Double Orthogonal Arrays Based Genetic Algorithm for Primer Design

Chungnan Lee, Yi-Te Li , Jain-Shing Wu, and Ta-Yuan Chou

Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan, ROC
cnlee@mail.cse.nsysu.edu.tw

Abstract. In this paper, a double orthogonal array based genetic algorithm is proposed to decrease the searching space and increase the feasible quality of primers. The key point of the proposed algorithm is to achieve the elitism goal by applying the orthogonal arrays (OAs) that is used in quality engineering with a small amount of experimental features. The result shows that the proposed algorithm converges faster than the traditional GA. Based on the design properties such as unique, GC content, melting temperature, temperature difference and annealing, the proposed algorithm can derive the solutions more precisely and efficiently. The primer pairs obtained can exactly match to the PCR traits.

Keywords: Primer Design, Orthogonal arrays, Genetic algorithm.

1 Introduction

With growing DNA researches, the cloning the DNA fragment becomes more and more important, since these researches need a large amount of the DNA sequence to make the experimental result more easily and clearly recognized. Polymerase chain reaction (PCR) [1], [2] is one method in bio-technology for fast and mass amplification of a specific DNA sequence. Finding a good primer pair is an essential step before performing PCR experiment. To yield a suitable primer pair, several major constraints such as the length of primer and its melting temperature, GC content, complement and specificity should be considered. All these constraints need to be fulfilled, in order to have a good primer pair. However, there are some conflicts between these constraints, such as the conflict between the specificity and the melting temperature. Therefore, efficiently finding the desired subsequences to exactly meet the constraints of primer design is a major issue.

Due to the practicability and generality of PCR, many approaches for the primer designs have been developed. The training finite state machines proposed by Ashlock *et al.* [3] trained primer pairs by using machine learning. PRIMEGENS [4] is used to select the gene-specific fragments and then design primer pairs for PCR. MethPrimer

[5], a program for designing bisulfite-conversion-based methylation PCR Primers, is based on the well-known primer design software Primer 3 [6]. Kämpke *et al*. [7] used dynamic programming to search the whole DNA space to find an optimal result. However the specificity constraint is not considered in these researches. Wu *et al*. [8] proposed an enhanced genetic algorithm for primer design, they considered not only the original constraints used in PCR experiment, but also other two most important constraints -- the specificity constraint and the restriction site. However, extra time is needed to ensure that these two constraints are satisfied.

The genetic algorithm [9] that adopts the spirit of natural selection gets good performance in the optimization problem. All individuals are evolved via the processes of *selection* (*reproduction*), *crossover*, *mutation* and other operations to search the optimal solution gradually. The individuals with stronger constitutions will survive rivalries and pass to the descendants. Since GA is sensitive to the parameter of GA, to find an optimal set of parameters often takes a long time, reducing the experiment times is necessary. The Taguchi method [10], [11], [12] is practical, experiment oriented in quality engineering and possess with quantitative qualities of robustness, statistic and convergence rapidly under the elitist strategy. Its purpose is to spend minimum cost and time to obtain better operating controls with the smallest experimental error. However, the Taguchi method needs to spend some additional cost in the computation of orthogonal arrays than the traditional method.

In order to avoid mis-priming situations and to increase the efficiency, double orthogonal arrays based (DOA-based) genetic algorithm (GA) is proposed to search the optimal primer pair. The proposed algorithm integrates the advantages of the traditional GA and the Taguchi method to obtain a better solution.

The remainder of this paper is organized as follows. In the next section, we describe the proposed algorithm for PCR primer design. Experimental results are given in Section 3. Conclusions are drawn in the last section.

2 The proposed algorithm

In this section, firstly, the essential related definitions of the proposed algorithm are given in Section 2.1. Then, the basic concept of the DOA-based subsystem is described in Section 2.2.

2.1 The essential related definitions of the proposed algorithm

There are some related definitions are given in the following paragraphs.

