ACTA *Acta Sci. Pol., Zootechnica 6 (4) 2007, 29–36*

A STUDY ON EFFICIENCY OF RECURSIVE ALGORITHM FOR ESTIMATING RELATIONSHIP COEFFICIENTS

Maciej Gierdziewicz, Joanna Kania-Gierdziewicz

Agricultural University of Krakow

Abstract. The aim of the work was to design a computer program which would be able to calculate relationship coefficients of a population or a subpopulation in the shortest possible time, applying directly the formula used in tabular method without creating any – permanent or temporary – relationship matrices. Data were pedigrees of 25 036 Polish Black and White bulls born in the years 1960–2000. The number of all animal ID's found in the pedigrees was 63 264. From these data the input data set for the computer program has been created. The data set contained pedigrees of the form „animal-sire-dam", sorted in chronological order. The number of all possible animal pairs was 2 001 135 216. The algorithm was tested on four different computers, three of them having multiprocessor architecture. The method used to calculate inbreeding and relationship coefficients involved recursive relationship function calls. The recursive method let not only cut the execution time of the program 10.7 to 18 times, but let prepare the programs for parallelization which was necessary to shorten computing time and to enable more complicated pedigree analysis and calculation of wide variety of statistics.

Key words: parallel computing, recursive algorithm, relationship coefficient

INTRODUCTION

For computing inbreeding and relationship coefficients and, consequently, the relationship matrix, of a large population, using the pedigree file of that population, the tabular method was often a standard way of performing calculations. This has been proposed by Cockerham [1954] and further developed by Henderson [1975, 1976] as a consequence of the need of processing large pedigree data sets. The challenge – e.g. in some simulation studies involving many repeated calculations – is always to find a time- and memory-efficient numeric method of calculating large and relatively dense matrix – a method in which the computer memory is storing only the necessary results and is doing this in the most economic way.

The number of programming tools (packages) exist which address the problem of complex pedigree analysis – for example, PEDIG [Boichard 2002], PyPedal [Cole and Franke 2002],

Corresponding author – Adres do korespondencji: D.Sc Eng. Maciej Gierdziewicz, Department of Genetics and Animal Breeding, Agricultural University of Krakow, Mickiewicza 24/28, 30-059 Kraków, Poland, e-mail: rzgierdz@cyf-kr.edu.pl

ENDOG [Gutiérrez and Goyache 2005], or CFC [Sargolzaei et al. 2006], focusing – to a different extent – on speed, time, memory and disk space requirements, flexibility, and user-friendliness. Since our main problem was to speed up the calculations which took weeks to execute even with restricted pedigree length, the program described in this paper was intended mainly to save time and, in addition, to remove the pedigree length limit.

If speed is essential, parallel computing is often proposed as an alternative to traditional sequential calculations. This is the case eg. in genome analysis [Janaki and Joshi 2003], in biomolecular simulation [Germain et al. 2005] or in analyzing the origin of species [Bader et al. 2001]. Solving large sets of equations in animal breeding value estimation may also be the reason for the researchers to seek parallel aternatives [Lidauer et al. 1998, Lidauer and Strandén 1999, Lidauer et al. 1999, Strandén 1999, Strandén 2000, Strandén and Lidauer 2001]. In general, it is good if the programs which should run very fast are easily parallelizeable.

The aim of the work was to design a specialized computer program which would be capable of calculating relationship coefficients of a population or a subpopulation in the shortest possible time, applying directly the formula used in tabular method without, though, creating any – permanent or temporary – relationship tables – a program which should be easily transformed to parallel version.

MATERIAL AND METHODS

Data were two-generation pedigrees (bull-parents-grandparents) of 25 036 Polish Black and White bulls. The number of all animal ID's found in the pedigrees was 63 264. From these data the input data set for the computer program has been created. That data set contained 63 264 one-generation pedigrees, i.e. pedigrees of the form „animal-sire-dam", sorted in „chronological" order – parents before their progeny. The number of all possible animal pairs was 63 264*(63 2641)/2 = 2 001 135 216.

