DNA Base-Code Generation for Bio-molecular Computing by Using a Multiobjective Approach Based on SPEA2

José M. Chaves-González and Miguel A. Vega-Rodríguez

Univ. Extremadura. Dept. Computers and Communications Technologies,
Escuela Politécnica. Campus Universitario s/n. 10003. Cáceres, Spain
{jm,mavega}@unex.es

Abstract. The design of DNA strands suitable for bio-molecular computing involves several complex constraints which have to be fulfilled to ensure the reliability of operations. Two of the most important properties which have to be controlled to obtain reliable sequences are self-assembly and self-complementary hybridizations. These processes have to be restricted to avoid undesirable interactions which could produce incorrect computations. Our study is focused on six different design criteria that provide reliable and robust DNA sequences. We have tackled the problem as a multiobjective optimization problem in which there is not only an optimal solution, but a Pareto set of solutions. In this paper, we have used the Strength Pareto Evolutionary Algorithm 2 (SPEA2) to generate reliable DNA sequences for three different real datasets used in bio-molecular computation. Results indicate that our approach obtains satisfactory DNA libraries that are more reliable than other results previously published in the literature.

Keywords: DNA Sequence Design, Multiobjective Optimization, SPEA2.

1 Introduction

Deoxyribonucleic acid (DNA) computing refers to a computational model proposed by Adleman in 1994 [1] which uses DNA molecules as computer storage units and their biological reactions as the operators to perform computations. In this context, the hybridization between DNA sequences is crucial, because undesirable hybridizations usually lead to incorrect computations [2]. Thus, the design of reliable sequences which generate specific duplexes while avoiding other undesirable reactions involves several conflicting design criteria which cannot be managed by traditional optimization techniques [2]. In this case, a design based on multi-objective evolutionary algorithms represents the most suitable alternative. Typical existing approaches for DNA sequence design problem include a wide range of non-exact algorithms, such as evolutionary algorithms, dynamic programming, and heuristic methods [2]. However, a design based on multi-objective evolutionary algorithms (MOEAs) represents the most appropriate design alternative [3] because MOEAs take into account several conflicting objectives simultaneously without the artificial adjustments which are included in classical mono-objective optimization methods.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2013, Part I, LNCS 8111, pp. 227–234, 2013.

In this paper, we consider six different conflicting criteria, two of them taken as restrictions and the other four managed as objectives, to generate reliable DNA sequences suitable for DNA computing by using the multiobjective standard: Strength Pareto Evolutionary Algorithm 2 (SPEA2) [4]. In addition, our results are validated by using other works published in the literature. As will be discussed, our MOEA generates very promising DNA sequences that surpass the results obtained with other relevant approaches previously published.

The rest of the paper is organized as follows: Section 2 describes the basic background on the problem and the multiobjective formulation followed. The SPEA2 adaptation developed is explained in Section 3. Section 4 is devoted to present and to analyze the results, as well as comparing our approach with other methods published in the literature. Finally, Section 5 summarizes the conclusions of the paper.

2 DNA Base-Code Generation for Reliable Computation

In recent years, there has been an increase in the technologies which are based on DNA molecules, such as nanotechnology, DNA sequencing or DNA computing [2]. In all those technologies, the design of reliable DNA libraries is a crucial task. One of the most important processes for DNA molecules is the Watson-Crick pairing [5], or the hybridization between a sequence and its basepairing complement. The problem here is to control undesirable hybridizations, because they can produce errors in the biological reactions, so they have to be avoided when sequences are designed.

DNA sequence design problem consists of designing sets of reliable sequences which form stable duplexes while avoiding undesirable interactions. Every sequence design criteria should contribute to improving reliability, because this property is a very important requirement for any system based on DNA sequences. There are several biological criteria that can be considered to achieve this purpose. According to their biological meaning, design criteria can be classified into four groups [6]. First, properties that avoid inconvenient reactions; second, criteria that control the generation of secondary structures; third, properties that control the biochemical characteristics of DNA sequences; and finally, criteria that restrict the sequences composition. From the first group, we have taken the *similarity* and the *h-measure* objectives. Similarity calculates the inverse Hamming distance between two sequences, while h-measure tests the possibility of unintended DNA basepairing. Both criteria are checked by considering shifts in sequences under study. Regarding to the second category, secondary structures formation, we have included the objectives: *hairpin*, which indicates the probability that the sequence under study can generate secondary structures and *continuity*, which counts the repetitions of identical bases. This is important because if one base is repeated several times, an unusual secondary structure could be formed. The third category refers to the biochemical characteristics of the sequences. It is important to control that every sequence have similar chemical features. We have included the following two restrictions from this category: *melting temperature*, which is the temperature at which half of the DNA strands are in the double-helical state and half are in a random coil state (dissociated), and *GC ratio*, which indicates the percentage of cytosine (C) and guanine (G) in a sequence.