Length of DNA Sequence, target DNA and Primers

First, Let D_S be the DNA sequence template, which is denoted as the template of the base-nucleic acid code sequence of DNA. The length of D_S is denoted as $| D_S |$, where

$$| D_S | = \#A + \#T + \#C + \#G$$

where A, T, C, and G are the base-nucleic acid codes, and #A means the number of nucleic acid code "A"; #T means the number of nucleic acid code "T"; #C means the number of nucleic acid code "C"; #G means the number of nucleic acid code "G". The set of forward primer is a subsequence of D_S, and it is denoted as P_f,

$$P_f = \{S_i \mid i \in \text{interval of } [\ p_f^s,\ p_f^s + \alpha_f\ ]\ \}$$

where p_f^s is the start position of Primer P_f and α_f is the length of Primer P_f. Similarly, for reverse primer,

$$P_r = \{\ S_j \mid j \in \text{interval of } [\ p_r^s,\ p_r^s + \beta_r\ ]\ \}$$

where p_r^s is the start position of Primer P_r and β_r is the length of Primer P_r. Besides, the summation of p_f^s, α_f, target length, and β_r should not be larger than $| D_S |$.

The length of the primer should be within 18 to 26 mer. Besides, the difference between the length of the forward primer and reverse primer should under 3.

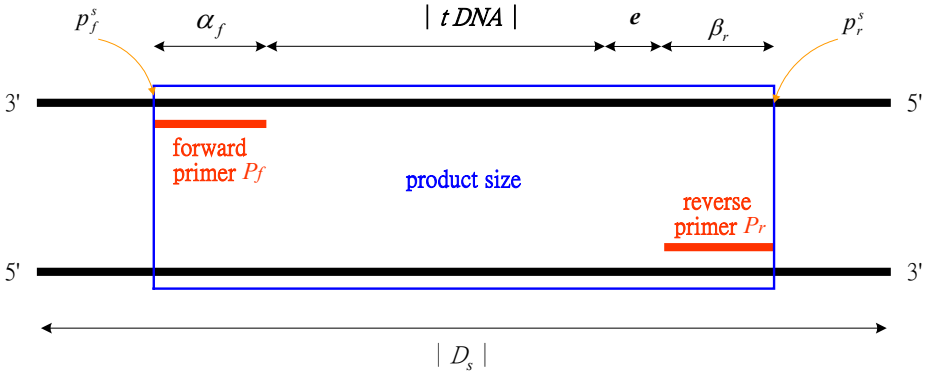

Figure 1. An example of a DNA and domains of related parameters

Figure 1 shows an example of a DNA and domains of related parameters. The total length of the target DNA (tDNA) is denoted as $| tDNA |$. The ε is the tolerance range of *tDNA* for controlling the product size of the PCR experiment.

Melting Temperature and GC Proportion

The melting temperature of a primer *P* is denoted as Tm(*P*) by using the rules proposed by Suggs *et al*. [13]. It is calculated by the summation of the total number of nucleic code "A" and "T" times 2 and the total number of nucleic code "C" and "G" times 4. and it is

$$\text{Tm}(P) = (\ \#A + \#T)*2 + (\ \#G + \#C)*4$$

Tm(*P*) should not be greater than 60℃

$$\mathrm{Tm}(P) \leqq 60℃$$

The temperature difference between forward and reverse primers is denoted as T_{diff}, and it is

$$T_{diff} = | \mathrm{Tm}(P_f) – \mathrm{Tm}(P_r) | .$$

In order to make the experiment successfully, the T_{diff} should be under 5℃. The GC proportion of the primer also affects the PCR experiment, since the higher GC proportion is, the higher melting temperature is. An appropriate GC percentage results in annealing temperature to fall in an acceptable range. The GC proportion GC(P) of the primer P is the 'G' and 'C' proportion of the primer, and it is

$$\mathrm{GC}(P) = [(\#G + \#C) / |P|]*100$$

and it should within 40% to 60%.

$$40\% \leqq \mathrm{GC}(P) \leqq 60\%$$

Complement

In base-nucleic acid codes A and T are complements, and C and G are complement with each other. When the complement code of different sequences at the same site, they can bind to each other. The 3' end sequences of one primer should not be able to bind to any site on the self or other primer. In a primer pair, one primer should not be the complement of the other one. In order to know the primer annealing condition between itself or the others, it is needed to calculate the number of complements of primer x and primer y. In order to calculate the complement, we give the following definition.

Definition (Anneal matching)

For any two base-nucleic acid code sequences A and B. Let A [1, $|A|$] and B [1, $|B|$] be the sequence arrays of A and B, where $|A|$ and $|B|$ are the lengths of A and B, and $|A| \geq |B|$ The anneal matching is to find all S, $-|B| < S < |A|$, such that $A[S+1, S+|B|]$: has the maximum number of complements with B[1, $|B|$] and is denoted as $A_m(A,B)$.

