

# Evolutionary Combination of Models in DSS based on Genetic Programming

Chaofan Dai

C4ISR Technology Key Laboratory of Science and Technology for National Defense, National University of Defense  
Technology, Changsha, China  
cfdai@263.net

Yanghe Feng, Jianmai Shi

C4ISR Technology Key Laboratory of Science and Technology for National Defense, National University of Defense  
Technology, Changsha, China  
fengyanghe@gmail.com

**Abstract**—The efficiency of model-aided decision making relies on the intelligent level of model selection. The purpose of this paper is to develop a new algorithm for model selection based on genetic programming. In the algorithm, the meta-models are classified according to the characteristics of the sample data, and the combined models are built as tree format. The genetic operations are performed under some constraints to produce combination models for users' reference. The process of the algorithm greatly decreases users' dependence on domain knowledge.

**Index Terms**—DSS; Model Base; Automatic Model Selection; Genetic Programming

## I. INTRODUCTION

Recently model-aided decision techniques have received increasing attention and a large number of models have been developed. It becomes very important, therefore, to efficiently utilize and share the model resources, especially by using more intelligent tools to select and combine current models.

A large number of model selection methods have been studied in the literature. Especially, there came an analytical model selection method, through which models can be selected according to the historical record of their applications and the characteristics of the decision problem<sup>[1]</sup>. The structure of the model can be represented by knowledge framework, then the reasoning trees are established according to the fact base and the knowledge base. Thus a knowledge-based automatic model selection method is proposed by integrating the experience and expert knowledge<sup>[2-4]</sup>. Gradient-based approaches are developed for model selection and present impressive gain in time complexity<sup>[5-6]</sup>, but they may fall into bad local minima.

Song and Lee<sup>[7]</sup> propose a Bayesian model selection method by applying the path sampling for computing Bayes factors. Gutiérrez-Peña et al<sup>[8]</sup> present objective Bayes procedures for model selection. Duan et al<sup>[9]</sup> develop an automatic model selection method through combining clustering and Bayesian information criterion, which is applied to forecast stock price. Curry and

Morgan<sup>[10]</sup> investigate model selection problems in Neural Networks. Egrioglu et al<sup>[11]</sup> propose a more complicated model selection strategy based on artificial neural networks. Alan Brookharta and Van Der Laanb<sup>[12]</sup> present a finite sample criterion based on cross validation, which can be applied to select a nuisance parameter model from a set of candidate models. Based on Genetic Algorithm, Lee and Doong<sup>[13]</sup> develop a model selection method for time constrained decision problems. Celik and Deha Er<sup>[14]</sup> present a FAD-based model selection interface for model selection. Hutter and Tran<sup>[15]</sup> propose a novel principle - the Loss Rank Principle - for model selection in regression and classification. There are also many other researches on model selection, such as robust model selection<sup>[16,17]</sup>, model selection with correlated variables<sup>[18]</sup>, and model selection in support vector machine<sup>[6, 19, 20]</sup> as well as in energy planning<sup>[21]</sup>.

Current methods can efficiently support users to automatically or intelligently select single model. For complex decision problems, however, it is usually necessary to combine multiple models to support users. Thus, the model joint selection and combination problem is studied in this paper, and a new model combination algorithm is developed based on the idea of genetic programming. The algorithm can create combination models through the sample data and greatly decrease the dependence of model selection on the domain knowledge.

## II. THE BASIC IDEA FOR PRODUCING COMBINATION MODELS

In this paper, the candidate models in the model database are named as meta-model. Users can select multiple models to build up the basic model set according to the requirement of decision problem. The combination models produced by the algorithm are all assembled by models in the basic model set. The algorithm is developed based on the following assumptions:

- The model database includes all the meta-models that are required to combine the decision model.
- The input and output parameters of the meta-model are numerical.

- There are sufficient inputs and outputs samples for the decision problem, and all these samples are effective.

A. *Related definitions and restriction*

When the combination model evolves through genetic programming, the basic model set is treated as the basic component. The model combination process is evolved through genetic operations under the control of adapting function where an ordered tree, ie. the combination model is obtained.

**Definition 1:** Given a model  $M$ , its input parameters are  $I_1, \dots, I_n$ , and the  $m$ th output parameter is denoted as  $M(I_1, \dots, I_n)[m]$ ,  $n \geq 1$ ,  $m > 1$ ; if model  $M$  has only one output parameter, it is denoted as  $M(I_1, \dots, I_n)$ .

From Definition 1, it can be known that the combination model can be represented as a multi-layer nesting of meta-models. When the model is combined into nesting structure, the parameters of the meta-models must match with each other. Thus, the rules for parameter matching must be set up firstly, so as to restrict the transforms between different types of data in the model combination process.

**Rule 1:** Parameters are allowed to transform from low precision to high precision.

The above transform rule is defaulted, and users can also force a transform from high precision to low precision. Therefore, in order to provide a transforming and matching tool for the parameters that can not be transformed implicitly, a set of models should be proposed to force this transform.

