

Acceleration of GMDH combinatorial search with HPC clusters

Oleksiy A. Koshulko¹, Anatoliy I. Koshulko¹

¹*Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv, Glushkov str. 40.*

koshulko@opengmdh.org

Abstract. *Compute intensity of combinatorial algorithms of the Group Method of Data Handling (GMDH) requires to use multiprocessor computing environments in order to reduce processing time. The properties of combinatorial GMDH let us to use the concept of processing acceleration and expand capabilities of personal computer GMDH program with power of compute clusters.*

In order to evaluate speed optimization and effectiveness of the GMDH program that call compute cluster during its work we proposed a method of measuring model processing rate of combinatorial algorithms and a method of a priori processing time estimation.

Keywords

Combinatorial GMDH, parallel processing, processing speed.

1 Computing acceleration

Compute intensity of combinatorial algorithms of the Group Method of Data Handling (GMDH) [1] cause obstacles to its effective use. Posing a computational experiment requires strong limitation of complexity of mathematical models in order to finish computations in acceptable time. To expand capabilities of combinatorial GMDH it is necessary to use parallel processing in multi-CPU environments that lets the algorithm to take into consideration greater number of partial models. We consider the use of remote multi-CPU environments for acceleration of GMDH combinatorial search is an effective solution to problem of limited computational capabilities of personal computers.

In the high performance computing (HPC) acceleration of computing usually means the use of separate compute devices with own memory that like external coprocessor are connected to central processor. When it is necessary an accelerator can be used to process some of the compute intensive parts of a program and bring the results back to central processor memory. For example, accelerators can be commodity graphic processor units (GPU), field-programmable gate arrays (FPGA) or custom processors.

We applied the programming concept developed for accelerators to an HPC cluster as a single computing device accessible to personal computer through the Internet. Access to the compute cluster is performed using secure shell (SSH) protocol. In our case acceleration means that a program with graphic user interface uses resources of a personal computer as long as estimated time of completion of a compute problem is acceptable and if necessary the program call the HPC cluster through the Internet. The remote HPC cluster performs combinatorial search accelerated by parallel processing and returns a set of best models to personal computer where model parameters should be restored and then the program continue its local execution such as post-processing and visualization. This HPC capable software called Parallel COMBI [2] consists of several components:

- graphic user interface (GUI);
- combinatorial GMDH implementation for personal computers;
- combinatorial GMDH implementation for HPC clusters.

Also Parallel COMBI requires an SSH client to be available at the local host.

2 Estimation of processing time

A priori estimation of processing time for a GMDH problem is used for decision making about use of acceleration and about time delay in communication with accelerator before transfer of computing results. Estimation of processing time is a complex problem that has a technical and a mathematical component. The technical one appears because an HPC cluster can have many users and as a consequence we have a random number of free processors during each call. The mathematical component of the problem is the average processing time of combinatorial search iteration that heavily depend on such parameters as maximal complexity of the base function, type of external criterion, size of testing sample and size of best models set. Each combination of these parameters gives different average speed of partial model processing.

A number of computational experiments (Fig. 1-3) with criterion of regularity have been done. Measured average processing speed has been obtained at the cluster of Intel Xeon 2.3GHz processors. From 8 to 32 processors were used depending on computational complexity of an experiment. The average single-processor processing speed showed in figures is obtained as the number of considered partial models divided by processing time and by the number of used processors.

Fig. 1 shows decrease of processing speed caused by growth of base function complexity. The testing sample and the number of the best models are set to one in order to minimize their impact. Here and below the learning sample consists of 50 points. The trend curve is specific to the method of solving of linear systems in the least squares method. The trend has better characteristics when linear systems solved by the square root method than by the Gauss' method. A tilt and an altitude of the trend characterize quality of speed optimization and can be used for testing of any GMDH implementation with full combinatorial search.

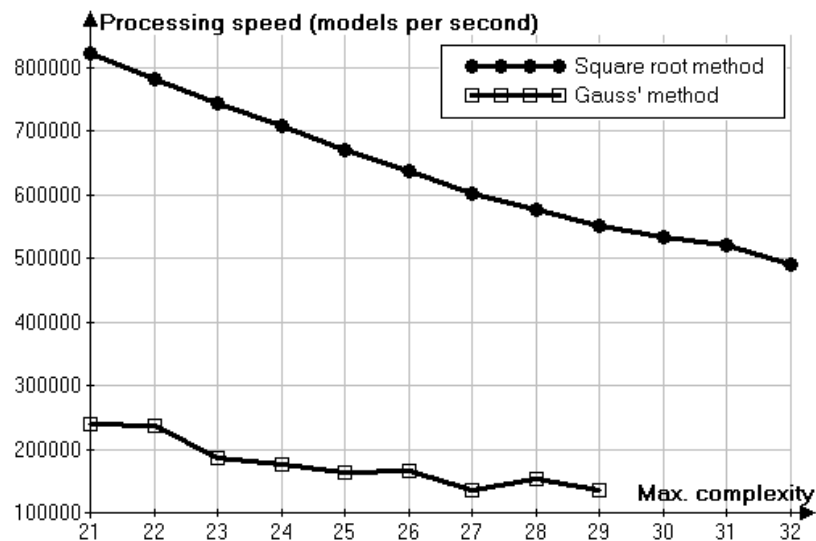


Fig. 1. Single-processor average processing speed achieved with different base function complexity and minimized impact of other parameters.

