A Comparison of Optimal Path Trace Back Sizing Using Genetic Algorithm (GA)

Nur Dalilah Ahmad Sabri, Nur Farah Ain Saliman, Syed Abdul Mutalib Al Junid,
Zulkifli Abd Majid, Nooritawati Md Tahir
Faculty of Electrical Engineering,
Universiti Teknologi Mara, 40450
Shah Alam, Malaysia
dalilah_sabri@yahoo.com

***Abstract* — This project presents a comparison over different sizing of Optimal Path Traceback (OTB). The technique involved to varying the sizing of OTB using Genetic Algorithm (GA). Commonly, GA consists of important steps which are selection of parent, crossover and finding the most similarity offspring. The idea behind OTB is to find the likely sequence alignment between the two parents. Therefore, these project will evaluate the system performance in terms of its accuracy and speed over DNA sequence alignment. Selected simulation results using ALTERA Quartus Cyclone IV E device family (EP4CE115F29C7) are presented to verify the evaluation operations after the design implementation.**

Keywords — DNA, Sequence alignment, genetic algorithm, optimal path trace back, bioinformatics.

I. INTRODUCTION

There are several methods exist in performing acceleration of sequence alignment activities such as FASTA [1] , BLAST [2], HMMER [3] and each of the methods has its own advantages. BLAST and FASTA method is popular because they are known as heuristic solution and in great condition when it came in speeds [4]. While the advantage for HMMER is temporal classification [5]. These methods also have been shown to be useful for many application problems. However each of these algorithms has not mentioned an optimal path guarantee [1] [6]. While with the ever increasing of human inhabitant, the biological sequence alignment has always increased and effects to the time consume [7]. The system biology of sequence alignment is a very challenging task in Bioinformatics because of this matter[8][9][10].

Nowadays, there are a lot of research have been conducted to optimize and accelerate the DNA sequences alignment which emphasized on the high sensitivity Smith-Waterman Algorithm (SW) method [11][4]. This algorithm has been proposed by TF Smith and MS Waterman in 1981 for solving the problem of DNA sequences alignment using the dynamic programming approach. Moreover, this algorithm consists of three main steps (initialization, Matrix Fill and Trace Back) [12] that have been developed in the SW. The SW method has the capability of having the best accuracy of searching DNA sequence alignment. But when it deals with very long bases pair of DNAs, the memory space will increase and eventually slows the speed [4]. The purpose of the research is to explore with the SW module and aiming only at the part where the trace back (OTB) lies and come up with an evaluation in terms of speed with different size of population (n) based on original GA.

GA has been discovered by John Holland in 1970s as recorded in [13].The GA can build data by taking random probabilities. Besides that, it is used for optimization process and purpose needed to develop the programming codes. GA is a random probability that based on the principle of common selection and natural genetics which gave many successes in finding solutions to the optimization problems [14] [15] [16]. For these reasons, GA is chosen to merge with the trace back module.

Later, the highest probability will be selected in the OTB to fulfill the criteria of the project. The objective considered as to develop an accurate optimal path trace back using GA. Other than that, it is also to minimize the time complexity and reduces the memory space. Moreover, this method involves with the heuristic approach which creates the initial population with better fitness function.

From this paper, the work and result are organized into 4 sections. This section is purposely for the overview of this research in the general manner. In section II tells about the concept of using GA for optimal path especially on the trace back issues and will be detailed elaborated. Section III is about the result and discussion. Lastly, the conclusion is in section IV.

II. GENETIC ALGORITHM IMPLEMENTATION

GA is an evolutionary algorithm [17]. Besides that it consists of several operators that need to be concerned which are selection, crossover and offspring. These operators will be included in the population and chromosomes for the research purpose. In terms of selection, two chromosomes will choose to generate random numbers as to initialize and it is generally known as parents. Then, both chromosomes went through the process of data crossover in order to develop children which the scientific terms is an offspring [14]. Heuristic approach is used for the initial population. The process of GA is shown as in Fig.1.

