

# Simulation Algorithm of Shortest Match Path Based on Time Series

M.J.Wu

Zaozhuang University  
Zaozhuang, Shandong Province, China

**Abstract**—Two new corresponding algorithms are proposed based on the analysis of several classical algorithms to solve the problem of the shortest match path based on time series. Experiments show formatting reverse branch algorithm with high execution efficiency can obtain all matching path. Dynamic path matching algorithm enhances the real-time property on the basis of getting all matching path. The new algorithms have incomparable advantage on both the accuracy and efficiency.

**Keywords**—simulation technology; shortest match path; format

## I. INTRODUCTION

The physical phenomena and physical system in the objective world can be described by mathematical equations. The simulation is the computation process of establishing the corresponding mathematical model of the physical system on the computer[1-2]. In the study of computer simulation technology, the matching problem of test data and template stored in advance is often encountered. The action is identified by finding the shortest characteristics distance between experimental data and the action template.

Assume that there are a set of test action data sequence  $v=(V_0, V_1 \dots V_m)$  and a set of action template sequence  $T=(T_0, T_1 \dots T_n)$  ( $m > n$ ), both sequences satisfy the action time sequence. The question is to find matching path between the two groups of sequences according to the characteristics distance matrix and the matching path must satisfy the time series. Each action of test action data must appear in the matching path, and the action template sequence does not consider it necessary.

There are many classic algorithms[3]. The advantage of exhaustive algorithm is that it can get the optimal solution by exhaustive comparing all possible solutions under the constraint conditions. It can guarantee the optimality of the solution. However, because the algorithm must consider all possible solutions, it has highest time complexity and space complexity. So it is suitable for the small-scale problems. In this problem, the approximate time complexity is  $O(n^{m/2})$ . The advantage of greedy algorithm has low time complexity and is used in all kinds of aspects in algorithm design. It is not easy to prove the correctness of greedy algorithm and sometimes the solution is not the optimal. Compared with the exhaustive algorithm, the biggest drawback of simulated annealing algorithm and genetic algorithm [4] is that the implementation mechanism is multifarious. They can be used to solve large scale problem. For small scale problems, its calculation process is worse than greedy algorithm and other methods, even worse than exhaustive algorithm. These

algorithms have more or less disadvantages (such as high time complexity exhaustive algorithm, simulated annealing algorithm with computational complexity, etc.)

For such problems of the shortest match path based on time series, two new corresponding algorithms are proposed based on the analysis of several classical algorithms and they are formatting reverse branch algorithm and dynamic path matching algorithm. The experiments show formatting single path search algorithm needs less storage space with high efficiency, and can find the first match path. Formatting reverse branch algorithm with high execution efficiency can obtain all matching path. Dynamic path matching algorithm enhances the real-time property on the basis of getting all matching path.

## II. RELATED DEFINITION AND THEOREM

We use the graph theory to solve the problem[5-6].

**Definition 1.** If the vertices set of a simple graph  $G$  is divided into two disjoint non-empty set  $V$  and  $T$ , the two vertices associated with each edge of the graph  $G$  are in  $V$  and  $T$  respectively.  $G$  is called the bipartite graph (or even graph), note as  $G = \langle V, T, E \rangle$ , where  $E$  is the set of edges. Association vertices of  $E$  in  $G$  are called saturation point.  $|E|$  is the length of the edge set.

**Definition 2.** For bipartite graph  $G = \langle V, T, E \rangle$ , if  $|V|=m$ ,  $|T|=n$ , and each vertex in the  $V$  is adjacent to all vertices in  $T$ , the graph is called complete bipartite graph, namely as  $K_{m,n}$ .

**Definition 3.** For bipartite graph  $G = \langle V, T, E \rangle$ ,  $X$  is a subset of the  $E(G)$ , if any two edges in  $X$  are not adjacent in  $V$  and are adjacent in  $T$ ,  $X$  is called a single-ended adjacency matching of graph.

**Definition 4.** For bipartite graph  $G = \langle V, T, E \rangle$ , if a single-ended adjacency matching  $X \subseteq E$  of  $G$  can saturate each vertex in the  $V$ ,  $X$  is called a complete single-ended adjacency matching.  $|X|$  is match length.

The problem given with the restrictions of the time constraint is to find the matching path which must satisfy the time series, namely the action in the back cannot appear in the front. For such problems, we give the following definitions.

