

# Optimal paralleling for solving combinatorial modelling problems

V.S. Stepashko, S.M. Yefimenko

*International Research and Training Centre of Information Technologies and Systems of the National Academy of Sciences and Ministry of Education and Sciences of Ukraine*

astrid@irtc.org.ua, syefim@ukr.net

**Abstract.** *Applying the parallel computing is one of ways for enhancement the modeling possibilities. The goals of the paper are to show the parallel computing effectiveness and possibility of providing the uniform load of all processors of the cluster. The scheme of algorithm with successive complication of structures is proposed. Test experiments of solving the problem of structural and parametrical identification on the cluster system scit-3 showed that the use of scheme of algorithm with successive complication of structures provides the uniform load of all processors of the cluster.*

## Keywords

GMDH combinatorial algorithm, parallel computing,  
cluster system, successive complication

## 1 Introduction

The exhaustive search of all possible variants of problem solving and choice of the best model forms the basis of GMDH combinatorial algorithm [1]. Complete enumeration of possibilities takes too much time and becomes impossible when the number of arguments is more than 30.

That is why the use of combinatorial algorithm on monoprocessor computers often does not allow solving problems with the large number of arguments. Applying the parallel computing is one of ways for enhancement the modeling possibilities [2].

The substantial problem here is non-uniformity of processors load of the cluster system. Different variants of organization of binary structural vector forming are used in combinatorial algorithm. Elements of this vector determine appropriate arguments that have to be included in the model.

## 2 GMDH combinatorial algorithm

We will consider the variants of organization of computations in the combinatorial generator, namely change of state methods of binary structural vector the elements of which specify including in the model of regressors with the proper numbers.

### a. Scheme of algorithm with the use of binary numbers generator

Such scheme uses the generation of binary numbers which correspond to sequential decimal numbers. Complication of partial models changes from 1 to the maximal number of  $m$ .

The scheme is simple enough and effective in case of uniprocessor system. However it is not quite suitable at paralleling of combinatorial algorithm on multiprocessor systems, because it does not provide the even loading on every processor of the cluster. Obviously the least computational loading (larger common amount of arguments at the identical amount of models) will fall on processors with a less sequence number in the cluster system, and most loading

– on processors with most sequence number, where models will be with the amount of arguments, near to the number of  $m$ . It is related to the feature of structural vector forming on principle of binary counter.

### b. Scheme of algorithm with successive complication of structures

This scheme uses such sequence of binary numbers generation when at first all connections appear with one unit in a structural vector ( $C_m^1 = m$  possible variants), then – with two units ( $C_m^2 = \frac{m(m-1)}{2}$  possible variants), and etc to one possible variant  $C_m^m = 1$  of including in the model of all arguments.

Such scheme can be easily enough applied for parallelization of combinatorial algorithm. Idea of the equal apportionment of common amount of models and arguments on all processors of the cluster system consists in the following. Amount of models  $C_m^i = \frac{m!}{(m-i)! \times i!}$  complexity of  $i$ ,  $i = \overline{1, m}$ , is evenly distributed between all processors of  $p$ , i.e. every processor “handles”  $\frac{m!}{p \times (m-i)! \times i!}$  models. Thus, it is necessary to define an initial point (first structural vector) for every processor.

## 3 Algorithm of determination of the initial state of binary structural vector by position at successive complication

Lets we have  $m$  arguments and  $k$  processors of the cluster system. We will write down the sequence of operations for the models of complication  $i$ ,  $i = \overline{1, m}$ :

1. Calculation of amount of combinations –  $C_m^i - 1$ .
2. Determination of the initial state of binary vector  $d$  for every processor  $j$ ,  $j = \overline{1, k}$  as a decimal number –  $\left[ \frac{C_m^i - 1}{k} \right] (j - 1) + 1$ .
3. Conversion from the decimal number to appropriate binary number for every processor:
 
$$\text{position} = \left[ \frac{C_m^i - 1}{k} \right] (j - 1) + 1;$$

$$u = i - 1, d = m - 1, C = C_d^u;$$
 Cycle on  $l$ ,  $l = \overline{1, m}$ 

$$\text{if position} \leq C \text{ then } b[l] = 1, u = u - 1, d = d - 1, C = C_d^u;$$

$$\text{else } b[l] = 0, \text{ position} = \text{position} - C, u = u - 1, C = C_d^u.$$

## 4 Application of scheme of paralleling with successive complication for solving high dimensional problems

The scheme of combinatorial algorithm with successive complication allows to apply the idea of exhaustive search for solving high dimensional problems (when the number of arguments  $> 50$ , and even the use of clusters with the large amount of processors does not allow to get a result).

We will consider, for example, problem of exhaustive search with the number of arguments  $m = 100$ . Lets consider that acceptable time of problem solving equals 1 minute – approximately so long it takes on one processor with speed of operation 2 GFLOPS (i.e. the processor of Intel Pentium 4 with a clock rate of 3 GHz) to execute an exhaustive search of 20 arguments (1048575 models). Let's suppose we have the cluster system with 64 processors. Then we can calculate how many processors have to be used to execute the exhaustive search of all models of complexity

$i, i = \overline{1, 100}$  in an acceptable time. If necessary amount of processors is less than available we will have to model and will not otherwise.