Some parts of the data will converge to the optimum solution. The nearer a solution to the local maximum will

have greater fitness. Fitness is the most important part because it is the part where the function to be optimized. In this research, a method is developed for recognizing the optimal score path which combines GA and a part of OTB approach.

A. Step 1:

The author defines the two population parameter to represent for parent search and target of DNA sequence data as S and T with the assumption that the length of S and T as S [3:0] and T [1:0]. Then, the author assumes the number input of base pair for S as 1010 and T as 10. S is assuming as the population number of parents while as for T, T is a singular parent. Since T is in 2 bits, The S must follow the number of bit T. The input number of S will be separate into 10, 01 and 10.

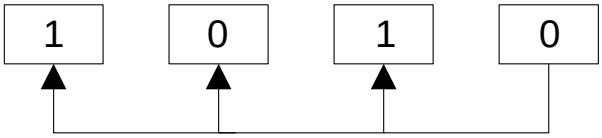

Fig.2. The population S is being selected.

Fig.2 shows how the population S is being selected. The number input of S will be separate into 10, 01 and 10 as in Table I.

TABLE I. RANDOM POPULATION OF PARENT S

Random Population of parents Search (S)		
S [1:0]	**S [2:1]**	**S [3:2]**
10	01	10

Then, the crossover is implemented to get the offspring by using XOR gate. Since assuming it as 2 bits per input then the number of S and T for crossover is S [1:0] = 11, S [2:1] = 00, S [3:2] = 11. It can illustrate as in the Table II. The logical equation XOR can be denoted as in (1).

$$\bar{A}B \oplus A\bar{B} = C \qquad (1)$$

Based on (1), the result offspring can be developed by assuming A as the target sequence (T), while B will assume as search sequence (S).

Example

1. $\overline{10}(10) + 10(\overline{10}) = 01(10) + 10(01) = 00$
2. $\overline{10}(01) + 10(\overline{01}) = 01(01) + 10(10) = 11$
3. $\overline{10}(10) + 10(\overline{10}) = 01(10) + 10(01) = 00$

TABLE II. RANDOM POPULATION OF PARENT S

Base sequence (T)	**Base sequence (S)**	$\bar{T}S \oplus T\bar{S}$
10	10	00
10	01	11
10	10	00

B. Step 2:

To achieve the score value, the author uses the fitness function to come out with the score value of matching residues between offspring and target sequence. This idea of the fitness function is refer from the previous researcher [18].

Fitness (score) = Number of matching residues

This step is conducted to find out which offspring will have the high similarity with the target parent. In order to come out with the similarity sequence score value, this comparison will use logical expressions of AND gate. Equation (2) shows on how to determine the similarity between children and parent DNA sequence.

$$A \,\&\, B = C \qquad (2)$$

Based on the (2), the result for similar sequence can be developed by assuming A as the target sequence (T), while B is assume as offsprings sequence. The offspring A&B result is listed in Table III.

Example

1. 10 & 00 = 00
2. 10 & 11 = 10
3. 10 & 00 = 00

TABLE III. OFFSPRINGS

Base sequence (T)	**Base sequence (S)**	$T\&S$
10	10	00
10	01	10
10	10	00

Then, score value will be determined by adding up each bit from previous comparing results. It can be determined as in the example.

Example

1. 0 + 0 = 0 (Score A)
2. 1 + 0 = 1 (Score B)
3. 0 + 0 = 0 (Score C)

C. Step 3:

Finally in the trace back, there will be three output i.e. T optimal, S optimal and Score optimal. The origin T will simultaneously become T optimal. Meanwhile the S optimal depends on the output score.