**Definition 5.** For bipartite graph  $G = \langle V, T, E \rangle$ , if complete matching  $X \subseteq E$  satisfy the following formulas,

$$X = \left\{ e_{i,j} : V_i \right. \\ \left. \rightarrow V_j \right| \begin{array}{l} i = 0, 1 \dots m; \\ j \in \{0, 1 \dots n\}; \\ \forall e_{i_1, j_1}, e_{i_2, j_2} \in G, \text{ if } (i_1 > i_2), \text{ then } j_1 \geq j_2. \end{array}$$

X is called a complete single-ended adjacency matching based on time sequence. Here, the data with small subscript comes first.

Definition 6. (1) If each edge of the graph has a weight, this graph is called weighted graph.

(2) If each edge of the bipartite graph has a weight, this graph is called weighted complete bipartite graph, denoted  $G = \langle V, T, E, W \rangle$ , where W is the weight.

Definition 7. For weighted bipartite graph  $G = \langle V, T, E, W \rangle$ , the single-ended completely adjacency matching with smallest weight sum is called the best single ended adjacency matching path, short for the best match path. If X is a matching path, the sum of weight on the path is called the length of matching path.

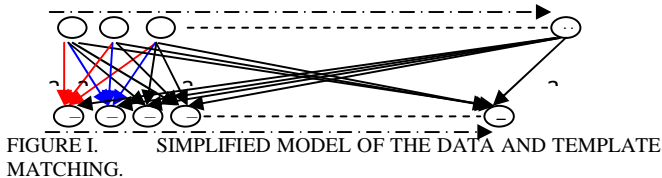
In this paper, the problem is equivalent to find the best single-ended adjacency matching path based on time sequence in the weighted complete bipartite graph  $G = \langle V, T, E, W \rangle$ .

Theorem 1. For weighted complete bipartite graph  $G = \langle V, T, E, W \rangle$ , there are at least one best single ended adjacency matching path based on time sequence and the length of matching path is  $|V|$ .

Proof: From definition 5, we know that complete single-ended adjacency matching based on time sequence needs to satisfy the time relations. The first action is matched firstly, the next action can't be matched ahead. The next matching scope can only be equal to or less than the previous. The search range of each row is likely to become smaller and it will not influence the minimum value within the limited scope. So there is at least a best single ended adjacency matching path based on time sequence, and the length of matching path is  $|V|$ .

### III. ALGORITHM DESCRIPTION

The matching model is as shown in Figure 1, where  $V_0, V_1 \dots V_m$  are ordered by increasing time and  $V_i$  is the matching data.  $T_0, T_1 \dots T_n$  is ordered by increasing time and  $T_i$  is the template action.  $a_{ij}$  is distance of matching path. For example,  $a_{01}$  is distance from  $V_0$  directly to the  $T_1$  and other similar.



#### A. Formatting Reverse Branch Search Algorithm

1) *Algorithm Characteristics* on the basis of formatting

single path search algorithm, after the minimum value of each line is obtained, record their columns. When the final line is reached, according to the records of the columns, search all of the minimum value with bottom-up way using breadth-first search strategy. All the matching paths can be obtained by this method.

2) *Algorithm Description* The algorithm is proposed on the basis of formatting single path search algorithm. After all of the data is formatted, the following steps should to do.

1) Find out the minimum value from the final line. There may be multiple minimum, all of them should be written down.

2) Breadth-first search is carried out from bottom to up on each of the minimum and find the minimum of previous line. The search range is from the upper of current minimum to the left. Find all of the minimum values and save their location.

3) Make the current minimum value as initial search point, repeat step 2 until the first line, all the requested matching path can be found.

input	10	5	15	17	8	8	4	10	11
format	11	16	7	8	9	10	19	3	15
sum	21	21	12	13	14	15	23	7	19
	21	21	12	12	12	12	12	7	7
	10	15	16	3	11	8	8	19	20
	31	36	28	15	23	20	20	26	27
	31	31	28	15	15	15	15	15	15
	9	8	7	10	11	5	15	16	17
	40	39	35	25	26	20	30	31	32
	40	39	35	25	25	20	20	20	20
	30	30	5	60	60	60	60	20	70
	70	69	40	85	85	80	80	40	90

FIGURE II. THE EXAMPLE OF FORMATTING REVERSE BRANCH SEARCH ALGORITHM.

input	10	5	15	17	8	8	4	10	11
sum	11	16	7	8	9	10	19	3	15
	21	21	12	13	14	15	23	7	19
	10	15	16	3	11	8	8	19	20
	31	36	28	15	23	20	20	26	27
	9	8	7	10	11	5	15	16	17
	40	39	35	25	26	20	30	31	32
	30	30	5	60	60	60	60	20	70
	70	69	40	85	85	80	80	40	90

FIGURE III. THE EXAMPLE OF DYNAMIC PATH MATCHING ALGORITHM.

