objective fitness function [coe00] needed by the GA to automate the classification system. The applicability of a multi-objective fitness function is due to its ability to measure classification performance as well as domain relations among two or more independent factors and a single (the feature related) criteria.

The Shifting Balance Genetic Algorithm (SBGA) [wo00] was developed as a means of providing subspecies with the ability to explore regions of the search space that seem most promising. Members of the subspecies were not treated as members of any of the "core" species but were allowed to migrate into a species based on their fitness resulting from individual fact-finding explorations of the disparate regions of the solution space. The resulting subspecies from this application do not include individuals that are near each cluster's boundaries. This, in turn, reduces the limitations of speciation [sou00],[caws00] in which each individual is restricted to a limited region of the search space.

Implementation of a parallel, genetic algorithm on a hypercube employed a methodology [tan87] that overcomes the time limitations associated with a large population size. This methodology required two genetic operators, crossover and mutation, that were controlled by Poisson frequencies. The *crossover-rate* for any chosen pair of individuals was a Poisson-distributed random variable that measured the average number of crossovers per mating. The *mutation-rate* for each individual resulted from the $mutation\text{-}rate_{trad}/length\_of\_chromosome$, whereas the traditional mutation rate, $mutation\text{-}rate_{trad}$, represented the average number (Poisson distributed) of mutations per individual. This methodology provides a foundation for developing a species by restricting the application of genetic operators only to subpopulations (as opposed to the whole population). The creation of a multi-dimensional population (subpopulation) space incorporates a mechanism that reduces the effect of premature convergence to a local optima – this occuring more readily with a one-dimensional population space. This approach can produce robust generations when different nodes use different (rate) parameter values without determining the best possible parameter values.

3 New Genetic Operator – Supersedure Mating

The application of methodologies associated with the editing, mutation, and crossover operators for GA/GP hybrid chromosomes can be simulated via the *supersedure* operator. The crossover operator can be applied multiple times where one female (even-numbered node ID) chromosome mates with one or more male (odd-numbered node ID) chromosomes – thus implementing a hybrid of the crossover and mutation operators, which results in *dissassortive* mating [ros96]. The use of this operator can result in *chromosome aberrations* [bis98] which account for the changes in various chromosome locations – whole chromosomes, or sets of genetic components. This new operator adds random noise to the whole process, a phenomenon that can be beneficial in the prevention of premature convergence and incorporation of various aspects of the methodologies such as supersteps and dissassortive mating, when selecting individual chromosomes from different species. Supersteps [dk96],[dk97] resulted from the application of two or more GP functionality components being sequentially applied as one operation via supersedure mating. Dissassortive mating results from selecting parents for the crossover operator from a disjoint list of individuals (species). The supersedure operator will:

1. Randomly select a female chromosome for mating.
   (a) Randomly select a male chromosome for mating.
   (b) Randomly select a genetic range.
   (c) Perform crossover on the two randomly selected parents.
2. If necessary, repeat steps 1a through 1c.

The traditional GA/GP, as well as the normal (steady-state) approach for the crossover operator, is restricted to one application per generation for a single set of chromosomes. The Tocorime Apicu application of the crossover operator will restrict mating to members of the same species with distinct genders. The individuals selected to mate are chosen via the proportional fitness method. The simulated mating ritual enhances the competition methodology as well as the quality of the total population via modification of several male chromosomes from any species group by using an operator with a mating selector heuristic to facilitate cross-species mating – thus representing a form of editing. The goal of this operator is not the survival of a chromosome but simulation of the actual biological mating process. This implies that the chosen individuals are selected to mate regardless of their fitness measure.

The selection of individuals allowed to participate in supersedure mating occurs via two "popular" selection methodologies: 1) the proportional fitness or roulette wheel selection, and 2) the tournament selection. The proportional fitness method assigns a random number to each individual and repeatedly selects individuals for mating. The individual with the selected random number is allowed to mate. The tournament selection method randomly chooses individuals who compete for the right to mate. The process of supersedure occurs when two or more females are within a chosen (fitness) tolerance

$$similarity(female_i, female_j) < tolerance, \quad for\ all\ i \neq j. \tag{1}$$

A randomly selected female node is allowed to mate a random number of times. This provides the female chromosome with a biological emulator of the mutation operator which occurs if a selected male chromosome was previously selected during the current application of this regulatory operator. The biological emulator aspect of this operator is the result of a male chromosome being selected only once during the application of the operator. However, the same male may be chosen one or more times. This operator provides a dynamic regulatory mechanism that regulates the simultaneous application of the crossover-rate and/or mutation-rate.