Example

1. (Score A > Score B) & (Score A > Score C) = Score optimal = Score A

2. (Score B > Score A) & (Score B > Score C) = Score optimal = Score B

3. (Score C > Score B) & (Score C > Score A) = Score optimal = Score C

The example above shows how Score optimization will be determined. Meanwhile, S optimal depends on the results score. It can be define as an example below:

Example

1. (Score A > Score B) & (Score A > Score C) = S optimal = S [1:0]

2. (Score B > Score A) & (Score B > Score C) = S optimal = S [2:1]

3. (Score C > Score B) & (Score C > Score A) = S optimal = S [3:2]

III. Result And Discussion

The simulation results have been demonstrated by implement the genetic algorithm flow work in Quartus II (Version 10). This implementation used Cyclone IV (EP4CE115F29C7) as a target device and used Verilog hardware description language. Table IV gives a comparison logic element used for three different size of initial population. The design is getting simpler and more efficient in terms of resource utilization. First design, the used n=4 for population size it gives 677 logic element. In contrast for second and third design, the utilization resource consumes with 881 and 908 logic element. This is expected because the size and total logic element is rising directly proportional.

TABLE IV. COMPARISON UTILIZATION RESOURCES

Cell Design	Resource Utilization
Size parent (n=4)	677
Size parent (n=8)	881
Size parent (n=16	908

Table V, shows that the different performance of runtime between three designs in timing analysis. The timing analysis can be divided into three types of timing analysis. Firstly, Tsu worst case delay can be defined as the delay time of pin to register delay plus with the micro setup delay and minus with clock to destination register delay. Secondly, the worst case for Tco is the delay total from clock to source register delay plus with micro clock to output delay and register to pin delay. Lastly, worst case delay for Th is defined as the total of clock to destination register delay plus with the micro hold delay of destination register and minus with pin to register delay. The performance analysis of delay runtime found that the delay is increasing. This happen because of the runtime of delay is increasing parallel with the increasing size of n population. The time delay consume for population size increase as the number of population size increase.

TABLE V. ANALYSIS TIMING

Population size, *n*	Worst case Tsu (*ns*)	Worst case Tco (*ns*)	Worst case Th (*ns*)
4	6.981	7.242	0.089
8	8.814	9.620	0.815
16	10.047	10.80	0.915

IV. Conclusion

This project demonstrates the operation of GA incorporating with OTB. Selected simulation results prove that accuracy for different sizing is successfully realized. Finally, it can be concluded that when the size of parent increase the size of architecture design also increase. However, it only gave the range of 1.30 to 1.34. The increment of the design is considered small and efficient.

Acknowledgment

The authors would like to acknowledge the Ministry of Science, Technology and Innovation (MOSTI) and Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM) for providing financial support under Science Fund Grant (100-RMI/SF 16/6/2 (17/2012)) and laboratory facilities.

References

[1] S. A. Shehab, “Fast Dynamic Algorithm for Sequence Alignment based on Bioinformatics,” vol. 37, no. 7, pp. 54–61, 2012.

[2] M. Cameron and H. E. Williams, “Comparing compressed sequences for faster nucleotide BLAST searches.,” IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 4, no. 3, pp. 349–64, 2007.

[3] A. K. Hudek and D. G. Brown, “FEAST: sensitive local alignment with multiple rates of evolution.,” IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 8, no. 3, pp. 698–709, 2011.

[4] Y. Liu, D. L. Maskell, and B. Schmidt, “CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units.,” BMC research notes, vol. 2, p. 73, Jan. 2009.

[5] S. L. Scott, G. M. James, and C. a Sugar, “Hidden Markov Models for Longitudinal Comparisons,” Journal of the American Statistical Association, vol. 100, no. 470, pp. 359–369, Jun. 2005.

[6] L. Hasan and H. Zafar, “Performance versus Power Analysis for Bioinformatics Sequence Alignment,” vol. 10, no. December, pp. 920–928, 2012.