**Definition 2:**  $a$  is the output parameter of model  $M_1$ , and  $b$  is the input parameter of  $M_2$ . If  $a$  can be transformed to  $b$  implicitly by Rule 1, the types of parameters  $a$  and  $b$  are consistent, which is denoted as  $a \triangleright b$ .

In order to ensure the correctness of the combination model, the consistent restriction for the parameters must be satisfied. The impact of multi-output model on the genetic operations should also be considered. Therefore, the following restriction rules are developed.

**Restriction :** The necessary condition for models  $M_1(x)$  and  $M_2(y)$  are combined into  $M_1(M_2(y))$  or  $M_1(M_2(y)[n])$  is  $M_2(y) \triangleright x$ .

B. *Model preprocess*

The candidate meta-models are selected according to the amount and types of the samples, which can greatly reduce the size of the basic model set and increase the efficiency of the algorithm. This strategy can also ensure that all the required meta-models are included in the basic model set.

**Definition 3:** For two parameter sets  $X = \{x_1, x_2, \dots, x_n\}$  and  $Y = \{y_1, y_2, \dots, y_m\}$ , if  $\forall x \in X$  for  $1 \leq i \leq n$ ,  $\exists y \in Y$  for  $1 \leq j \leq m$ , satisfies  $x \triangleright y$ , then parameter set  $X$  is included in  $Y$ , which is denoted as  $X \subset Y$ . If  $X \subset Y$  and  $Y \subset X$ ,  $X$  is equivalent to  $Y$ , and note as  $X \Leftrightarrow Y$ .

In order to reduce the range of the initial population, according to the layers of the nodes in the tree, the models located at root node, middle node and leaf node are called as top level model, middle level model and bottom level model respectively. Their selecting rules are as follows:

**Rule 2:** Selecting Rule for top level models. Assume that the output parameter set provided by users is  $O = \{o_1, o_2, \dots, o_n\}$ . If the output parameter set of the model,  $Y = \{y_1, y_2, \dots, y_n\}$ , satisfies  $O \Leftrightarrow Y$ , the model can be used as the top level model, and the set formed by all top level models is noted as  $TMSet$ .

**Theorem 1:** If the output parameter set of the combination model is  $O = \{o_1, o_2, \dots, o_n\}$ , its top level model  $m$  satisfies that  $m \in TMSet$ .

**Proof:** Since the output parameter set of top level model  $m$  is  $O = \{o_1, o_2, \dots, o_n\}$ , from Rule 1, it can be known that  $m \in TMSet$ .

Rule 2 presents the condition for selecting top level models. Since the roles that top level models act are the final output of the combination model, their output parameters must be consistent with the combination model, that is, the number and types of the parameters must match with each other. Theorem 1 ensures the integrity of the top level model set.

**Rule 3:** Selecting Rule for the bottom level models. Assume that the output parameter set provided by users is  $I = \{i_1, i_2, \dots, i_n\}$ . If the input parameter set of the model,  $X = \{x_1, x_2, \dots, x_m\}$ , satisfies  $X \subset I$ , they can be used as the bottom level model of the combination model, and the set formed by all bottom level models is noted as  $BMSet$ .

**Theorem 2:** If the input parameter set of the combination model is  $I = \{i_1, i_2, \dots, i_n\}$ , its bottom level model set  $M$  satisfies that  $M \subset BMSet$ , and the parameter set of  $BMSet$ ,  $X = \{x_1, x_2, \dots, x_n\}$ , satisfies  $X \Leftrightarrow I$ .

**Proof:** For  $\forall m_i \in M$ , the input parameter set of bottom level model  $m_i$ , satisfies  $I_i \subset I$ . Therefore, according to Rule 2, it can be known that  $m_i \in BMSet$ . From Assumptions 1 and 3, we know that  $I \subset X$ , where  $X = \{x_1, x_2, \dots, x_n\}$  is the parameter set of  $BMSet$ . By Rule 2, it can be obtained that  $X \subset I$ . Thus,  $X \Leftrightarrow I$ .

Rule 3 presents the condition for selecting bottom level models. Since the input parameters of all bottom level models form the input parameters of the combination model, the input parameter set of every bottom level model must be included in the input parameter set of the combination model. Theorem 2 ensures the integrity of the bottom level model set.

The middle level model set,  $MMSet$ , is difficult to determine, and it is determined in the process of producing the combination model by the algorithm in the following Section according to the characteristics of the top level models and the bottom level models.

The model preprocess is to select meta-models according to the input and output parameters of the combination model, and the knowledge included by model is not sufficiently utilized. Thus, our algorithm allows users to select more precise top level and bottom level models according to the knowledge of the combination model and meta-models in the model database, so as to increase the correctness and efficiency of the algorithm.

### C. Algorithm for producing initial population

The initial population is created by the following algorithm according to the bottom level model set, the middle level model set and the top level model set.

The input parameter set for the samples that provided by users is noted as  $I = \{i_1, i_2, \dots, i_n\}$ , and the capacity of the initial population is  $c$ .

The algorithm for producing the initial population of the combination model is presented below:

Step 1: Randomly select the biggest depth of the combination model,  $n$ , according to the maximum levels set by users for the combination model.