4 Tocorime Apicu Approach

The approach taken in this project does not focus on the evolution of populations of programs, but instead on the efficient growth and emulation of chromosome data structures and their evolution. The focus of typical GP applications reflects the growth of program size which represents a maturing population. The typical GP runs can suffer from the "bloating" phenomenon [alt94],[ros96] which occurs when the population of solutions (programs) grows following a repeated application of the GP operators. The bloating phenomenon is needed in this study for the representation of chromosome structures since it can be viewed as a component in the evolution of a species. The bloating phenomenon reads as an uneven distribution of Web documents in this application and can occur during any generation. Use of the chromosome structure emulates the technique of Web page (process) migration [dqp00] – this being a mechanism used to balance distribution of tasks related to clustering Web documents. Some GA and GP applications impose a "parsimony" condition in the form of parsimony pressure [dk97],[kc00],[caws00], which indicates that bloating has occurred, and a negative weight is added to the fitness measure for this particular individual.

Use of the chromosome data structure emulates the technique of process migration as a method for balancing the distribution of tasks. The migration operator was applied to the transfer of a set(s) of genetic components of randomly selected Web documents during the application of the crossover operator. The hybrid chromosome structure associated with the project's indexer [wal00a],[wal01] emulates the methodologies of GA and GP. Initially, a single structure was used to represent subsets (subpopulations) of Web documents that reside at each node (Web site). However, this approach was expanded to an allocation of two or more pairs of chromosomes per node, thus simulating a double-stranded RNA genome [gdg+99]. Each strand of genes that reside on each $Node_i$ (Web site) can be viewed as a set of the genetic components of an individual member of a simulated species.

The use of gender [coe00] in this parallel clustering system provided a mechanism for automatically generating species within the total population. Use of the fitness measures to generate species results in the emergence of subspecies within a specie since gender diversity is maintained. The classification process of species [sz98] in this project was based on aspects of: 1) identity-by-state (IBS) since the parents are subsequently replaced by their offspring, and 2) identity-by-descent (IBD) which exists in nature where parents coexist with their respective offspring(s). The gender of a chromosome (node) remains unchanged during execution of the Tocorime Apicu clustering system, thus resulting in a simulated version of IBD (the child retaining the sex of the parent). A generalized form of IBD was implemented as the adaptive operator probabilities (ADOPP) mechanism [tedh00] which assigns a credit to each crossover and mutation operator that contributes to a genetically im-

Table 3: The GA/GP hybrid parameters.

Parameter	Version A	Version B	Version C	Version D
Population size (Web documents)	1024	1024	1024	1024
Max number of generations	200	200	$\leq$ 200	$\leq$ 200
Chromosome length	1024	1024	1024	1024
Min/Max size of each genetic component	0/Unlimited	0/Unlimited	0/Unlimited	0/Unlimited
Number of transmitted components	Unrestricted	Restricted to 1	Restricted to 16	Restricted to 1
Crossover types	One/two point	One/two point	One/two point	One/two point
Supersedure rate Random mating rate Speciation rate	N/A 100% N/A	N/A 100% N/A	Variable rate Variable rate Variable rate	Variable rate Variable rate Variable rate
Probe set	16 words/ Static	16 words/ Static	16 words/ Dynamic	16 words/ Dynamic
Fitness calculations		Inverted (Out-of-core processing)	Inverted (Out-of-core processing)	Inverted (Out-of-core processing)

proved offspring over a series of successive generations.

