DYNAMIC PROGRAMMING AS INTRUSION DETECTION TOOL

Vukašin Pejović, Slobodan Bojanić, Octavio Nieto-Taladriz, Departamento de Ingeniería Electrónica, Universidad Politécnica de Madrid, Spain
Jelena Popović, Elektrotehnički Fakultet, Univerzitet u Beogradu

Abstract – Dynamic programming is used as powerful problem solving tool in many areas of modern science. It is essential method in area of bioinformatics as it represents a basis for most bioinformatics algorithms. Versatility of this method enables its use in area of intrusion detection while hardware implementation provides necessary speed up for real time, modern network speed compliant, implementations.

1. INTRODUCTION

Dynamic programming (DP) was introduced by Richard Bellman in 1957 as cost based representation of optimization problem and is used as powerful tool for problem solving in many areas of modern engineering. It is also used in economics and business or even in agriculture i.e. actually in any field of science where an optimization is needed. It can be presented in continuous and discrete domain although discrete variant is far more suitable for modern digital implementation and almost all mentioned application use this type of approach.

This method has been proved to be extremely useful in bioinformatics area as it is the basis of almost every comparison method for genetic material. Classic algorithms for this purpose are Needleman-Wunsch global alignment (NW) algorithm [1] and Smith-Waterman local alignment (SW) algorithm [2]. The software implementations of these algorithms are very time consuming as it can be expected of the algorithms based on matrix structure. Hardware implementations as intrinsically parallel provide speed improvement balanced with area utilized.

Intrusion detection as important and even indispensable segment of network and by that global security has to follow adequately the traffic and bandwidth increase of network segment of network and by that global security has to follow improvement balanced with area utilized.

2. DYNAMIC PROGRAMMING ALGORITHMS

Introduced as special genetic material comparison algorithms NW and SW algorithms have been used mostly in that domain [6]. It is possible to apply them on almost any string comparison problem having in mind that there has to be a reason for that kind of approach. These algorithms are very suitable for environments, such as genetics, because they perform similarity or difference measurements. In genetic material world, although comparison material is unique for every owner, there are always similarities, as some proteins and their combinations are repeating. As mentioned, this type of environment is ideal for comparison measurement, providing information of potential alignment of genetic material.

\[
M_{i,j} = \max\{M_{i-1,j-1} + \Delta_k\} \\
j \in (1, I - i), k \in (1, J - j)
\]

(1)

The DP algorithms generate a similarity matrix whose elements are cost values that have to be compensated in order to transform one sequence to another. Higher the value, lower the similarity. In NW case matrix is generated according to formula (1). In formula (1), for NW case, \(\Delta = 1\) when compared characters match and \(\Delta = 0\) for mismatch. \(M_{i,j}\) is a correspondent matrix value in with coordinates \(i, j\), while \(I\) and \(J\) are matrix dimensions. Formula (1) allows any gap length when matching arrays, this might be more rigorous, depending on need and also it is possible to introduce mismatch costs. NW algorithm generates higher value for more matches, thus higher value shows more similarity. Fig 1 presents matrix generated according to formula (1). Matrix generation direction is show with arrows.

\[
\begin{array}{ccccccc}
A & A & B & B & B & C & C \\
\hline
A & 4 & 4 & 3 & 2 & 0 & 0 \\
B & 2 & 2 & 3 & 3 & 1 & 0 \\
B & 1 & 1 & 2 & 2 & 2 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{array}
\]

In the SW case, the formula (2) presents a matrix generation rules. Where \(a_i\) and \(b_j\) are units of sequences that are being compared on positions \(i\) and \(j\), respectively, \(H_{i,j}\) is matrix value on position with coordinates \(i, j\), respectively and \(W_k\) is the weight for deletion of \(k\) sequence units. Example matrix generated by SW algorithm is shown on Fig 2, using \(W_k = 1 + k/3\), \(s(a_i, b_j)\) has a value 1 in the case of a match, and a value of \(-1/3\) in a case of a mismatch. SW matrix is generated starting from matrix cell with indexes 0,0 and also shows it shows the best similarity by highest value in matrix.