When checking the self-annealing, x equals y, such as $P_f P_f$ or $P_r P_r$. When checking the pair annealing, x and y are different, such as $P_f P_r$ or $P_r P_f$.

Besides, all aligned locations have to be considered and a maximum annealing grade have to be measured. If the 3′ termini have the two following situations, then annealing grades additional annealing penalty (denoted as P_{Anl}) must be added, and it is

$$P_{Anl} = \begin{cases} A_m(x,y) \times 10 & \text{if annealing occurred at the first four nucleotides} \\ A_m(x,y) \times 4 & \text{if annealing occurred from the fifth to the tenth nucleotides} \end{cases}$$

Specificity

The specificity is to check the primer just annealing at the right position on DNA sequence. Mis-priming may be led the PCR experiment to reproduce the incorrect products, such as clamping several fragments in whole genome. Although the primer satisfies the other primer design constraints, if the primer doesn't satisfy the specificity, then this primer might lead to mis-priming. For a specific head site of primer p_f (or p_r), its specificity degree is denoted as $P_{Uni}(p_f)$ (or $P_{Uni}(p_r)$).

$$P_{Uni}(p_f) = A_m(DNA, p_f)$$

$$P_{Uni}(p_r) = A_m(DNA, p_r)$$

Chromosome

In order to conveniently record and compute chromosome's altering situations in "survival of the fittest of natural selection", the start position and lengths of primer are encoded in chromosome. Hence, chromosomes use their head start location and nine different lengths (18~26) as its genes. These genes are useful in the following related evolving computation. After testing the primer with lower annealing grade, a pool of the forward and reverses primers is built. Then chromosomes are randomly chosen from forward pool which is denoted as P_{f_pool} and reverse primers pool which is denoted as P_{r_pool} and stored in parent pool which is denoted as P_pool. The P_{f_pool} is defined as

$$P_{f\_pool} = \{(p_f^s , \alpha_{f1} , \alpha_{f2} , ..., \alpha_{f9})_i \mid (\alpha_{f1} , \alpha_{f2} , \ldots , \alpha_{f9}) \text{ is a permutation of (18, 19, …, 26), and } i = 1, 2, \ldots, \text{population size} \}$$

where the p_f^s is the start point of the forward primer, α_{f1} , α_{f2} , … , and α_{f9} are the lengths of the forward primers.

The P_{r_pool} is defined as

$$P_{r\_pool} = \{(p_r^s , \beta_{r1} , \beta_{r2} , \ldots , \beta_{r9} \}_j \mid (\beta_{r1} , \beta_{r2} , \ldots , \beta_{r9}) \text{is a permutation of (18, 19, …, 26), and } j = 1, 2, \ldots, \text{population size})$$

where the p_r^s is the start point of the reverse primer, β_{r1} , β_{r2} , … , and β_{r9} are the lengths of the reverse primers.

The parent pool is defined as follows:

$$P\_pool = P_{f\_pool} \cup P_{r\_pool} .$$

2.2 The Proposed DOA-based Subsystem

The flowchart of DOA-based genetic algorithm is illustrated in Figure 2. The upper part and the bottom part show the evolution of forward and reverse primers, re-

spectively. In Figure 2, the initialization step is needed to select the individuals from DNA sequence and to build the first generation. Then, the selection step is to create the primer pairs and to prepare the intermediate data (named mating pair) for the DOA-based crossover afterward. In the crossover step, the better chromosome is decided by Orthogonal Arrays. In the mutation step, the chromosome is randomly chosen to mutate. In Figure 2, the mutated chromosome is labeled by using different color.