Table 1 presents the results of theoretical experiment with initial conditions mentioned above. As table shows, we can execute the exhaustive search of all models with complication less than 5 in acceptable time.

**Tab.1** Results of theoretical experiment

Amount of arguments	Amount of models	Amount of processors
1	100	0.0001
2	4950	0.00495
3	161700	0.1617
4	3921225	3.921225
5	75287520	75.28752
...	...	...
99	100	0.0001
100	1	0.000001

## 5 The results of experiments

With the purpose of efficiency estimation of the developed schemes the test experiments were carried out with the use of cluster system scit-3 [3] (the array of incorporated by communication network computational nodes each containing 2 dual core processor of Intel Xeon 5160 with a clock rate 3 GHz).

Run-time of combinatorial algorithm was measured at varying of processors amount from 1 to 16 and arguments amount from 20 to 24 (due to acceptable time of modeling).

Table 2 represents run-time of program with the use of proposed schemes (letter “b” corresponds to scheme with binary generator, and letter “s” corresponds to scheme with successive complication). Table 3 shows efficiency of schemes in case of  $k$  processors calculated as follows:

$$E_k = \frac{T_1}{k \times T_k} \times 100\%, \quad k = 2^i, \quad i = \overline{0, 4}, \quad (1)$$

where  $T_k$  is run-time of program on  $k$  processors.

The efficiency of scheme with the use of binary numbers generator decreases with increasing number of processors due to non-uniform loading of the processors.

As table 3 shows the efficiency of scheme with successive complication remain at level 98-100% when the number of processors equals 16. It is an evidence of providing of the even loading on all processors of the cluster system.

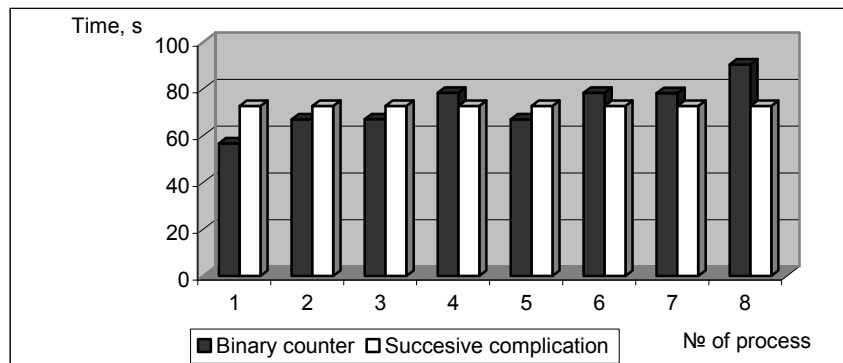
We also carried out comparative experiment with separation of all steps of combinatorial algorithms on 8 processes and successive implementation on one processor. The experiment can be accepted as pure (the result is close to theoretical), because of exception of interprocessor communication. The represented in the form of time plot result on figure 1 evidently demonstrates the high efficiency of the algorithm with successive complication of structures. Table 4 shows numerical value of efficiency of paralleling for this experiment.

**Tab. 2.** Run-time of combinatorial algorithm

	Number of arguments									
	20		21		22		23		24	
Number of processors	Time, seconds									
	b	s	b	s	b	s	b	s	b	s
1	70	70	152	152	329	331	715	711	1541	1524
2	38	35	82	76	177	165	385	354	824	766
4	20	17	44	38	95	83	205	181	441	386
8	11	9	24	19	51	41	110	91	235	192
16	6	4	13	10	29	21	58	45	125	96

**Tab. 3.** – Efficiency of schemes of paralleling of combinatorial algorithm

	Number of arguments									
	20		21		22		23		24	
Number of processors	Efficiency, %									
	b	s	b	s	b	s	b	s	b	s
1	100	100	100	100	100	100	100	100	100	100
2	92	100	93	99	93	100	93	100	93	99
4	86	100	66	100	77	100	77	98	77	99
8	79	100	60	99	80	100	81	98	82	99
16	74	99	55	99	70	99	77	100	77	100



**Fig.1.** Time plot for 8 processes on one processor

**Tab. 4.** Efficiency of paralleling

	binary numbers generator	successive complication
Efficiency	80,7%	99,8%

## 6 Conclusion

The scheme of operations paralleling in a combinatorial algorithm on principle of binary counter is explored. It is shown that it does not provide the uniform loading on all processors of the cluster system. With the increase of the number of processors of cluster system efficiency of paralleling decreases considerably.

The new method of paralleling is developed on the basis of algorithm of generation of the successively complicated structures of models. By the tests experiments it is shown on the cluster system scit-3 that the use of the offered scheme provides the equal total amount of models and estimated parameters on every processor. Efficiency of scheme applying when using 32 processors is 98-100% while efficiency of scheme on principle of binary counter is less than 70%.

## References

- [1] Ivakhnenko A.G., Stepashko V.S.: Noise stability of modeling, Naukova Dumka, Kyiv, 1985, 216 p. (In Russian)
- [2] Stepashko V.S., Yefimenko S.M., Rozenblat O.P., Chernyack A.I. On application of parallel computations in tasks of modelling on the basis of inductive approach // Problems in programming. – 2006. – № 2-3. – PP. 170-176.
- [3] <http://cluster.icyb.kiev.ua>.