Step 2: Randomly select the top level model  $m \in TMSet$ , and obtain its input parameter set  $I_n = \{i_1^n, i_2^n, \dots, i_n^n\}$ .

Step 3: If the depth of current combination model  $L \leq n-2$ , go to Step 4.1; otherwise, go to Step 4.2.

Step 4.1: For the input parameter set of Level  $L$ ,  $I_L = \{i_1^L, i_2^L, \dots, i_n^L\}$ , if the input parameter  $i_j^L \in I_L$ , and  $\exists i \in I$  satisfies  $i_j^L \triangleright i$ , then perform operation  $a$ ; otherwise, perform operation  $b$ . Go to Step 5.

a) Select model adding operation and input parameter determining operation at the predetermined probability  $\tau$ . The model adding operation uses other models' output as the input parameters of the node, while input parameter determining operation sets the parameters of node to be the input parameters of the combination model, that is, to match with the input parameter set of the related sample. If more models are required, randomly select model  $m' \in MMSet$ , where its output parameter set is  $O_{m'}$ , and  $\exists o \in O_{m'}$  satisfies  $o \triangleright i_j^L$ . If the input parameters are required to be determined, randomly select the input parameter  $i_j \in I$  at uniform probability, where  $i_j \triangleright i_j^L$ .

b) Randomly select model  $m' \in MMSet$ , whose output parameter set is  $O_{m'}$  and  $\exists o \in O_{m'}$  satisfies  $o \triangleright i_j^L$ .

Step 4.2: For the input parameter set of Level  $L$ ,  $I_L = \{i_1^L, i_2^L, \dots, i_n^L\}$ , if the input parameter  $i_j^L \in I_L$ , and  $\exists i \in I$  satisfies  $i_j^L \triangleright i$ , then perform operation  $a$ ; otherwise, perform operation  $b$ . Go to Step 5.

a) Select model adding operation and input parameter determining operation at the predetermined probability  $\tau$ . If more models are required, randomly select model  $m' \in BMSet$ , where its output parameter set is  $O_{m'}$ , and  $\exists o \in O_{m'}$  satisfies  $o \triangleright i_j^L$ . If the input parameters are required to be determined, randomly select the input parameter  $i_j \in I$  at uniform probability, where  $i_j \triangleright i_j^L$ .

b) Randomly select model  $m' \in BMSet$ , whose output parameter set is  $O_{m'}$  and  $\exists o \in O_{m'}$  satisfies  $o \triangleright i_j^L$ .

Step 5: If the depth of current combination model  $L > n$ , go to Step 1. For the situation  $L \leq n$ , if the input parameter set of current combination model  $I' \Leftrightarrow I$ , put the combination model into the initial population and go to Step 6; otherwise, go to Step 3.

Step 6: If the amount of current initial population  $c' = c$ , stop the algorithm; otherwise, go to Step 1.

### D. Adaptive fitness function design

Adaptive fitness function describes the gap between the combination model produced by the algorithm and the goal model, which is used to control the genetic operations. When the combination model is produced, it is tested by the samples provided by users and a series of test results are obtained, based on which the adaptive fitness function is set up.

The overall error of the combination model,  $\omega$ , represents the sample gap between the combination model and the goal model, and it can be obtained by statistically analyzing the results.

$$\omega = \sum_{i=1}^n |y_i - x_i|, \text{ where } n \text{ is the total number of the}$$

samples,  $y_i$  represents the results obtained by the combination model, and  $x_i$  is the output results of the sample.

The distribution shape similarity between the combination model and the goal model can be represented by the total deviation of the error.

$$s = \sum_{j=1}^n [|y_j - x_j| - \sum_{i=1}^n (y_i - x_i)/n]^2, \text{ where } n \text{ is the total}$$

number of the samples,  $y_i$  represents the results obtained by the combination model, and  $x_i$  is the output results of the sample.

Then the adaptive fitness function can be described as follows:

$$fitness = \alpha\omega + (1-\alpha)s,$$

where  $\alpha$  is the error weighting coefficient ( $0 \leq \alpha \leq 1$ ). The bigger  $\alpha$  is, the more important the error of distribution shape is.

The adaptive fitness function,  $fitness \geq 0$ , represents the gap between the combination model and the goal model, which is used to control the genetic operations. The smaller its value is, the closer the combination model is to the goal model.

### E. Genetic operation

Genetic operation is a detailed process to produce the combination model. There are three types of operations: selection, crossover and mutation. All of them are performed under the control of the adaptive fitness function, but their father generation producing methods are different.

Let  $I = \{i_1, i_2, \dots, i_n\}$  be the input parameter set of the produced combination model.  $BMSet$ ,  $MMSet$  and  $TMSet$  represent the bottom model set, the middle model set and the top model set respectively.  $mset$  denotes the father generation model set, while  $child(m)$  is child model under the root node  $m$ .