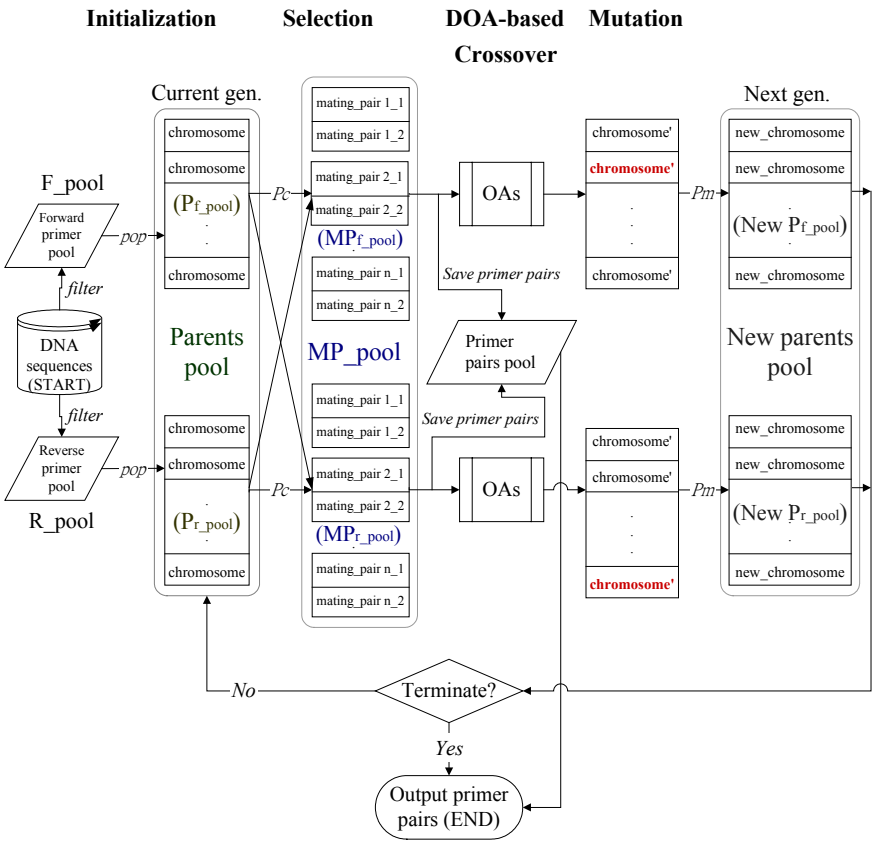

Figure 2. Flowchart of the proposed algorithm

In evolution, it is needed to know the quality of individual. Hence, the chromosome data should be appraisable. After populations are chosen enough amounts from the sample space, it is needed to compute chromosome's annealing grade and filter out the possible head sites of primers. Then, it is needed to evaluate the fitness of chromosomes and to build the population of the next generation for the subsequent evolving computation. The fitness indicates the degree of excellence of a primer. So a chromosome is mutated with the following rules. If the self fitness of a head site in chromosome is high, then mutate a new head site. If the length of the primer causes higher self fitness, then mutate a new length. Finally, the restriction site is optionally

specified by a user. If the user has indicated the type of restriction sites, then primer pairs are tested before output.

Fitness Evaluation

Several factors should be considered for fitness, such as complement avoidance, specificity, regular normal G/C content and a tolerable temperature difference. Hence, the usability of any primer pair found in a given DNA sequence depends on settings of the fitness function. The fitness is computed in two phases. At the initial step, it just calculates the fitness of "the same primer", called the self fitness and denoted as $F_S(p_f)$ or $F_S(p_r)$. $F_S(p_f)$ and $F_S(p_r)$ only compute the annealing grade, melting temperature and GC content of single primer, and are written as

$$F_S(p_f) = A_{ff} + P_T + P_{GC},$$

$$F_S(p_r) = A_{rr} + P_T + P_{GC},$$

where P_T and P_{GC} are the penalty of melting temperature and the penalty of GC percentage, respectively.

$$P_T = \begin{cases} \text{high penalty}, & \text{if } T_f \text{ (or } T_r) > 56, \\ \text{no penalty}, & \text{else if } T_f \text{ (or } T_r) \le 56. \end{cases}$$

$$P_{GC} = \begin{cases} \text{no penalty}, & \text{if } 40 \le GC_f \text{ (or } GC_r) \le 60, \\ \text{relative higher penalty}, & \text{otherwise.} \end{cases}$$

To proceed "not the same survival primer" in the DOA-based crossover step, it needs further to calculate the fitness of different primer called the pair fitness, which is denoted as $F_P(p_f, p_r)$. Pair fitness depends on the self fitness of these two primers, anneal grade of primer pair, and temperature difference of primer pair.

$$F_P(p_f, p_r) = F_S(p_f) + F_S(p_r) + A_{fr} + P_{Tdiff},$$

$$F_P(p_r, p_f) = F_S(p_r) + F_S(p_f) + A_{rf} + P_{Tdiff},$$

where P_{Tdiff} is the penalty of temperature difference (T_{diff}) of p_f and p_r.

$$P_{Tdiff} = \begin{cases} \text{no penalty}, & \text{if } T_{diff} < 5, \\ \text{high penalty}, & \text{otherwise.} \end{cases}$$

If two primers have low annealing grade to each other (called pair annealing) and each primer has high anneal grade to itself (called self end annealing), then these two primers are not the good choices. However, if these two primers have low self annealing to themselves and high pair annealing to each other, then they are not a good primer pairs either.