5 Model of the Parallel Clustering System

The goals of the knowledge discovery in databases (KDD) model are to obtain a better understanding of an ever-changing environment in order to find various types of *hidden* knowledge based on the complexity of the implemented model. The system requirements for the Web page clustering mechanisms of this evolutionary model place numerous demands on each node, as well as on the file server. Each word within each Web page may be compared with millions of other words extracted from Web documents that may eventually reside on a node over a period of generations following the application of genetic operators. Application of the GP operators, as well as intermediate application of the fitness measures, are used to recalculate the normalized fitness of each node. System requirements can be reduced in multiple ways [cec00],[szat00],[sla98], such as: 1) removal of noise in each Web page which, without losing pertinent information, reduces the signal-to-noise ratio, as opposed to using the raw Web page, 2) reducing the computational complexity associated with applying the query techniques of KDD, and 3) the non-genetic transmission of cultural information. The removal of noise in this context refers to the KDD process of data cleaning and represents a crucial component of data preparation. This process may incorporate the removal of duplicate Web documents as well as comments, stop words, non-stop words, non-contributory tags and their attributes, non-contributory attributes, and possibly corrects typographical errors such as commonly misspelled words.

Adaptive probe sets [po00] were used with high-performance KDD application to incorporate the most commonly occurring information items using the technique that determines the *influential attributes* of a given data set. This approach was used in Versions C and D of the clustering system (see Experimental Results). Adaptive techniques are required in order to adequately classify/cluster the contents of selected Web documents across the $n-1$ nodes in this high-performance implementation. The set of search strings associated with this application can have as many as $500,000$ keywords [cmf99]. This classification process also assists in that evolutionary process of forming species via the fitness measures [wal00a]. The stochastic nature [az96] of the evolutionary operators and the KDD model associated with this GA/GP application requires the repeated application of operators over a period of generations in order to obtain reliable results, which would then generate solutions that are neat as well as compact.

6 Experimental Results

6.1 Application of Genetic Programming Fitness Measures

Several versions of the clustering system were studied to improve performance of the initial implementation [wal01]. The initial implementation only provided timing studies associated with transcribing 512 Web documents. The current versions were restricted to 1024 Web documents which, in turn, were used to create adaptive probe sets that were distributed to each indexer node. The probe sets were used to compute the fitness measures associated with each indexer node, which in turn were used to create sub-species for the purpose of applying the new and traditional GA/GP operators. The distinct versions of the clustering system are shown in Table 3. Versions A and B are similar, with the exception of restricting the size of genetic components transferred during application of the crossover operator. Version A did not impose any size

Table 4: Best case crossover/mutation rates for 1024 Web documents.

		Number of nodes in subcluster							
Version	Types of mating	1	3	4	5	6	7	8	9
A	Random	N/A	100%	100%	100%	100%	100%	100%	100%
B	Random	N/A	100%	100%	100%	100%	100%	100%	100%
C	Supersedure	N/A	0%	0%	6.5%	13.0%	5.0%	15.5%	19.5%
	Random	100.0%	94.5%	84.0%	72.0%	67.0%	34.0%	27.5%	21.0%
	Speciation	N/A	5.5%	16.0%	21.5%	20.0%	61.0%	57.0%	59.5%
D	Supersedure	N/A	0.0%	0.0%	2.0%	3.5%	23.4%	14.5%	7.0%
	Random	100.0%	98.0%	83.5%	84.5%	31.0%	19.4%	28.0%	9.0%
	Speciation	N/A	2.0%	16.5%	13.5%	65.5%	57.2%	57.5%	84.0%

restrictions. This resulted in enormous loads on the file server as the number of nodes were increased in the cluster.

The probe sets associated with Versions A and B were chosen using the AWK programming language [akw88] and reflected the most commonly occurring words. When performing a comparison between the computed fitness for these versions, the fitness measures reflect this trend. However, the probe sets associated with Versions C and D were chosen randomly (for each generation) which resulted in some of the fitness measures computing a minimum value of 0.0.

The clustering systems parameters shown in Table 3 show that Versions A and B are very similar, as are Versions C and D. The differences between Versions A and B are: 1) the number of the transmitted genetic components, and 2) the methodology used to compute the fitness measures. The number of transmitted genetic components was initially unlimited, which in the worst case could result in the transmission of 1024 genetic components among two chromosomes (nodes) selected for sexual reproduction. This worst-case scenario results in a saturated file server when mating is in the form of either supersedure or speciation. Both of these cases can facilitate the mating of several male/female pairs. Version B limited the number of transmitted components to 1. The crossover points associated with all forms of sexual reproduction were based on randomly selecting a starting position and incrementing this starting position by either a randomly chosen number (Version A), 1 (Versions B and D), or 16 (Version C only). The transmission of all 1024 components of selected nodes would result in *zero*-point crossover; this is rare but possible in Version A since the starting position and the number of components were randomly selected.