From the calculations made before [Kania-Gierdziewicz 2003] it is known that the number of related animal pairs in the population was not less then 120 000 000, i.e. 6% of the above maximum. So, creating a data file containing those non-zero relationship coefficients (two animal ID's plus their relationship coefficient, which means 12 bytes per pair) would require at least about one and a half gigabyte disk space, assuming single precision. Such large data sets are possible to create, but they are not efficient to access without a very fast computer. For that reason, in the above study each pair of animals has been analyzed separately, as it was explained by Tier [1990], who applied that methodology to calculate relationship coefficient between the parents of the animal, and then used the computed value to obtain the inbreeding coefficient. The final result of the calculations done by Kania-Gierdziewicz [2003] with this program consisted of the following statistics: the total number of animals, the number of inbred animals, the average inbreeding coefficient, the total number of pairs, the number of related pairs, and the average relationship coefficient. An auxiliary data set with nonzero inbreeding coefficients was also created. Because assigning computer memory to temporary relationship matrices was time-comsuming, the restriction was imposed to construct pedigrees not longer than 6 generations back. The total time of calculations with Tier's algorithm, used in the above paper, was chosen as the reference value for this study because the program was executed in each computer.

Our modification of Tier's algorithm was creating a dynamic hierarchical (dendroidal) structure of recursive function calls, instead of a small temporary relationship matrix, for each pair. The recursive function uses directly the formula

$$a_{ij} = 0.5\ (a_{ip} + a_{iq})\ /\ c\ (F_p, F_q),$$

where: animals p and q are parents of animal j; a_{ij}, a_{ip} and a_{iq} are the additive relationship coefficients between animals: i and j, i and p, i and q, respectively; $c(F_p,F_q)=[(1+F_p)(1+F_q)]^{0.5}$; F_p and F_q are inbreeding coefficients of animals p and q. This formula is known as a basis of the tabular method; the relationship coefficient between animals i and j (i is older than j) is the average relationship of animal i with the parents of animal j, corrected for their inbreeding coefficients. With this version of the program the same statistics have been calculated. In addition, because of the shorter execution time, the number of generations in the animal pedigree was not restricted. Moreover, for the modified computer program more statistics calculated „by row" (i.e. taking all a_{ij} for i<j) are possible to create, for example: the total number of the related „ancestor-progeny" pairs, or the sums of the corresponding relationship coefficients (a_{1j}+a_{2j}+a_{3j}+...+$a_{(j-1)j}$). They may be then used to calculate average relationship coefficient of an animal with its ancestors. The older the animal, the smaller the number of ancestors. That is, if the convention is applied to label a row of the relationship matrix with the name (number) of the progeny, and a column – with the name (number) of the ancestor (see Fig. 1), the lower triangular part of the relationship matrix has been analyzed. The results were then stored both for each row of the matrix and for the matrix as a whole, and the latter statistics were output together with the data set with individual results for each animal.

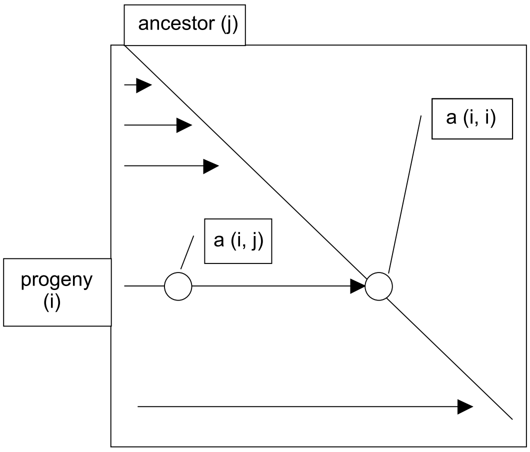

Fig. 1. Partitioning the lower triangular part of the relationship matrix A row by row
Rys. 1. Podział dolnej podmacierzy trójkątnej macierzy A na wiersze

To apply the recursion correctly, the pairs (i,p) and (i,q) have been sorted chronologically before making next recursive function calls. This arrangement was necessary because in the recursive formula the assumption is made that the relationship coefficients of the older animal with the parents of the younger animal of the pair are calculated.