The father generation models are selected according to the following probability:

If  $\forall m_i \in mset$  and the value of the adaptive fitness function for  $m$  is  $fitness(m_i)$ , the selecting probability for model  $m$  is

$$P(m_i) = 1 - \left[ \frac{\text{fitness}(m_i)^\lambda}{\sum_{j=1}^n \text{fitness}(m_j)^\lambda} \right]$$

where  $\lambda$  is the adaptation effecting coefficient ( $\lambda > 0$ ), and  $n$  is the total number of elements in set  $mset$ .  $\lambda$  represents the importance of the adaptive fitness function. When  $\lambda \in (0,1)$ , the influence of the adaptive fitness function on the model selecting probability is decreasing. When  $\lambda > 1$ , the influence is increasing.

The father generation model selecting rules for the three genetic operations are as follows:

a) Selection rule. The selecting probability for father generation model  $m_i$  is  $P(m_i)$ .

b) Crossover rule. Let  $child(m_i)$  be the child model at the cross node of father generation model  $m_i$ , and its output parameter set is  $O(child(m_i))$ . Let  $child(m_j)$  be the child model at the cross node of father generation model  $m_j$ , and its output parameter set is  $O(child(m_j))$ . Then  $O(child(m_i)) \Leftrightarrow O(child(m_j))$ , and the input parameter sets for the two produced child generation models,  $I_1$  and  $I_2$ , must satisfy Rule 3, that is,  $I_1 \Leftrightarrow I$  and  $I_2 \Leftrightarrow I$ . The selecting probabilities for  $m_i$  and  $m_j$  are  $P(m_i)$  and  $P(m_j)$ , and the cross node is randomly produced at uniform probability.

c) Mutation rule. Let  $meta(m)$  denote the meta-model at the mutation node of father generation model  $m$ , and its input parameter set and output parameter set are  $I(meta(m))$  and  $O(meta(m))$  respectively,  $meta(m) \in BMSet$  ( $MMSet$  or  $TMSet$ ).  $I(meta(m)')$  and  $O(meta(m)')$  are the input parameter set and the output set for the meta-model  $meta(m)'$  that used to replace  $meta(m)$ . Then the followings must be satisfied:  $I(meta(m)) \Leftrightarrow I(meta(m)'), O(meta(m)) \Leftrightarrow O(meta(m)'),$  and  $meta(m)' \in BMSet$  ( $MMSet$  or  $TMSet$ ). The selecting probability for  $m$  is  $P(m)$ , and the mutation node and the replacing model are randomly produced at uniform probability.

#### F. The procedure of the algorithm

The detailed procedure of the model combination algorithm based on genetic programming is as follows:

Step 1: Determine the biggest number for the generations. Set the values for the error weighting coefficient, the adaptation effecting coefficient and the probabilities of the genetic operations. In the initial stage, more crossover operations are preformed, and the probability for mutation increases gradually in the following stages.

Step 2: Preprocess the models according to the combination model samples, set up the basic model set, and determine the bottom level model set, the middle model set and the top model set. Create the initial population of the combination model by producing algorithm in Section 2.3.

Step 3: Perform all the combination models in current population by the samples, and calculate their values for the adaptive fitness function.

Step 4: Select the genetic operations by their probabilities, and perform selection, crossover and mutation operations by the related rules. Create child generation combination models until the new population is as big as the old.

Step 5: Replace the old population by the new one and go to Step 3. When the required combination model is obtained or the biggest generation is reached, stop the algorithm.

### III. EXPERIMENT ANALYSIS

Thirty five representative numerical computation models are selected to form the basic model set, and one hundred samples are used to test the genetic programming algorithm. The combination model is obtained through the evolution procedure.

a) The thirty five scientific computation models are implemented by Matlab 7.5.0., which are shown in Table 1.

b) There are one hundred samples for the combination model, and each sample has three input parameters and one output parameter, which are shown in Table 2.

c) Error weighting coefficient  $\alpha=1$ , adaptive fitness impacting coefficient  $\gamma=1$ , the longest generation is 100 (Test 1), 10 (Test 2), 100 (Test 3), the probabilities for crossover, selection and mutation operations are 0.7, 0.2, and 0.1 respectively in all of tests. When the number of generations is over 50, the probabilities for crossover, selection and mutation operations are changed to 0.6, 0.15, and 0.25 respectively in all of tests. The population size is 50 (Test 1), 500 (Test 2), 1000 (Test 3), and the experiment is performed 5 times. The combination model with lowest adaptive fitness is selected as the experiment results.

d) In order to observe the best results that the algorithm can obtain in predetermined number of generations, the stopping criterions of the algorithm are that the adaptive fitness is equal to zero or the biggest number of generations is reached.

e) Record the adaptive fitness, the number of nodes and the number of generations for the combination models that are produced by every sample. Present the tree structure of the combination model through figure.

Respectively, figure 1, 4 and 7 show the number of levels and the numbers of nodes (including the number of model nodes and the number of parameter nodes) for the combination model with the best adaptive fitness in each generation of the population in Test 1, Test 2 and Test 3. Similarly, figure 2, 5 and 8 illustrate the best adaptive fitness in each generation of the population in Test 1, Test 2 and Test 3, and Figure 3, 6 and 9 presents the structure of the combination model in Test 1, Test 2 and Test 3.

The following conclusions can be obtained through the experiment:

1) Based on samples, the algorithm can present multi-level multi-node combination models through genetic operations while it is not necessary for the users to understand the models. Therefore, users without domain knowledge can also create models though their samples.