The required execution time between the two versions differed by an approximate factor of 5 hours which reflects a reduced file server workload. This reduction in the number of transmitted genetic components will also reduce the possibility of applying the crossover and mutation operators which result in sharp oscillations in fitness measures when the number of Web documents is drastically increased from 1024. The methodology used to compute the fitness measure [wal00a] was based on: 1) transcribing the raw Web page, and 2) computing the *raw fitness*, *standardized fitness*, and *adjusted fitness* (the *normalized fitness* in Versions C and D). The raw fitness measures were computed by applying the first word in the probe set to the 1024 files which formed the training set, followed by applying the second word in the probe set to the 1024 files, and then repeating this process for the remaining words in the probe set. The inverted approach resulted in the application of probe strings to each file, this process in turn reducing the workload on the file server. Restricting the number of transmitted genetic components and the inverted application of the probe set resulted in out-of-core (OoC) [blo+00] reads/writes which facilitate the use of each node's hard drive as an extension of its cache. The normalized fitness was used in the similarity test in order to cluster the chromosomes (nodes) into species.

The maximum number of generations for all of the studies was 200. Versions A and B used only random mating which required 200 generations each. Random mating was implemented as follows: 1) the first parent is randomly chosen, and 2) the second parent is chosen by incrementing the node ID of the first parent, thereby mimicking the ring communication pattern based on the MPI rank in order to determine adjacent nodes. Recall that the even-numbered nodes represented female chromosomes and the odd-numbered nodes represented male chromosomes (excluding node 0 which is the indexer program manager). Crossover was applied to the selected chromosomes for each generation. Versions C and D required a reduced number of generations since the supersedure and speciation operators were allowed to use supersteps resulting from multiple applications of the crossover operator. The number of Web documents comprising each genetic component was within the range of 0 to ∞. The maximum number of genetic number components allocated to each gene (genetic component) was initially limited by the amount of available dynamic memory. An OoC methodology was used in order to circumvent the limitations associated with the dynamic allocation of memory based on the size of the heap and stack allocated at runtime.

Versions A and B relied on a set of 16 strings stored in a static probe set in order to compute the associated fitness measures for each Web page. This approach was adequate for the development of the clustering system, but is not deemed appropriate for real world applications. A static probe set, which does not have adequate *a priori* knowledge about randomly chosen Web documents that are needed to supplement the training set, must be developed by a human. The static probe set was created by the AWK programming language and reflected the 16 most commonly occurring strings within the 1024 transcribed Web documents.

Versions C and D relied on the set of 16 strings resulting from the use of dynamic probe sets in order to compute the associated fitness measures for each Web page. This approach reflects the true nature of the test (beta) version of the experimental search engine.

The dynamic probe sets are created by randomly selecting a node, and then deriving a set of 16 search strings from its collection of words. The set of derived words are distributed among the indexers within the cluster in order to determine which of the three types of mating will occur: 1) speciation , 2) supersedure, or 3) random mating. The use of static probe sets to determine the type of mating is not effective when the number of genetic components being transmitted is 1. The dynamic probe set compensates for the apparent steady-state solutions resulting from transmitting small quantities of genetic components.

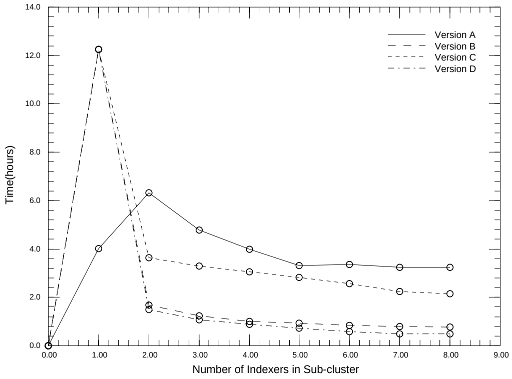

Figure 1: Timing results for the distinct versions of the clustering system.