\[
H_{i,j} = \max_{k \geq 1} \left\{ H_{i-1,j-1} + s(a_i, b_j), \max_{k \in \{1, \ldots, I\} \setminus j} H_{i-1,j-1}, \max_{k \in \{1, \ldots, J\} \setminus i} W_k \right\}
\]

(2)

Differences between algorithms that have to be noted are that NW is a global alignment method and is more suitable for sequences of approximately same length which
have some level of obvious similarity, because scoring system highlights that kind of similarities, while SW is a local alignment method and is to be used for comparisons of two sequences with highly different lengths. This means that global alignment gives an overall similarity picture while local alignment highlights locally similar regions.

<table>
<thead>
<tr>
<th>A</th>
<th>A</th>
<th>B</th>
<th>B</th>
<th>B</th>
<th>C</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1.33</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0.33</td>
<td>2.66</td>
<td>1.33</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig 2 Cost matrix generated by SW algorithm

Switching, again, to intrusion detection domain, algorithm proposed by Coull et al. [5] differs from original SW and NW algorithms by different scoring, cost, values. The input data in this algorithm are user signatures, easily explained as a set of commands that user executes while using a system, and can be obtained from system log files. Although structures, user signature, are obviously different from genetic material they still have most important property necessary for applying GP, and that is the fact that those are derived from the same limited set of commands, so the similarity is inevitable.

By use of different scoring scheme it is possible to override problems encountered with strictly global or strictly local alignment algorithms incorporating characteristics of both [5], creating semi-global alignment algorithm, shown to perform very good as a tool used for masqueraded user detector [7]. Semi-global alignment approach balances both the local and the global approaches and is a good tool for use on user signature structures, as which type of approach should be preferred is deeply dependent on very user being analyzed. The goal of this type of approach is potential recognition and anticipation of user behavior, an important security factor.

3. HARDWARE IMPLEMENTATION

Final and probably crucial step in providing efficient use of DP in intrusion detection systems (IDS) must be speed, or throughput increase to levels compatible with actual network throughputs, 2.5/10/40 Gbps for STM-16/64/256 as specified for synchronous digital hierarchy transmission. In order to provide improvements in that context inherent parallelism of algorithms must be exploited. It is possible to use mentioned characteristics representing appropriately the matrix structures DP algorithms are based on, for example in modern FPGA technology, by systolic array or processing element approaches, which are basically various implementable interpretations of the same problem, the matrix implementations. Besides only implementation capabilities modern FPGA chips can provide other potentially interesting properties, such as partial or complete off and online reconfiguration, adaptive to system changes and other changeable factors or use of embedded microprocessors in reconstruction and interpretation processes of DP based algorithm results. Therefore even some processing power can be translated to hardware subsystem.

Interconnecting host system, PC for example via fast interconnection buses, such as PCI-Express, could provide appropriately wide IO connection for hardware subsystem and host system communications and interactivity. Hardware based security supervising system constructed on presented principles would be necessary security enhancement, enforcing “user consciousness” of a system. That means that system would have appropriate level of “consciousness” about users that are using it and their behavior, enabling and improving decision making capabilities related or based on those properties.

4. CONCLUSION

Here is presented an approach for improvement of system security robustness based on specialized hardware implementation of dynamic programming algorithms. A compact solution presented here reflects one of possibilities of optimization based dynamic programming approach. Since optimization theory provides various approaches, such as linear programming, heuristic programming, integer programming, non-linear programming and others which are yet to be tested and their potentials evaluated, while further work on improvement of presented DP algorithms is pending such as other scoring schemes, additional hardware implementation improvements, all in order to provide more compact, reliable, and fast enough solution.

Idea of implementing independent hardware based security device would besides the essential speed improvement provide clearing of all software and operating system dependency issues, thus fundamentally increasing system defense. It might potently provide versatile system recovery characteristics, and also attack analysis capabilities, all through one compact, robust and reconfigurable solution, as proposed.

LITERATURE