The following computers were used to test the algorithms:

1) a relatively slow SGI Indy computer („indy");

2) SGI 2800 computer („grizzly");

3) SGI Altix 3700 („baribal");

4) SGI Altix 4700 („panda").

SGI Indy and SGI 2800 computers were used with Fortran 90 compilers [MIPSpro™ Fortran 90 2003], whereas both Altix computers had Intel Fortran compilers [Silverio et al. 2003]. The characteristics of the computers are given in Table 1.

Table 1. Characteristics of the computers
Tabela 1. Charakterystyka komputerów używanych do obliczeń

"Nickname" of computer Nazwa komputera	Computer type Typ komputera	Processor Procesor	No. of processors Liczba procesorów	Clock Zegar [GHz]	Memory Pamięć [GB]	Operating system System operacyjny	Fortran compiler Kompilator Fortranu
Indy	SGI Indy	R5000	1	0.15	0.25	Irix 6.5	f90
Grizzly	SGI 2800	R14000	64	0.5	48	Irix 6.5	f90
Baribal	SGI Altix 3700	Intel Itanium 2	128	1.5	256	SUSE Linux Enterprise Server 9	Intel Fortran
Panda	SGI Altix 4700	Intel Itanium 2	32	1.66	64	SUSE Linux Enterprise Server 9	Intel Fortran

As it may be concluded from above, the relationship coefficient of any pair of animals is calculated independently from the other pairs. So the only thing needed to parallelize calculations (to develop the multiprocessor version of the computer program) was to assign all the pairs to currently available processes. To test the efficiency of our algorithm the „Panda" computer was chosen, the number of processes was defined as either 8 or 16, and the set of all the animal pairs was divided into subsets („chunks"), equivalent to continuous ranges of matrix A rows, consisting of the following animal pairs:

Chunk 1: Pairs (i,j) for i = 2, 3, ..., 100, and j = 1, ..., i–1
Chunk 2: Pairs (i,j) for i = 101, 102, 103, ..., 200, and j = 1, 2, ..., i–1
...
Chunk 633: Pairs (i,j) for i = 63 201, 63 202, ..., 63 264, and j = 1, 2, ..., i–1

Since (1) „chunks" were not of equal size, and (2) may have been some external reasons to suspend the execution of any process, it was not possible to determine exactly the time needed to execute each „chunk". So, the dynamic way of assigning „chunks" to processes was chosen [Silverio et al., 2003]. When this option is used, each of the processes finishes calculation for its „chunk“, looks for the next available „chunk“, and begins to execute it.

The total and the CPU execution time was measured with Irix system commands or with Speedshop Analysis Package [SpeedShop User's Guide, 2003].

The programs were run at the Academy of Agriculture in Krakow and at The Academic Computer Centre „CYFRONET AGH“ in Krakow, as a part of the grant No. MNiI/SGI2800/AR/067/2004 („Parallelization of programs used to estimate breeding values“).

RESULTS

Time needed for recalculating relationship coefficients in all available computers with the program, which used Tier's algorithm and created small temporary relationship matrices in computer memory, the same as it has been done before by Kania-Gierdziewicz [2003], is presented in Table 2.

Table 2. Execution times of both programs in different computers
Tabela 2. Czasy wykonania obu programów na różnych komputerach

„Nickname“ of computer Nazwa komputera	Tier's algorithm Algorytm Tiera		Recursive algorithm Algorytm rekursywny		Acceleration ratio Wskaźnik przyspieszenia
	total time łączny czas [min]	CPU time czas CPU [min]	total time łączny czas [min]	CPU time czas CPU [min]	
Indy	41 000	40 000	3840	3800	10.5 – 10.7
Grizzly	5400	5390	224	223.1	24.1 – 24.2
Baribal	1485	1482	82.8	82.7	17.92 – 17.93
Panda	1354	1219	74	74	16.5 – 18.3

Approximately 10- to 24-fold decrease in computing time could be noted.