6.2 Timing Results

The best case results associated with the sequential and parallel application of the evolutionary operators are shown in Table 4. The corresponding best-case sequential and parallel timing results are presented in Figures 1 to 3. The sequential results found in Version A reflect the oscillatory nature associated with using a training set of 512 pages coupled with a 1024 page pseudo-chromosome. These results are represented as percentages. This table shows the four versions in this study along with the application of the various evolutionary mating strategies (speciation, supersedure, and random mating). The execution time for Version A displayed a decrease in the time needed to parse and apply the crossover operator. The required execution time became relatively constant, starting with 6 nodes when the number of nodes was increased. The timing and efficiency patterns exhibited by Version A reflect the need to study the scalability associated with an increase in the number of Web documents and to provide the indexer manager (node 0) with more search engine related responsibilities. Version A results reflect the impact of the file server's saturation due to the non-inverted application of probe sets which shuffle all the pages in order to compute the fitness measures, as well as the random selection and transmission of an unknown amount of genetic material. These timing results showed that approximately 4.0 hours were required by sub-cluster sizes of 1 and 5 nodes, while the 4- node sub-cluster required approximately 4.75 hours. The sub-cluster of 3 nodes required approximately 6.25 hours, and the remaining sub-clusters required approximately 3.25 hours. The sequential version associated with B, C, and D required approximately 12.25 hours.

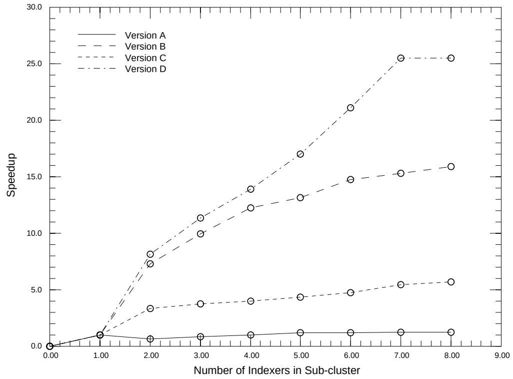

Figure 2: Speedup for the distinct versions of the clustering system.

OoC processing provided no benefit to the sequential versions. Sequential results showed that requiring each node to process 1024 Web documents impacts the efficiency of the file server which supports n nodes in its corresponding sub-cluster. Version B reveals the benefits of OoC processing. The 3-node sub-cluster required approximately 1.75 hours, with the 4-node sub-cluster requiring approximately 1.25 hours. The 5- and 6-node sub-clusters required approximately 1.0 hours, and the remaining sub-cluster required approximately 0.75 hours. This version showed consistent decreases in the required execution time, but these decreases were not drastic. Similar results were generated for Versions C and D. Speedup has resulted from fitness-enhancing mechanisms that provide information results from previous fitness measurements of previous generations, such as the non-genetic transmission of cultural information.

7 Future Work/Expansion of the Biological Model

The determination of an adaptable update criteria for the inclusion of new Web documents on a continous basis will be needed. One proposed approach includes the case where the crossover rate falls to 10% of the total crossover events for

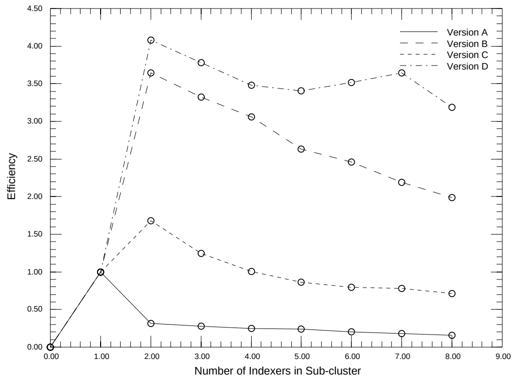

Figure 3: Efficiency for the distinct versions of the clustering system.

any given generation. Additional studies will be conducted to determine the effect of various computing architectures such as IBM SP2 clusters and Beowulf clusters. The HyperText Markup Language (HTML) parser needs to be expanded to incorporate additional Web page formats. The effects of the file server are apparent in the results presented in this paper. The addition of multiple file servers to each cluster reduces the simultaneous page request on each server. Expansion of the single cluster environment into a multiple cluster environment will provide additional computing/storage resources. This addition will result in increased complexity of the simulated biological model, a component of the evolutionary aspects of the true biological system.