TABLE 1 MATHEMATICAL COMPUTATION MODEL LIST

Model name	The number of input parameters	Model name	The number of input parameters	Model name	The number of input parameters	Model name	The number of input parameters	Model name	The number of input parameters	Model name	The number of input parameters
divide	2	acos	1	sinh	1	acosd	1	log2	1	square	1
sqrt	1	atan	1	sin	1	acosh	1	min	2	log10	1
power	2	times	2	cos	1	ceil	1	expml	1	round	1
abs	1	plus	2	tan	1	conj	1	max	2	csc	1
log	1	minus	2	tanh	1	cot	1	hypot	2	loglp	1
ceil	1	mod	2	asin	1	cotd	1	floor	1		

TABLE 2 SAMPLE DATA SET

Samples	Input parameter 1	Input parameter 2	Input parameter 3	Output parameter	Sampl es	Input parameter 1	Input parameter 2	Input parameter 3	Output parameter
1	2.5119	1.1553	0.9072	5.3251	51	7.1539	6.3842	5.8314	35.7389
2	2.5650	1.1954	0.9416	5.3531	52	7.3053	6.6063	6.0525	38.5668
3	2.6193	1.2370	0.9773	5.3967	53	7.4598	6.8360	6.2820	40.3383
4	2.6747	1.2801	1.0143	5.4573	54	7.6176	7.0738	6.5202	40.6943
5	2.7313	1.3246	1.0528	5.5364	55	7.7787	7.3198	6.7674	39.6508
6	2.7890	1.3706	1.0927	5.6351	56	7.9433	7.5744	7.0240	37.6285
7	2.8480	1.4183	1.1341	5.7546	57	8.1113	7.8379	7.2903	35.2709
8	2.9083	1.4676	1.1771	5.8953	58	8.2829	8.1105	7.5667	33.1649
9	2.9698	1.5187	1.2218	6.0576	59	8.4581	8.3926	7.8536	31.6507
10	3.0326	1.5715	1.2681	6.2413	60	8.6370	8.6845	8.1514	30.8085
11	3.0968	1.6262	1.3162	6.4459	61	8.8197	8.9865	8.4604	30.5628
12	3.1623	1.6827	1.3661	6.6704	62	9.0063	9.2991	8.7812	30.7998
13	3.2292	1.7413	1.4179	6.9135	63	9.1968	9.6225	9.1141	31.4391
14	3.2975	1.8018	1.4716	7.1733	64	9.3913	9.9572	9.4597	32.4665
15	3.3672	1.8645	1.5274	7.4476	65	9.5900	10.3035	9.8184	33.9541
16	3.4385	1.9293	1.5853	7.7342	66	9.7928	10.6619	10.1906	36.0878
17	3.5112	1.9964	1.6454	8.0305	67	10.0000	11.0327	10.5770	39.1990
18	3.5855	2.0659	1.7078	8.3337	68	10.2115	11.4165	10.9780	43.7420
19	3.6613	2.1377	1.7726	8.6414	69	10.4275	11.8135	11.3942	50.0694
20	3.7388	2.2121	1.8398	8.9510	70	10.6481	12.2244	11.8263	57.8313
21	3.8178	2.2890	1.9095	9.2603	71	10.8734	12.6496	12.2746	65.1792
22	3.8986	2.3686	1.9819	9.5675	72	11.1034	13.0896	12.7400	68.8849
23	3.9811	2.4510	2.0571	9.8713	73	11.3382	13.5448	13.2231	66.7551
24	4.0653	2.5363	2.1351	10.1707	74	11.5781	14.0159	13.7244	60.4809
25	4.1513	2.6245	2.2160	10.4655	75	11.8230	14.5034	14.2448	54.2006
26	4.2391	2.7158	2.3000	10.7558	76	12.0731	15.0079	14.7848	50.4613
27	4.3288	2.8102	2.3872	11.0425	77	12.3285	15.5299	15.3454	49.3868
28	4.4203	2.9080	2.4778	11.3267	78	12.5893	16.0700	15.9272	50.4463
29	4.5138	3.0091	2.5717	11.6102	79	12.8556	16.6289	16.5311	53.9171
30	4.6093	3.1138	2.6692	11.8948	80	13.1275	17.2073	17.1579	61.9190
31	4.7068	3.2221	2.7704	12.1830	81	13.4052	17.8058	17.8084	78.1969
32	4.8064	3.3342	2.8754	12.4772	82	13.6887	18.4251	18.4836	99.2567
33	4.9081	3.4501	2.9845	12.7803	83	13.9783	19.0659	19.1844	104.1272
34	5.0119	3.5701	3.0976	13.0952	84	14.2740	19.7291	19.9118	87.1747
35	5.1179	3.6943	3.2151	13.4253	85	14.5759	20.4153	20.6667	71.5228
36	5.2261	3.8228	3.3370	13.7747	86	14.8843	21.1253	21.4503	66.1404
37	5.3367	3.9557	3.4635	14.1483	87	15.1991	21.8601	22.2636	67.5928
38	5.4496	4.0933	3.5948	14.5527	88	15.5206	22.6204	23.1077	76.4244
39	5.5649	4.2357	3.7311	14.9968	89	15.8489	23.4072	23.9838	100.8649
40	5.6826	4.3830	3.8726	15.4935	90	16.1842	24.2213	24.8931	132.1729
41	5.8028	4.5355	4.0194	16.0616	91	16.5265	25.0638	25.8369	121.3317
42	5.9255	4.6932	4.1718	16.7283	92	16.8761	25.9355	26.8165	92.7624
43	6.0509	4.8565	4.3300	17.5328	93	17.2331	26.8376	27.8333	83.2073
44	6.1789	5.0254	4.4941	18.5297	94	17.5976	27.7710	28.8886	87.4502
45	6.3096	5.2002	4.6645	19.7908	95	17.9699	28.7369	29.9839	111.5863
46	6.4430	5.3810	4.8414	21.4026	96	18.3500	29.7364	31.1207	158.0834
47	6.5793	5.5682	5.0249	23.4531	97	18.7382	30.7707	32.3006	141.3787
48	6.7185	5.7618	5.2154	26.0030	98	19.1345	31.8409	33.5253	104.9646
49	6.8606	5.9622	5.4132	29.0352	99	19.5393	32.9484	34.7964	100.7545
50	7.0057	6.1696	5.6184	32.3939	100	19.9526	34.0944	36.1157	125.2346