The next stage of speeding up execution of the algorithm was to parallelize it. The example results obtained with „Panda“ computer are presented in Table 3.

Table 3. Further reduction of total execution time after paralellization – „chunk” size for one process was 100 rows of the relationship matrix
Tabela 3. Dalsze skrócenie całkowitego czasu trwania obliczeń dzięki parallelizacji – jednorazowa porcja obliczeń dla jednego procesu wynosiła 100 wierszy macierzy spokrewnień

„Nickname“ of computer Nazwa komputera	Recursive algorithm – 1 processor Algorytm rekursywny – 1 procesor	Recursive algorithm – 16 processors Algorytm rekursywny – 16 procesorów	Acceleration ratio Wskaźnik przyspieszenia	Recursive algorithm – 32 processors Algorytm rekursywny – 32 procesory	Acceleration ratio Wskaźnik przyspieszenia
	„Wall clock” time czas rzeczywisty [min]	„Wall clock” time czas rzeczywisty [min]		„Wall clock” time czas rzeczywisty [min]	
Panda	74	5.08	14.6	2.83	26.1

Again the decrease in computing time was noticed, this time by the factor depending on the number of processors. The value of the acceleration ratio was smaller than the number of processors, but the differences were rather small (over 14-fold acceleration for 16 processors and over 26-fold for 32 processors). This time the acceleration factor exceeded 80% of the number of processors which seems to be quite a good result compared to e.g. the results of Strandén [1999].

Applying the recursive algorithm not only let shorten the execution time, but made it possible to analyze full-length pedigrees and, consequently, to use all the available pedigree information. In future calculations the values of statistics should differ from the results obtained before from the pedigrees of restricted length [Kania-Gierdziewicz 2003], because there should be more pairs of animals related only to a very small extent.

CONCLUSIONS

The major advantage of the presented method of calculating relationship coefficients over the Tier's algorithm was using explicit recursion built in programming language instead of the „quasi-tabular“ method. This resulted in about 20-fold decrease in the computing time.

Applying the recursion let parallelize the algorithm, which in turn led to further reduction in computing time by a factor of 80% of the number of used processors.

REFERENCES

Bader D.A., Moret B.M.E., Vawter L., 2001. Industrial Applications of High Performance Computing for Phylogeny Reconstruction. Proc. of SPIE ITCom: Commercial Applications for High-Performance-Computing, 19–24 August 2001, Denver, Colorado.

Boichard D., 2002. PEDIG: a FORTRAN package for pedigree analysis suited for large populations. Comm. 28–13 in Proc. 7th World Congr. Genet. Appl. Livest. Prod., Montpellier.

Cockerham C.C., 1954. An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. Genetics 41, 138–141.

Cole J.B., Franke D.E., 2002. Pedigree analysis using the Python programming language. J. Anim. Sci. 80 (Suppl. 1), 323.

Germain R.S., Fitch B., Rayshubskiy A., Eleftheriou M., Pitman M.C., Suits F., Giampapa M., Ward T.J.C., 2005. Blue matter on blue gene/L: massively parallel computation for biomolecular simulation. Proc. of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, September 19–21.IX.2005, Jersey City, NJ, 207–212.

Gutiérrez J.P., Goyache F. 2005. A note on ENDOG: a computer program for analysing pedigree information. J. Anim. Breed. Genet. 122, 172–176.

Henderson C.R., 1975. Rapid method for computing the inverse of a relationship matrix. J. Dairy Sci. 58, 1727–1730.

Henderson C.R., 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32, 69–83.

