CLUSTERING HERDS TO MINIMIZE MEMORY REQUIREMENTS OF ANIMAL MODEL PROGRAMS

I. MISZTAL
Department of Animal Sciences, University of Illinois
Urbana, Illinois 61801, USA

SUMMARY

In computer programs for calculating animal model solutions, the memory requirement is 4 to 5 times smaller if the equations are ordered by effects nested in herds and not by factors. This study investigated further memory savings by ordering by clusters of herds that exchange animals. The proposed clustering algorithm resulted in additional memory savings of up to 39%. Using effective ordering and computers with large virtual but small real memory, large problems can be handled effectively by simple programs or by general computer packages.

INTRODUCTION

Straightforward implementation of computer programs for large, animal model evaluation systems requires much memory and disk space. Although the disk space needed can be reduced by application of Jacobi iteration and iteration "on data" (Misztal and Gianola, 1987), the memory requirement is affected significantly by the ordering of equations. In ordering by effects such as in these mixed model equations ordered by factors:

$$\begin{bmatrix} \mathbf{H'H} & \mathbf{H'B} & \mathbf{H'Z} & \dots \\ \mathbf{B'H} & \mathbf{B'B} & \mathbf{B'Z} & \dots \\ \mathbf{Z'H} & \mathbf{Z'B} & \mathbf{Z'Z}+\mathbf{G}_z & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \hat{\mathbf{h}} \\ \mathbf{b} \\ \mathbf{z} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{H'y} \\ \mathbf{B'y} \\ \mathbf{Z'y} \\ \vdots \end{bmatrix}$$

computer memory should be sufficiently large to contain all solutions because solutions are accessed during iteration almost in random order and using disk storage for some solutions would lead to serious inefficiency. Access to solutions is localized if the system of equations is reordered to a doubly-bordered block-diagonal form (Duff *et al.*, 1989) such as:

$$\begin{bmatrix} \mathbf{B}_1 & & & \dots & \mathbf{O}_1' \\ & \mathbf{B}_2 & & \dots & \mathbf{O}_2' \\ & & \mathbf{B}_3 & \dots & \mathbf{O}_3' \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{O}_1 & \mathbf{O}_2 & \mathbf{O}_3 & \dots & \mathbf{C} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \vdots \\ \mathbf{r}_c \end{bmatrix}$$

For example, blocks $\mathbf{B}_i$ could correspond to effects nested in one or more herds, and block C could correspond to effects of animals connecting the blocks. In the doubly-bordered block-diagonal form, only solutions for two blocks (the one being processed plus the last one) need to remain in memory. Other solutions can be stored on disk and retrieved if necessary. If large virtual memory is available, solving a doubly-bordered block-diagonal system is efficient as long as solutions for c and the largest $\mathbf{b}_i$ fit in the physical memory. Consequently, simple programs that require much memory can be used.

General methods to reorder arbitrary matrices to a doubly-bordered block-diagonal form, e.g., one-way dissection or nested dissection (George and Liu, 1981), are too demanding computationally. Blocking by effects nested within single herds currently is used for U.S. evaluation of dairy cattle (Wiggans *et al.*, 1988a). Blocking by effects nested within clusters of herds can provide a reduction in memory requirements because the blocks absorb some across-herd connections and c is smaller. Wiggans *et al.* (1988b) described such clustering based on proximity of herd codes. Clusters of herds were called superherds and animals connecting superherds, tie animals. Clusters obtained this way possibly could be improved by considering actual connections between herds because herds exchanging animals do not necessarily have similar herd codes.

This study investigated the potential of clustering in reducing memory requirements. A clustering algorithm that takes into account connections between herds was developed and evaluated.

DATA AND METHODS

Two data sets were used. The first contained production records of 129,396 Ayrshire cows with first lactations reported in 5299 herds. In addition, pedigree information was available for 12,710 dams without records and 7734 sires, for a total of 149,840 animals. The largest herd had 1263 animals. A subset of this data set excluded records from later herds for cows that changed herds, which reduced the number of herds to 4850, the largest with 1250 animals. The second data set consisted of 2.08 million first records for final score of registered Holstein cows in 33,009 herds. The largest herd had 7371 animals. Pedigree information was available for 196,638 dams and 113,384 sires, for a total of 2.39 million animals.

The problem of clustering herds into blocks of limited size can be viewed as partitional clustering (Jain and Dubes, 1988), which does not have optimal algorithms for realistic problems. A heuristic algorithm that was developed especially for this application follows.

Clustering algorithm

The algorithm clustered n animals with maximums set for number of clusters of herds (c_{max}) or number of animals per cluster (a_{max}) so that as many interherd connections as possible were absorbed by clustering. To reduce computations, only connections by animals present in exactly two herds were considered. The algorithm operated iteratively. Initially, all clusters were empty. During each round, every herd was either assigned to a cluster, reassigned to another cluster, or stayed in the same cluster according to the following rules:

1) Assignment to permissible cluster with largest number of connections with herd.

2) If no connections with permissible cluster, assignment to least populated cluster.

Permissible clusters were those that would not overfill after adding the herd, and the number of connections with a cluster was the number of connections with all herds in that cluster. Iteration stopped when no herd changed clusters during a round. Initial runs determined that a good choice for c_{max} was $1.3(n/a_{max})$. Also, efficiency was higher if a_{max} was reduced by 20% in the first round.