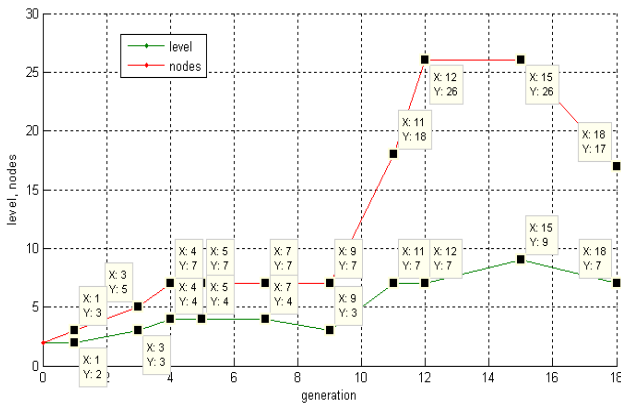


Figure 1 The best number of levels and nodes in each generation (Test 1).

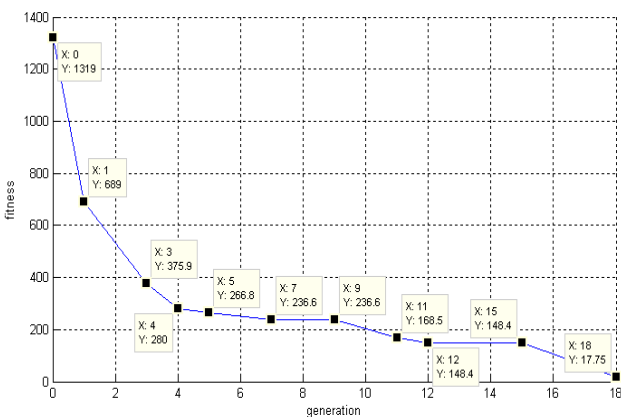


Figure 2 The best adaptive fitness in each generation (Test 1).

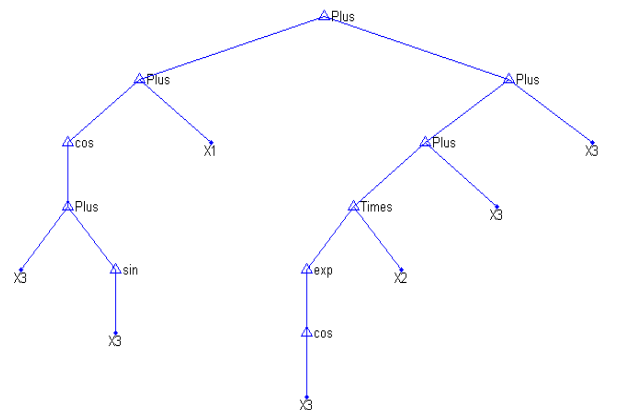


Figure 3 The structure of the combination model (Test 1).

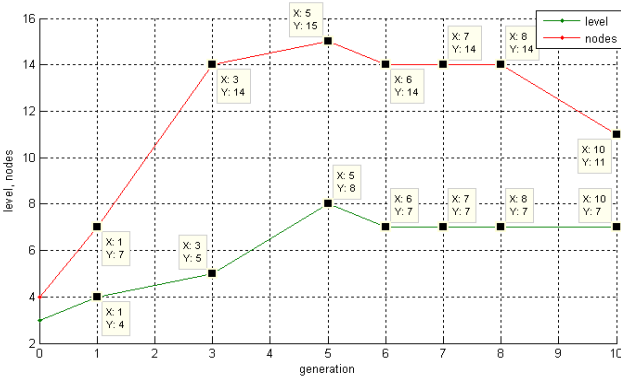


Figure 4 The best number of levels and nodes in each generation (Test 2)