Application of Orthogonal Arrays

In DOA-based crossover, the "pick out" method is used to select the better genes in chromosome. Therefore, the weaker gene is eliminated through the competitions in the orthogonal arrays estimation. Table 1 is the application of the orthogonal arrays $L_{16}(2^{15})$. The calculation of related parameters is described as follows.

The selection of genes first needs to calculate the whole representation (denoted as F_k , k = 1, 2, …, 16) of all genes of a chromosome and then to estimate the main effect of each competitive gene pair. The value of level j of factor i , $L_{i,j}$, shows the affected part in every representation (F_k) of the *i'th* gene, i = 1, 2, ……, 9, j = 1, 2. Hence, the main effect depends on $L_{i,j}$ to select the operating operand. If $L_{i,j} = 1$, operand is $F_S(p_f)$, else $L_{i,j} = 2$, operand is $F_S(p_r)$. For each experiment k, it needs to know the representation (F_k) of a chromosome. F_k is the average of the sum of the square of $F_P(p_f, p_r)$ of row k.

$$F_k = \left(\sum_{i=1}^{9} F_P(p_f, p_r)^2 \right) / 9$$

$S1_i$ and $S2_i$ are main effects of factor i with level j, which represent the contributions of gene i of parent 1 and parent 2, individually. For each main effect (S1 and S2) of each column i is calculated as follows.

$$S1_i = \mathbf{F} * \sum_{i=1}^{16} (F_i^2 \text{ where factor } i \text{ is level 1 of column } i)$$

$$S2_i = \mathbf{F} * \sum_{i=1}^{16} (F_i^2 \text{ where factor } i \text{ is level 2 of column } i)$$

where **F** is $F_S(p_f)$ when proceeding forward primers and **F** is $F_S(p_r)$ when proceeding reverse primers.

Main effect difference (MED) row is the rule that is applied to select good genes from parent 1 or parent 2 and leaves them to next generation. For each factor i, MED_i is equal to $S1_i - S2_i$,

$$MED_i = S1_i - S2_i$$

Depend on MED_i value, the length with stronger fit replaces the length with weaker fit in a single tournament of each column. Here, the Rank is the increasing score which arranges absolute value of MED_i (denoted as $ABS(MED_i)$) from the smallest to the largest.

Let C1 and C2 be children (new chromosome) survived to next generation. Which genes in two parent chromosomes are left is decided by the following rules. When MED_i is less than or equal to 0, children are chosen from parent 1. If MED_i is larger than 0, children are chosen from parent 2. But there is one exception in order to make C1 and C2 difference, we select the *i'th* gene of C1 and C2 from the different parents,

Table 1. Application of the orthogonal arrays $L_{16}(2^{15})$

Factors / $L_{i,j}$ / exp. no.	1 2 3 4 5 6 7 8 9	F_k
1	1 1 1 1 1 1 1 1 1	F_1
2	1 1 1 1 1 1 1 2 2	F_2
3	1 1 1 2 2 2 2 1 1	F_3
4	1 1 1 2 2 2 2 2 2	F_4
5	1 2 2 1 1 2 2 1 1	F_5
6	1 2 2 1 1 2 2 2 2	F_6
7	1 2 2 2 2 1 1 1 1	F_7
8	1 2 2 2 2 1 1 2 2	F_8
9	2 1 2 1 2 1 2 1 2	F_9
10	2 1 2 1 2 1 2 2 1	F_{10}
11	2 1 2 2 1 2 1 1 2	F_{11}
12	2 1 2 2 1 2 1 2 1	F_{12}
13	2 2 1 1 2 2 1 1 2	F_{13}
14	2 2 1 1 2 2 1 2 1	F_{14}
15	2 2 1 2 1 1 2 1 2	F_{15}
16	2 2 1 2 1 1 2 2 1	F_{16}

where *i'th* gene has the minimum (ABS(MED_i)). That is, if the *i'th* gene of C1 is chosen from parent 2, then the *i'th* gene of C2 is chosen from parent 1.