8 Conclusion

The combination of diverse methodologies for the implementation of this experimental search engine has resulted in the early-stage development of an adaptable search engine and in an efficient parallel clustering system. The current approach will be compared to other clustering approaches in order to enhance the methodologies that were incorporated in this work. The evolutionary preservation mechanisms of the biological model allow it to adapt to various external factors which are reflected in the ability of an existing species to grow/shrink. The effect of this process was facilitated through the use of GA/GP methodologies which allow: 1) individual chromosomes to increase/decrease in size, and 2) individuals to migrate from one subspecies clustering to another. Another variation of species expansion can be reflected in the ability of an adaptable model to incorporate computer environments consisting of additional networks of workstations and their respective file servers in order to form new groupings which emulate the evolutionary swarming process of the biological model.

9 Acknowledgements

The author wishes to thank Walter Karplus and Zhen-Su She for their direction and suggestions. The author acknowledges Elias Houstis, Ahmed K. Elmagarmid, Apostolos Hadjidimos, and Ann Catlin for their support, encouragement, and access to the computer systems at Purdue University. Special thanks to the reviewers for their helpful suggestions. Support for this work came from the Raytheon Fellowship Program, Tapicu, Inc., and Honeybee Technologies. Plots were generated using *PSplot* written by Paul Dulaney (Raytheon Corporation). The implementation results associated with the network of workstations originated on computers located in the Department of Computer Sciences, Purdue University, West Lafayette, IN.

Bibliography

[akw88] (1988) Aho, A.V., Kernighan, B.W. and Weinberger, P.J. *The AWK Programming Language*, Addison-Wesley.

[alt94] (1994) Altenberg, L. “Emergent Phenomena in Genetic Programming,” in *Proc. of the 3rd Conf. on Evolutionary Programming*, World Scientific Press, Singapore.

[az96] (1996) Adriaans, P. and Zantinge, D. *Data Mining*, Addison-Wesley, Harlow, England.

[bfn96] (1996) Banzhaf, W., Francone, F.D. and Nordin, P. “The Effect of Extensive Use of the Mutation Operator on Generalization in Genetic Programming Using Sparse Data Sets,” in *Proc. of the 5th Conference on Parallel Problem Solving from Nature*, Springer-Verlag, Berlin, Germany.

[bis98] (1998) Bishop, M.J. “Comparative Mapping in Humans and Vertebrates,” in *Guide to Human Genome Computing (2nd edition)*, ed. M.J. Bishop, Academic Press, San Diego, CA.

[blo+00] (2000) Baraglia, R., Laforenza, D., Orlando, S., Palmerini, P. and Perego, R. “Implementation Issues in the Design of I/O Intensive Data Mining Applications on Clusters of Workstations,” *LNCS 1800*, Springer-Verlag, Berlin Heidelberg New York.

[caws00] (2000) Collins, D.J., Agah, A., Wu, A.S., and Schultz, A.C. “The Effects of Team Size on the Evolution of Distributed Micros Air Vehicles,” in *Proc. of GECCO-2000*, Morgan Kaufman Publishers, Inc., San Francisco.

[cec00] (2000) Costa, M.C.A. and Ebecken, N.F.F. “Data Mining High Performance Computing using Neural Networks,” in *Applications of High-*

Performance Computers in Engineering VI, WIT Press, Ashurst, Southampton, UK.

[ckss00] (2000) Chang, C. and Kurc, T., Sussman, A. and Saltz, J. "Optimizing Retrieval and Processing of Multi-dimensional Scientific Datasets," in *Proc. of IPDPS 2000*, IEEE Press, Los Alamitos, CA.

[cmf99] (1999) Chen, Z., Meng, X. and Fowler, R. "Searching the Web with Queries," *Knowledge and Information Systems*, **1(3)**, 369-375.

[coe00] (1999) Coello, C.A.C. "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, **1(3)**, 269-308.

[cq99] (1999) Chaikla, N. and Qi, Y. "Feature Selection Using the Domain Relationship with Genetic Algorithms," *Knowledge and Information Systems*, **1(3)**, 377-390.

[dj97] (1997) Duda, J.W. and Jakiela, M.J. "Generation and Classification of Structural Topologies with Genetic Algorithm Speciation," *Journal of Mechanical Design*, **119**, 127-131.