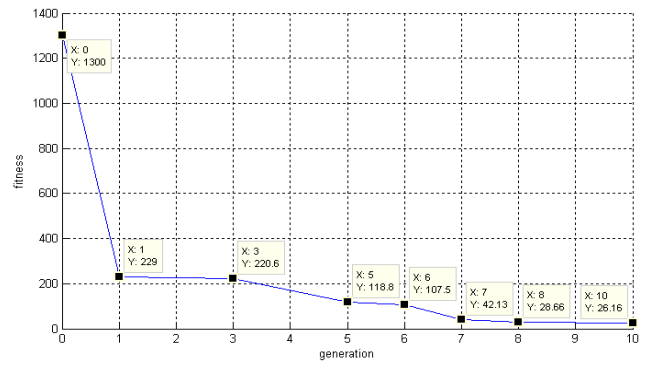


Figure 5 The best adaptive fitness in each generation (Test 2).

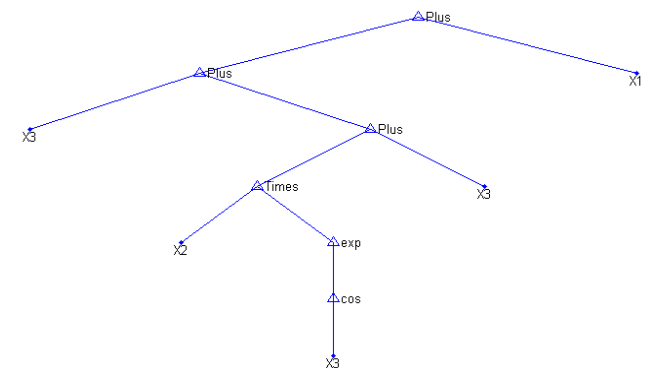


Figure 6 The structure of the combination model (Test 2).

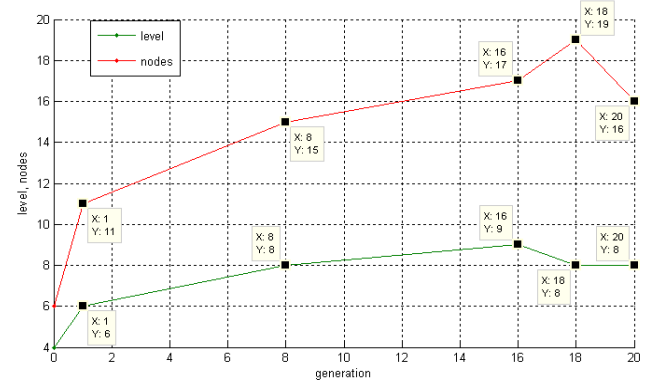


Figure 7 The best number of levels and nodes in each generation (Test 3).

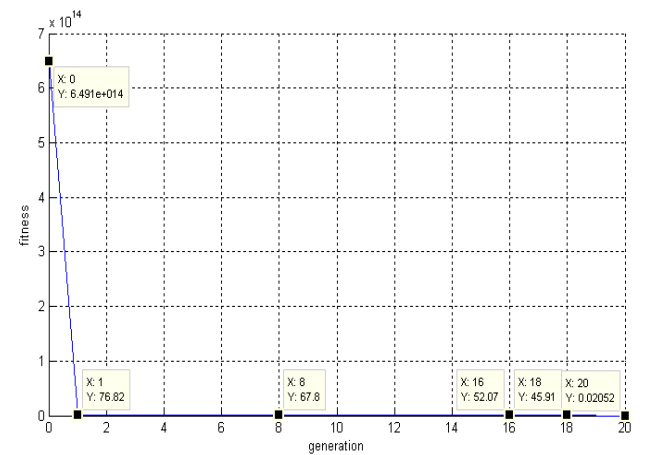


Figure 8 The best adaptive fitness in each generation (Test 3).

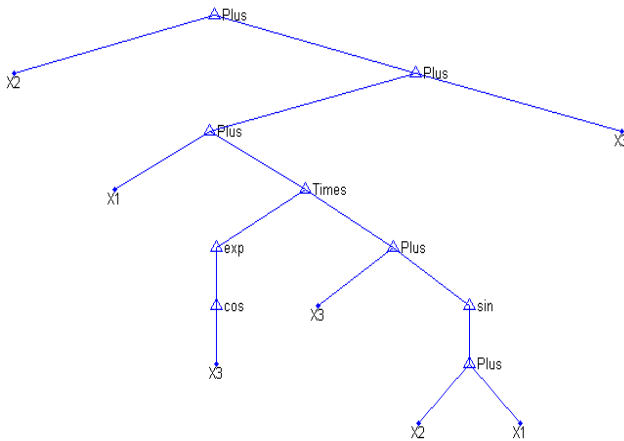


Figure 9 The structure of the combination model (Test 3).

2) The capacity of the population and the number of generations have great influence on the efficiency of the algorithm. Thus, they must be set carefully according to the requirement.

3) The algorithm produces combination models only based on samples, thus the results can reach the requirement on data. The combination model itself may be difficult to understand, and there may be logic conflict sometimes. Therefore, the algorithm can be used as decision support tool and provide models for reference.