Mating Pairs and Primer Pair Pool

Mating pool (denoted as MP_pool) is used to store mating pairs that mate from the forward chromosome pool and reverse chromosome pool separately. In order to produce a mating pair for DOA-based crossover phase of the proposed algrithm, we choose two individuals, one is forward chromosome selected from P_{f_pool} and the other is reverse chromosome selected from P_{r_pool}, to mate each other. Crossover phase is to evaluate which two selected primers are better. The forward and the reverse mating pair are processed respectively. The following paragraphs defines the format of a mating pair of MP_{f_pool} and MP_{r_pool} .

A forward mating pair MP_{f_pool} is

$$MP_{f\_pool} = (p_f , p_r , \alpha_{f1} , \beta_{r1} , \alpha_{f2} , \beta_{r2} , \ldots\ldots , \alpha_{f9} , \beta_{r9})$$

and a reverses mating pair MP_{r_pool} is

$$MP_{r\_pool} = ((p_r , p_f , \beta_{r1}', \alpha_{f1}', \beta_{r2}', \alpha_{f2}', \ldots\ldots , \beta_{r9}', \alpha_{f9}').$$

MP_pool are united of MP_{f_pool} and MP_{r_pool} .Thus,

$$MP\_pool = MP_{f\_pool} \cup MP_{r\_pool} .$$

At the same time, the primer pair pool (denoted as pp_pool) stores the sorted coupling data that are derived from the mating process. Randomly matching two mating pairs could produce nine primer pairs which are kept in the primer pair pool. All

generated pairs data are sorted by fitness of {(p_{f1} , α_{f1} , p_{r1} , β_{r1}), (p_{f2} , α_{f2} , p_{r2} , β_{r2}), … , (p_{f9} , α_{f9} , p_{r9} , β_{r9})} or { (p_{r1} , β_{r1}', p_{f1} , α_{f1}'), (p_{r2} , β_{r2}', p_{f2} , α_{f2}'), … , (p_{r9} , β_{r9}', p_{f9} , α_{f9}')}. Hence, the final data in the primer pair pool are the sorted results with a decent order from the excellent to imperfection.

3. Experimental Results

Homo sapiens CDK2-associated protein 1 (CKD2AP1) coding DNAs (CDs) (GenBank Acc#NM_004642; 523..870) and Homo sapiens proliferating cell nuclear antigen (PCNA), (GenBank Acc#NM_002592) are used as our targets for PCR amplification verified experiment and subjected to primer designs in our experiments.

The simulation was run on a PC with AMD K6-350MHz CPU, 128MB RAM and OS with Windows 2000 platform. The program was written in Borland C++ Builder v5.0. In the evolving processes, the crossover rate is 80% and the mutation rate is 10% and the population of the proposed algorithm is 50.

In Table 2 and Table 3 each lists one primer pair that is generated by the proposed algorithm and used in the first experiment and the second experiment. Due to the random selection of initial population of GA, each trial may obtain different set of primer pairs that satisfy the design constraints. In the following we evaluate the convergence performance. The parameter settings are given as follows. The population of the proposed algorithm is 50 and GA is running with 200 and 150 individuals. Ten trials were run for each experiment and each experiment was recorded from the first to the tenth generations. The average objective fitness evaluations (with small-the-best characteristic) are shown in Figure 3. It shows that the proposed algorithm has a faster convergence than GA, but it needs to spend some extra computing cost in OAs than GA.

Table 2. The primer pair selected by the proposed algorithm used in the first experiment.

Primer pairs (the loci of 3′-termini, length)	Product size	GC(%)	M.T. (℃)	T_{diff} (℃)	Specificity (F & R)
(23,18):3´ CGGTTCAAGCCAAACATT	326	44.4	52	4	yes yes
(314,18):3´ CGAACCGTCTTTGCCTTG		55.6	56		

Table 3. The primer pair selected by the proposed algorithm used in the second experiment.