[7] J. Zola, M. Aluru, A. Sarje, S. Member, and S. Aluru, "Parallel Information-Theory-Based Construction of Genome-Wide Gene Regulatory Networks," vol. 21, no. 12, pp. 1721–1733, 2010.
[8] S. A. M. Al Junid, N. Md Tahir, Z. Abd Majid, Z. Othman, and K. K. Mohd Shariff, "Reducing memory complexity using data minimization technique on FPGA," 2012 International Conference on Computer & Information Science (ICCIS), pp. 431–434, Jun. 2012.
[9] H. Ramaswamy and C. Purdy, "An extended library of hardware modules for genetic algorithms, with applications to DNA sequence matching," 2008 51st Midwest Symposium on Circuits and Systems, pp. 209–212, Aug. 2008.
[10] L. Hasan, Z. Al-Ars, and S. Vassiliadis, "Hardware acceleration of sequence alignment algorithms-an overview," 2007 International Conference on Design & Technology of Integrated Systems in Nanoscale Era, pp. 92–97, 2007.
[11] K. O. Boateng and E. Y. Baagyere, "A Smith-Waterman Algorithm Accelerator Based on Residue Number System," vol. 5, no. 1, pp. 99–112, 2012.
[12] L. Hasan and Z. Al-Ars, "Performance improvement of the smith-waterman algorithm," Annual Workshop on Circuits, Systems and Signal …, 2007.
[13] M. Mitchell, "Genetic Algorithms," in MIT Press, 1996, p. 133.
[14] C. Liu and S. Liu, "The research on DNA multiple sequence alignment based on adaptive immune genetic algorithm," Proceedings of 2011 International Conference on Electronics and Optoelectronics, no. Iceoe, pp. V3–75–V3–78, Jul. 2011.
[15] C. B. Congdon, J. C. Aman, G. M. Nava, H. R. Gaskins, and C. J. Mattingly, "An evaluation of information content as a metric for the inference of putative conserved noncoding regions in DNA sequences using a genetic algorithms approach.," IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 5, no. 1, pp. 1–14, 2008.
[16] L.-Z. Liu, F.-X. Wu, and W. J. Zhang, "Inference of biological S-system using the separable estimation method and the genetic algorithm.," IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 9, no. 4, pp. 955–65, 2012.
[17] C.-H. Yang, Y.-H. Cheng, C.-H. Yang, and L.-Y. Chuang, "Mutagenic primer design for mismatch PCR-RFLP SNP genotyping using a genetic algorithm.," IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 9, no. 3, pp. 837–45, 2012.
[18] A. Bir and J. Dongardive, "Building Consensus of Human Papillomavirus using Genetic Algorithm," Nature & Biologically …, pp. 2–7, 2009.

APPENDIX

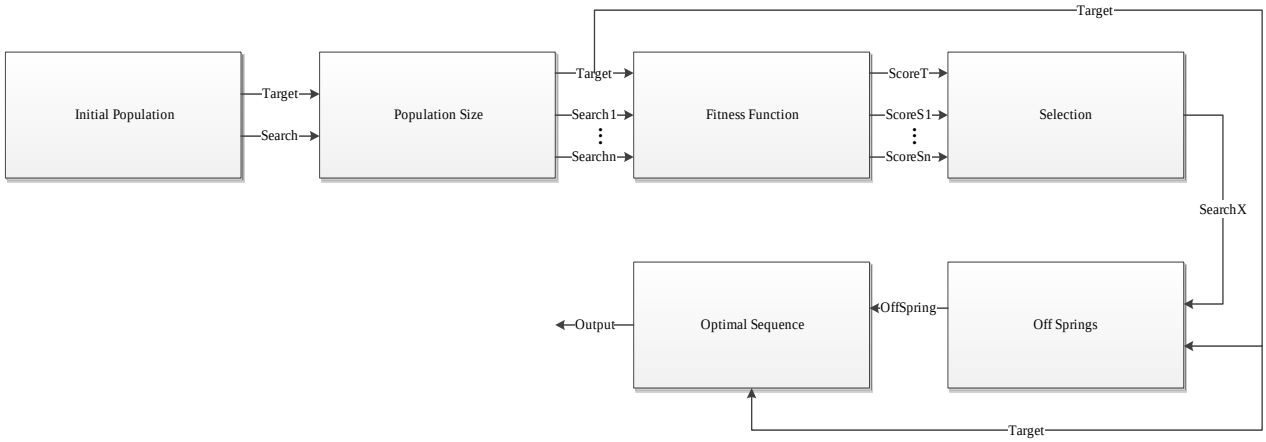

Fig.1. Working procedural of the Genetic Algorithm system.