Fig. 2 shows that calculation of criterion value is also a compute intensive part of combinatorial GMDH algorithm in case of criterion of regularity. More complex criteria, for example criterion of unbiasedness, are expected to cause stronger decrease of processing speed.

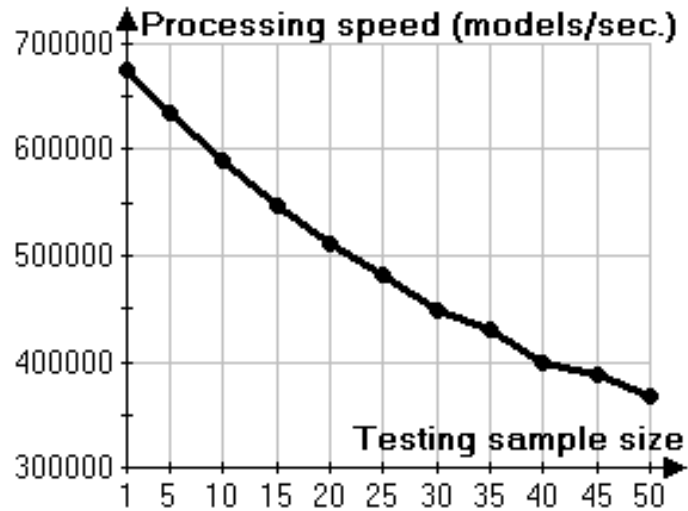


Fig. 2. Single-processor average processing speed achieved with base function complexity 25 and different testing samples.

Fig. 3 shows that size of the best models set is not very important comparatively to other parameters. Even strong increase of size of best models set can't significantly decrease the average processing speed. The numbers in Fig.3 were obtained for base function complexity 25 and single point testing sample.

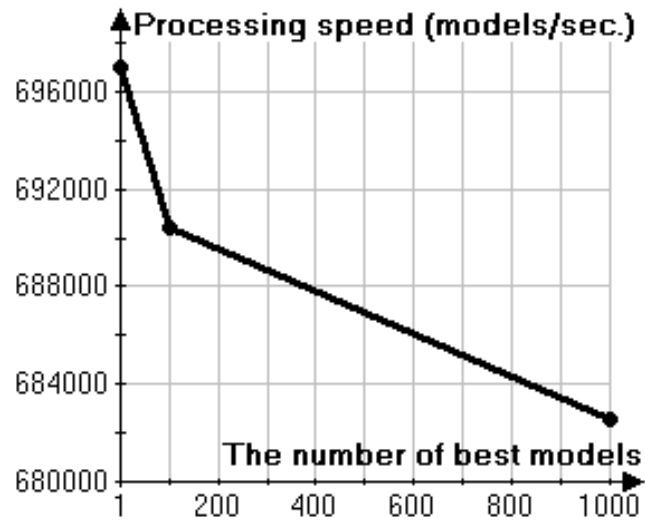


Fig. 3. Single-processor average processing speed achieved with base function complexity 25, single point testing sample and different sizes of best model sets.

Our approximate estimation of processing time is based on trends of processing speed built for different complexity $PS(c,1)$, $21 < c < 32$ (Fig. 1) and testing sample $PS(25,s)$, $1 < s < 50$ (Fig. 2).

We use polynomial models:

$$PS(c,1) = a_0 + a_1c + a_2c^2, \quad PS(25,s) = b_0 + b_1s + b_2s^2,$$

where a, b – coefficients estimated with least square method.

Then processing time $T(c,s)$ specific to our GMDH implementation is:

$$T(c,s) = \frac{R_{CPU}}{N_{CPU}} \left(T(c,1) + 2^c \left(\frac{T(25,s) - T(25,1)}{2^{25}} \right) \right)$$

or

$$T(c,s) = \frac{R_{CPU} 2^c}{N_{CPU}} \left(\frac{1}{PS(c,1)} + \frac{1}{PS(25,s)} + \frac{1}{PS(25,1)} \right),$$

where R_{CPU} is the relative processor speed (equals one for the processor used in our experiments) and N_{CPU} is the number of used processors. Testing of the expression proposed for $T(c,s)$ shows that its accuracy is quite acceptable (Tab. 1).

Tab.1. Testing of method of $T(c,s)$ estimation, $N_{CPU}=1$.

$T(c,s)$	Estimated time	Actual time
$T(22,45)$	10,05	10,31
$T(27,27)$	309,37	319,56
$T(30,12)$	2301,02	2333,69

3 Conclusion

Application of programming concept of acceleration to combinatorial GMDH software makes HPC cluster automatically accessible from a personal computer when estimated processing time exceeds user defined limit. Processing time depends most of all on such parameters of GMDH as base function complexity and testing sample size. Since the parameters show clear trends that can be represented by polynomial models of processing speed it is possible to predict processing time for any parameter combination inside investigated bounds of base function complexity and testing sample size. Determining the trend for processing speed achieved with minimal size of testing sample is also a good method to compare speed optimization of different combinatorial GMDH implementations.

References

- [1] Madala H.R., Ivakhnenko A.G.: Inductive Learning Algorithms for Complex Systems Modeling. – CRC Press, 1994. – 368 p.
- [2] Koshulko O.A., Koshulko A.I.: Adaptive parallel implementation of the Combinatorial GMDH algorithm. - Proceedings of International Workshop on Inductive Modelling 2007, September 22-26, Czech Technical University in Prague, ISBN-978-80-01-03881-9. pp. 71-77.