4) There is some randomness for the results of the algorithm, and the same initial conditions can not always obtain the same combination model. Thus, the algorithm must be performed for a number of times, and the best combination model is selected from the results.

#### IV. CONCLUSION

Current model selection algorithms do not pay much attention to the data of the samples and can not present decision support models based on data. In this paper, a new model combination algorithm is developed based on genetic programming. Our algorithm can present combination models through the sample data. Experiment results show that the algorithm can efficiently utilize the potential information included in the sample data and present the structure of combination model for reference. The algorithm is independent of the domain knowledge; thus nonprofessionals can also create their combination models according to their sample data. But the efficiency of algorithm depends greatly on the samples and can not judge the validity of the sample data. Also, the produced combination model is not easily understood. All of these are the future work of our research.

#### ACKNOWLEDGMENT

This research work is partially supported by the National Natural Science Foundation of China under Grant No.70701038 and No.70771109.

#### REFERENCES

[1] Klein G, "Developing model string for model management". *Journal of Management Information System*. 1986,vol.4, pp.94-110.

- [2] Zhang Zhihua, "Model Selection based on expert knowledge", Shanghai Jiao Tong University (in Chinese), 2005.
- [3] Banejee S and Basu A. "Model Type Selection in an Integrated DSS Environment", *Decision Support Systems*. 1993, vol.9, pp.75-89.
- [4] Yanghe Feng, Chaofan Dai, Su Deng."Algorithm of Model Selection in Decision Support System". *ITCS2009*, pp.334-337, Kiev,Ukraine,2009,.
- [5] Bengio,Y., Gradient-based optimization of hyper-parameters. *Neural Comput*. 2000, 12 (8), 1889-1900.
- [6] Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., Choosing multiple parameters for support vector machines. *Mach. Learn*. 2002, 46 (1-3),131-159.
- [7] Song X.Y., Lee S.Y. A Bayesian model selection method with applications, *Computational Statistics & Data Analysis*. 2002, 40, 539-557
- [8] Gutiérrez-Peña E., Rueda R., Contreras-Cristán A. Objective parametric model selection procedures from a Bayesian nonparametric perspective, *Computational Statistics and Data Analysis*, 2009, 53, 4255-4265
- [9] Duan Jiangjiao, Wang Wei, Zeng Jianping, Zhang Dongzhan, Shi Baile. A prediction algorithm for time series based on adaptive model selection, *Expert Systems with Applications*. 2009, 36, 1308-1314
- [10] Curry B., Morgan P.H. Model selection in Neural Networks: Some difficulties, *European Journal of Operational Research*. 2006, 170, 567-577
- [11] Egrioglu, E., Aladag, C. H., & Gunay, S. A new model selection strategy in artificial neural networks. *Applied Mathematics and Computation*. 2008, 195(2), 591-597.
- [12] Alan Brookharta M., Van Der Laanb M. A semiparametric model selection criterion with applications to the marginal structural model, *Computational Statistics & Data Analysis*. 2006, 50, 475-498
- [13] Lee C., Doong S. Evolutionary selection of model for time constrained decision problems: A GA approach, *Expert Systems with Applications*. 2008, 34, 1886-1892
- [14] Celik M., Deha Er I. Fuzzy axiomatic design extension for managing model selection paradigm in decision science, *Expert Systems with Applications*. 2009, 36, 6477-6484
- [15] Hutter M., Tran M.N. Model selection with the Loss Rank Principle, *Computational Statistics and Data Analysis*. 2010, 54, 1288-1306
- [16] Salibian-Barrera M., Van Aelst S. Robust model selection using fast and robust bootstrap, *Computational Statistics and Data Analysis*. 2008,52, 5121-5135
- [17] Çetin M. Robust model selection criteria for robust Liu estimator, *European Journal of Operational Research*. 2009, 199, 21-24
- [18] Daye Z.J., Jeng X.J. Shrinkage and model selection with correlated variables via weighted fusion, *Computational Statistics and Data Analysis*. 2009, 53, 1284-1298
- [19] Huang C., Lee Y., Linb D., Huang S. Model selection for support vector machines via uniform design, *Computational Statistics & Data Analysis*. 2007, 52, 335-346
- [20] Peng X., Wang Y. A geometric method for model selection in support vector machine, *Expert Systems with Applications*. 2009, 36, 5745-5749
- [21] Topcu, Y. I., & Ulengin, F. Energy for the future: An integrated decision aid for the case of Turkey. *Energy*. 2004,29(1), 137-154.



**Caofan Dai** born in 1973, is an associate professor in C4ISR Technology Key Laboratory of Science and Technology for National Defense, National University of Defense Technology. He received the Ph.D. degree in Management Engineering in 2002 from National University of Defense Technology.

His research is focused on group decision science, decision support systems, soft computing and Cyber-Physical systems.

**Yanghe Feng** born in 1985, is a doctoral student in C4ISR Technology Key Laboratory of Science and Technology for National Defense, National University of Defense Technology. His research is focused on computational organization science, decision support systems.

**Jianmai Shi** born in 1982, is a doctoral student in C4ISR Technology Key Laboratory of Science and Technology for National Defense, National University of Defense Technology. His research is focused on group decision science and decision support systems.