[dk96] (1996) Dracopoulos, D.C. and Kent, S. "Bulk Synchronous Parallelisation of Genetic Programming," in *Proc. of PARA'96*, Springer-Verlag, Berlin, Germany.

[dk97] (1997) Dracopoulos, D.C. and Kent, S. "Genetic Programming for Prediction and Control," *Neural Computing & Applications*, **6(4)**, 214-228.

[dqp00] (2000) Dantas, M.A.R., Queiroz, W.J. and Pfitscher, G.H. "An Efficient Threshold Approach on Distributed Workstation Clusters," in *Proc. of HPC 2000*, SCS Press, San Diego, CA.

[gdg+99] (1999) Gouet, P., Diprose, J.M., Grimes, J.M., Malby, R. and Burroughs, J.N., Zientara, S., Stuart, D.I. and Mertens, P.P.C. "The Highly Ordered Double-Stranded RNA Genome of Bluetongue Virus Revealed by Crystallography," *Cell*, **97**, 481-490.

[kc00] (2000) Knowles, J. and Corne, D. "Heuristics for Evolutionary Off-line Routing in Telecommunications Networks," in *Proc. of GECCO-2000*, Morgan Kaufman Publishers, Inc., San Francisco.

[po00] (2000) Parthasarathy, S. and Ogihara, M. "Exploiting Dataset Similarity for Distributed Mining," in *LNCS 1800*, Springer-Verlag, Berlin Heidelberg New York.

[ros96] (1996) Rosca, J.P. "Generality versus Size in Genetic Programming," in *Proc. of the 1st Genetic Programming Conf.*, MIT Press, Cambridge, MA.

[sla98] (1998) Slater, G.S.C. "Human EST Sequences," in *Guide to Human Genome Computing (2nd edition)*, ed. M.J. Bishop, Academic Press, San Diego, CA.

[sou00] (2000) Soule, T. "Heterogeneity and Specialization in Evolving Teams," in *Proc. of GECCO-2000*, Morgan Kaufman Publishers, Inc., San Francisco.

[sz98] (1998) Sham, P. and Zhao, J. "Linkage Analysis Using Affected Sib-Pairs," in *Guide to Human Genome Computing (2nd edition)*, ed. M.J. Bishop, Academic Press, San Diego, CA.

[szat00] (1999) Srivastava, A., Han, E., Kumar, V. and Singh, V. "Parallel Formulations of Decision-Tree Classification Algorithms," *Data Mining and Knowledge Discovery*, **3(3)**, 237-261.

[tan87] (1987) Tanese, R. "Parallel Genetic Algorithm for a Hypercube," in *Proc. of the 2nd International Conf. on Genetic Algorithms*, Lawrence Erlbaum Associates, Hilsdale, New Jersey.

[tedh00] (2000) Testa, L.J., Esterline, A.C., Dozier, G.V. and Homaifar, A. "A Comparison of Operators for Solving Time Dependent Traveling Salesman Problems Using Genetic Algorithms," in *Proc. of GECCO-2000*, Morgan Kaufman Publishers, Inc., San Francisco.

[wal00a] (2000) Walker, R.L. "Development of an Indexer Simulator for a Parallel Pseudo-Search Engine," in *Proc. of HPC 2000*, SCS Press, San Diego, CA.

[wal00b] (2000) Walker, R.L. "Dynamic Load Balancing Model: Preliminary Assessment of a Biological Model for a Pseudo-Search Engine," *LNCS 1800*, Springer-Verlag, Berlin Heidelberg New York.

[wal01] (2001) Walker, R.L. "Search Engine Case Study: Searching the Web Using Genetic Programming and MPI," *Parallel Computing*, **27(1/2)**, 71-89.

[wo00] (2000) Wineberg, M. and Oppacher, F. "Enhancing the GA's Ability to Cope with Dynamic Environments," in *Proc. of GECCO-2000*, Morgan Kaufman Publishers, Inc., San Francisco.

[xjk99] (1999) Xu, X., Jager, J. and Kriegel, H. "A Fast Parallel Clustering Algorithm for Large Spatial Databases," *Data Mining and Knowledge Discovery*, **3(3)**, 263-290.