Janaki C., Joshi R.R., 2003. Accelerating comparative genomics using parallel computing. Silico Biol. 3 (4), 429–440.

Kania-Gierdziewicz J., 2003. Struktura genetyczna krajowej populacji buhajów czarno-białych [Genetic Structure of the Polish population of the Black and White sires]. Ph.D. thesis AR, Kraków [in Polish].

Lidauer M., Mäntysaari E.A., Strandén I., Kettunen A., Pösö J., 1998. DMUIOD: A multitrait BLUP program suitable for random regression testday models. Proc. 6th World Congr. Genet. Appl. Livest. Prod., 12–16 January 1998, Armidale, NSW, XXVII, 463–464.

Lidauer M., Strandén I., 1999. Fast and flexible program for genetic evaluation in dairy cattle. Proc. Computational Cattle Breeding '99, 18–20 March 1999, Tuusula. Interbull Bull. No. 20, International Bull Evaluation Service, Uppsala.

Lidauer M., Strandén I., Mäntysaari E.A., Pösö J., Kettunen A., 1999. Solving Large Test-Day Models by Iteration on Data and Preconditioned Conjugate Gradient. J. Dairy Sci. 82, 2788–2796.

MIPSpro™ Fortran 90, 2003. Commands and Directives Reference Manual. Silicon Graphics Inc.

Sargolzaei M., Iwaisaki H., Colleau J.J., 2006. CFC: a tool for monitoring genetic diversity. Comm. 27–28 in Proc. 8th World Congr. Genet. Appl. Livest. Prod., Belo Horizonte.

Silverio C.J., Graves D., Hogue C., 2003. Fortran 77 Programmer's Guide. Silicon Graphics Inc.

Silicon Graphics Inc., 2003. SpeedShop User's Guide.

Strandén I., 1999. Parallel benefits in test-day evaluations. Proc. Int. Workshop on Computational Cattle Breeding '99, 18–20 March 1999, Tuusula. Interbull Bull. 20, 26–32.

Strandén I., Lidauer M., 2001. Parallel Computing Applied to Breeding Value Estimation in Dairy Cattle. J. Dairy Sci. 84, 276–285.

Tier B., 1990. Computing inbreeding coefficients quickly. Genet. Sel. Evol. 22, 419–430.

BADANIE EFEKTYWNOŚCI ALGORYTMU REKURSYWNEGO DO OBLICZANIA WSPÓŁCZYNNIKÓW SPOKREWNIENIA

Streszczenie. Celem pracy było zaprojektowanie programu komputerowego, który wyliczałby współczynniki spokrewnienia w populacji lub subpopulacji w najkrótszym możliwym czasie, wykorzystując bezpośrednio wzór używany w metodzie tabelarycznej, jednak bez tworzenia jakichkolwiek macierzy spokrewnień. Dane stanowiły rodowody 25 036 polskich buhajów czarno-białych urodzonych w latach 1960–2000. W rodowodach występowały łącznie 63 264 zwierzęta. Z tego zbioru danych utworzono zbiór wejściowy do programu, zawierający 63 264 rodowody postaci „zwierzę–ojciec–matka", uporządkowane chronologicznie. Liczba wszystkich możliwych par zwierząt wynosiła 2 001 135 216. Algorytm przetestowano na czterech różnych komputerach, z których trzy były maszynami wieloprocesorowymi. W metodzie użytej do obliczania współczynników inbredu i spokrewnienia wykorzystano rekursywne wywołania funkcji spokrewnienia. Metoda rekursywna nie tylko pozwoliła skrócić czas trwania obliczeń od 10,7 do 18 razy, ale też przygotować programy do paralelizacji oraz umożliwi późniejsze oszacowanie większej liczby różnorodnych statystyk.

Słowa kluczowe: algorytm rekursywny, przetwarzanie równoległe, współczynnik spokrewnienia

Accepted for print – Zaakceptowano do druku: 17.09.2007