#### B. Dynamic Path Matching Algorithm

1) *Algorithm characteristics* This algorithm is similar to the formatting method above. This method carries out first ly real-time calculation with top-down method. When the output of test action data completes, we just find out all of the minimum value. This algorithm also can find the all of the shortest features distance.

2) *Algorithm description* 1) The data of first row is added 0 implicitly.

2) Starting from the next line, each data of the current line searches the minimum value from right to left from the previous line. The search range is from upper the current minimum value to the left. Find the minimum value on the left side of it, save its position. Calculate the sum of minimum value and the current input data and save till the end of the

line.

3) Save the path of the first sum  $a_0$  and backward search the sum value  $a_i$ . If they are equal, we save the path. If  $a_i$  is greater than  $a_0$ ,  $a_i$  is ignored. If  $a_i$  is smaller than  $a_0$ , save it as a new search position and search backwards until the end of the line.

4) Repeat 2) and 3) until all the input line search is completed.

5) Search all the paths obtained, the path with minimum value is the matching path (may be more).

IV. ALGORITHM COMPARISON

To solve this problem, comparison on accuracy and time efficiency of five algorithms (formatting reverse branch search algorithm, the dynamic path matching algorithm, greedy algorithm, sorting exhaustive algorithm, exhaustive algorithm) was carried on. 50 groups of  $15 \times 10$  distance matrix given were tested. The experiments show that the accuracy of formatting reverse branch search algorithm, the dynamic path matching algorithm, exhaustive algorithm are all 100%. The accuracy of greedy algorithm is 29.5% and the sorting exhaustive algorithm is 67.5%.

TABLE I. THE ACCURACY.

algorithms	The accuracy
formatting reverse branch search algorithm	100%
dynamic path matching algorithm	100%
greedy algorithm	29.5%
sorting exhaustive algorithm	67.5%
exhaustive algorithm	100%

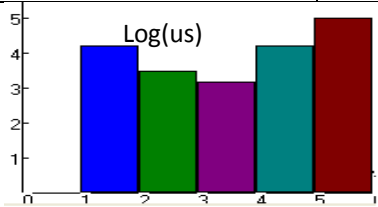


FIGURE IV. COMPARISON ON EXECUTION EFFICIENCY (THE ALGORITHMS ARE FORMATTING REVERSE BRANCH SEARCH ALGORITHM, DYNAMIC PATH MATCHING ALGORITHM, GREEDY ALGORITHM, SORTING EXHAUSTIVE ALGORITHM, EXHAUSTIVE ALGORITHM FROM LEFT TO RIGHT.)

For execution time, the difference of execution time on formatting reverse branch search algorithm, dynamic path matching algorithm and the greedy algorithm is not big, the exhaustive algorithm is maximum as shown in table 1 (the machine configuration: Pentium (R) 4, CPU (2.66 GHz), memory (256 m)).

From the above results we can see that two new methods in both the accuracy and efficiency have incomparable advantage. They are effective methods on the solution of this kind of problems.

V. CONCLUSION

In this paper, the new algorithms proposed in this paper not

only can solve the problem of finding a single matching path, but also can find all of the matching paths. The solving process shows the time complexity of the algorithms is extremely low, speed is high and little space is taken up. The real-time of dynamic path matching algorithm with strong applicability is very good. The above two kinds of algorithms can be optionally use to solve the problem of the shortest match path based on time series.

REFERENCE

[1] Wu Chongguang. simulation technique. Chemical Industry Press. 2000.  
[2] XiongGuangleng. computer simulation applications. Tsinghua University Press. 2000.  
[3] Zhou Peide. The design and analysis of algorithm. China Machine Press. 2002.  
[4] Zhou Ming. genetic algorithm theory and application. National Defence Industry Press. 2000.  
[5] Yang Zhensheng. Combinatorial mathematics and its algorithm. University of science and technology of China press. 2003.  
[6] Yin Jianhong. Graph theory and its algorithm. University of science and technology of China press. 2003.