RESULTS

Distribution of animals with records or progeny in a given number of herds is in Table 1. For both data sets, only 17 to 25% of animals connected herds, and only 4 to 8% connected three herds or more. Results for the Ayrshire production subset (first herds only) were similar to those for the Holstein type data set, which included only records from first herds. Percentages of animals present in only one cluster after clustering either by proximity of herd code or by connections between herds (algorithm) are in Table 2. For both clustering methods, the number of eliminated connections was

between 46 to 77% of all two-herd connections, with the smallest percentage for Holsteins. The algorithm performed better than did assignment by herd codes for all but the largest cluster size for the full Ayrshire data set. The algorithm's performance also was less dependent on cluster size. Relative memory requirement was defined as number of connection animals plus cluster size (or number of animals in the largest herd) divided by total number of animals (Table 3). Without clustering but with ordering in doubly-bordered block-diagonal form, relative memory requirement was .171 for Holstein type, .187 for Ayrshire first herd production, and .262 for Ayrshire all herd production. Using clustering by herd codes, memory requirements were reduced by 10% for Holsteins and 19% (first herd) and 25% (all herds) for Ayrshires; reductions after clustering by the algorithm were 27, 33, and 39%. With the algorithm, optimum cluster sizes were the smallest, which were only slightly larger than the numbers of animals in the largest herds.

Table 1. Percentage of animals connecting a given number of herds by data set.

Data set	Number of herds connected					
	1	2	3	4	5	>5
Holstein type	83.2	13.0	2.5	.5	.2	.6
Ayrshire production						
First herd only	82.1	13.7	2.8	.6	.2	.6
All herds	74.6	17.2	5.0	1.6	.7	.9

Table 2. Percentage of animals without connections outside clusters for a given maximum number of animals per cluster (a_{max}) by data set and clustering method.

Data set	Clustering method[1]	a_{max}						
		NC[2]	2000	5000	10,000	20,000	50,000	100,000
Holstein type	Herd codes	83.2	...	...	84.3	85.2	86.7	88.0
	Connections	...	...	...	87.9	88.2	88.7	89.2
Ayrshire production								
First herd only	Herd codes	82.1	86.1	87.6	89.1	91.2	...	...
	Connections	...	88.8	89.5	90.2	91.3	...	...
All herds	Herd codes	74.6	81.4	83.6	85.8	88.7	...	...
	Connections	...	85.4	86.2	86.6	87.9	...	...

[1]Clustering was based on either proximity of herd code or number of connections between herds (algorithm); [2]NC = no clustering.

Table 3. Relative memory requirement[1] for a given maximum number of animals per cluster (a_{max}) by data set and clustering method.

Data set	Clustering method[1]	a_{max}						
		NC[2]	2000	5000	10,000	20,000	50,000	100,000
Holstein type	Herd codes	.171	...	...	.161	.156	.154	.162
	Connections	...	...	...	.125	.126	.134	.150
Ayrshire production								
First herd only	Herd codes	.187	.152	.157	.176	.221	...	...
	Connections	...	.125	.138	.164	.220	...	...
All herds	Herd codes	.262	.199	.197	.209	.246	...	...
	Connections	...	.159	.171	.201	.254	...	...

[1]Equals 1 if equations ordered by factors; [2]NC = no clustering.

DISCUSSION

Theoretically, if clustering herds resulted in almost complete elimination of two-herd connections and partial elimination of higher order connections, the memory requirement could drop to less than 5% of that for the unblocked case. In fact, elimination was not more than 77% of two-herd connections for Ayrshires and was only 46% for Holsteins. Correspondingly, memory requirements did not drop below 12.5% of that for the unblocked case. This can be explained by three reasons. First, the algorithm was not optimal. Many modifications to the algorithm were investigated (including taking higher order connections into account), but no further reduction in memory requirements resulted. Second, the data sets were only subsets of real populations and contained mostly registered animals for Ayrshires and only registered animals for Holsteins. Such animals are used more frequently as breeding material and subsequently connect herds more often. Third, the exchange of genetic material with many herds and over large areas occurs more often than expected.

Programs using the doubly-bordered block-diagonal form with or without herd clustering could use up to 8 times less memory. Without extensive programming, a computer with a 16-Mbyte physical memory and using second-order Jacobi iteration could compute solutions for 8 million animals without clustering and for 12 million animals with clustering. This assumes a model as in Wiggans *et al.* (1988a) and that records of cows in later herds are not considered.

ACKNOWLEDGMENTS

This research was supported by a grant from the Holstein Association, Brattleboro, Vermont. Computer support on the Cray 2 system by the National Center for Supercomputer Applications at the University of Illinois at Urbana-Champaign is acknowledged gratefully as is assistance by T.J. Lawlor, Jr., R. Tempelman, G.R. Wiggans and S.M. Hubbard.

REFERENCES

DUFF, I.S., ERISMAN, A.M. and REID, J.K. 1989. Direct methods for sparse matrices. Clarendon Press, Oxford, 341 pp.

GEORGE, A. and LIU, J.W. 1981. Computer solution of large sparse positive definite systems. Prentice-Hall, Inc., Englewood Cliffs, NJ, 324 pp.

JAIN, A.K. and DUBES, R. 1988. Algorithms for clustering data. Prentice Hall, Englewood Cliffs, NJ, 320 pp.

MISZTAL, I. and GIANOLA, D. 1987. J. Dairy Sci. 70: 716-723.

WIGGANS, G.R., MISZTAL, I. and VAN VLECK, L.D. 1988a. J. Dairy Sci. 71(Suppl. 2): 54-69.

WIGGANS, G.R., MISZTAL, I. and VAN VLECK, L.D. 1988b. J. Dairy Sci. 71: 1319-1329.