Primer pairs (the loci of 3′-termini, length)	Product size	GC(%)	M.T. (℃)	T_{diff} (℃)	Specificity (F & R)
(198,18):3´ CCTCCTTTCAGATCGACC	847	55.6	56	2	yes low
(1008,20):3´ CTACTTCTTCCTAGAATCCG		45.0	58		

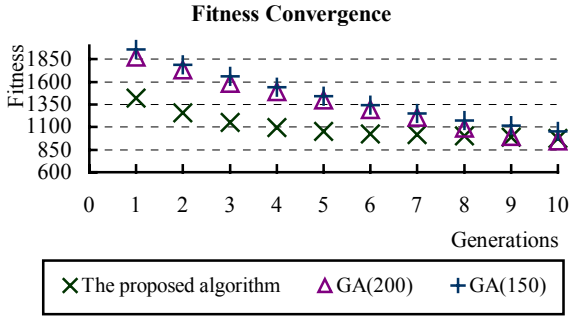

Figure 3 Fitness convergence from the first generation to the tenth generation of the proposed algorithm and GA

For short sequence, such as CDK2AP1, the proposed algorithm can get better performance than GA. But for a long sequence, such as PCNA, it spent too much time on OA calculation, as a result, the total computation time is longer than GA. In GA, the higher sampling with more good individuals makes a better convergence rate. Though the populations used by the proposed algorithm are less than GA, and it still performs better in fitness.

4. Conclusions

In this paper, we have proposed a DOA based genetic algorithm to search the primers with the best fit to the PCR constraints. It is evolved from the traditional genetic algorithm by the integration of the Taguchi method. Though it needs to spend some extra computation in orthogonal arrays, the systematic evolving reduces the time-consuming intensive computation on specificity checking and unnecessary trial-and-error random testing. As a result, it converges to the required result faster. Based on the design properties such as unique, GC content, melting temperature, temperature difference and annealing, the proposed algorithm can derive the solutions more efficiently.

References

1. Sambrook, J. and Russell, D. W.: Molecular Cloning, 8.1-8.126, 3rd edn (Cold Spring Harbor Laboratory Press, New York) (2001)
2. Dieffenbach, C. W. and Dvksler, G. S.: PCR primer: a laboratory manual, (CSHL press, Cold Spring Harbor, USA) (1995)
3. Ashlock, D., Wittrock, A., and Wen, T. J.: Training finite state machines to improve PCR primer design, Evolutionary Computation, Proceedings of the 2002 Congress on CEC '02, Vol. 1 (2002) 12-17
4. Xu, D., Li, G., Wu, L., Zhou, J., and Xu, Y.: PRIMEGENS: robust and efficient design of gene-specific probes for microarray analysis, *Bioinformatics*, Vol. 18, (2002) 1432-1437 And also available at http://compbio.ornl.gov/structure/primegens/
5. Cheng, L. L. and Dahiya, R.: MethPrimer: designing primers for methylation PCRs, *Bioinformatics*, Vol. 18 (2002) 1427-1431
6. Rozen, S. and Skaletsky, H. J.: Primer3, Whitehead Institute for Biomedical Research (1998). Code available at http://www-genome.wi.mit.edu/genome_software/other/primer3.html
7. Kämpke, T., Kieninger, M. and Mecklenburg, M.: Efficient primer design algorithms, *Bioinformatics*, Vol 17 (2001) 214-225
8. Wu, J. S., Lee, C. N., Wu, C. C., and Shiue, Y. L. (2003): Primer Design Using Genetic Algorithm, *Bioinformatics*, ID: BIOINF-2003-0395, accept to appear soon.
9. Goldberg, D. E. : Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York (1989)
10. Phadke, M. S. : Quality Engineering Using Robust Design , Prentice Hall, New Jersey (1989)
11. Ross, P. J. : Taguchi techniques for quality engineering : loss function, orthogonal experiments, parameter and tolerance design, McGraw-Hill, New York (1996)
12. Ho, S. Y., Shu, L. S. and Chen, H. M. : Intelligent genetic algorithm with a new intelligent crossover using orthogonal arrays Proceedings of the 1999 Genetic and Evolutionary Computation Conference, (1999) 289-296
13. S. V. Suggs, T. Hirose, T. Miyake, E. H. Kawashima, M. J. Johnson, K. Itakura, and R. b. Wallace: In Brown, D.D. (ed.), ICN-UCLA Symposia on Developmental Biology Using Purified Genes. Academic Press Inc., New York, Vol. 23 (1981), 683-693