2.1 Multiobjective Formulation

DNA base-code generation can be naturally formulated as a multiobjective optimization problem in which the objectives and constraints are the design criteria that every sequence has to satisfy to ensure reliability. We have considered six different design criteria to cover a wide range of aspects which contribute to reliability [6]. Four are considered as objectives: *Similarity* and *h-measure* avoid inconvenient reactions between sequences. On the other hand, *continuity* and *hairpin* control the generation of secondary structures. Finally, *melting temperature* and *GC ratio* are considered as constraints for the problem and they assure that DNA sequences are in the similar bio-chemical ranges. The four objectives have to be minimized, so the problem can be described as follows.

$$\begin{aligned} &\text{Minimize } F(X) = (f_1(X), f_2(X), f_3(X), f_4(X)) \\ &\qquad\qquad \text{subject to } c_1(X) \text{ and } c_2(X) \end{aligned} \tag{1}$$

where $f_i(X)$ are the objectives previously mentioned (similarity, h-measure, continuity and hairpin), $c_i(X)$ are the melting temperature and the GC ratio constraints, and X is the set of DNA sequences under study.

A formal definition of each design criterion included in equation (1) is given below.

1) Similarity: This objective computes the similarity in the same direction of two given sequences to keep each sequence as unique as possible, including position shifts. For a more complete comparison, the target sequence is extended by adding its own sequence to the 3'-end with gaps. Moreover, we consider continuous (s_{cont}) and discontinuous (s_{disc}) similarities. The mathematical definition for this measure is described in (2).

$$\begin{aligned} f_{similarity}(x, y) = Max_{g,i}(&s_{disc}(x, shift(y, g, i)) + \\ &+ s_{cont}(x, shift(y, g, i))) \end{aligned} \tag{2}$$

where x and y are parallel sequences and *shift* indicates a shift of sequence y by i bases and g gaps. s_{disc} is a real value between 0 and 1, and s_{cont} is an integer between 1 and the length of the sequences. Finally, we have to indicate that similarities have to surpass a threshold that has to be established by experimentation to be considered.

2) H-measure: This objective is similar to similarity, but instead of considering sequences in parallel, they are managed as complementary. H-measure prevents cross hybridization between DNA strands. We consider elongated sequences with gaps for a more reliable measure. The mathematical definition is given in (3).

$$\begin{aligned} f_{h\_measure}(x, y) = Max_{g,i}(&h_{disc}(x, shift(y, g, i)) + \\ &+ h_{cont}(x, shift(y, g, i))) \end{aligned} \tag{3}$$

where x and y are anti-parallel sequences and *shift* indicates a shift as in the similarity case. h_{disc}, h_{cont} and the threshold have also analogous values to the similarity measure.

3) Continuity: This measure calculates the degree of successive occurrences of the same base in a sequence. The measure prohibits consecutive runs of the same base over a given threshold. For example, if the threshold is 3, in the sequence AGGCAATAAAACGAAATGGGC, only the third subsequence of adenines (A) violates the continuity. The mathematical definition for this measure is given in (4).

$$f_{continuity}(x) = \sum_{i=1}^{\max} \sum_{a \in \{A,C,G,T\}} T(c_a(x,i),t)^2 \tag{4}$$

where x is the sequence under study, *max* is the difference between the length of the sequence and the threshold (T), $c_a(x,i)$ is equal to ε if ∃ ε s.t. $x_i \neq a$, x_i+1=a for $1 \leq j \leq \varepsilon$, $x_{i+\varepsilon+1} \neq a$, and 0 otherwise.

4) Hairpin: This restriction represents the probability of secondary structures creation. For simplicity, it is calculated through the Hamming distance by considering the length of hairpin loop and the number of hybridized pairs. It is assumed that a hairpin has at least $R_{\min}$ bases as a loop and a minimum of $P_{\min}$ base pairs as a stem. It is also considered the penalty for formation of hairpins of various sizes at every position in the sequence. In (5) are considered hairpins with r-base loop and p-base pairs stem to be formed at position i in the sequence x, if more than half bases in the subsequence $x_{i-p}\ldots x_i$ hybridize to the subsequence $x_{i+r}\ldots x_{i+r+p}$. The number of matches in these subsequences is defined as the penalty for this hairpin.

$$f_{hairpin}(x) = \sum_{p=P\min}^{\max I} \sum_{r=R\min}^{\max R} \sum_{i=1}^{\max I} T(hp(x,p,r,i), \frac{pinlen(p,r,i)}{2})$$

$$hp(x,p,r,i) = \sum_{j=1}^{pinlen(p,r,i)} bp(x_{p+i+j}, x_{p+i+r+j}) \tag{5}$$

where the function $pinlen\ (p,r,i) = min(p+i,\ l-r-i-p)$ and denotes the maximum number of possible basepairs when a hairpin is formed at center $p+i+r/2$.

5) GC content: This criterion indicates the percentage of bases C and G in the sequence. This is important because the GC content affects to the chemical properties of DNA sequences. For example, the GC% of the DNA sequence ACGTT is 40.

6) Melting temperature, T_m: This measure predicts DNA thermal denaturation, which is a key factor for DNA computing. Both sequence and base composition are important determinants of DNA duplex stability. There are many ways to calculate this relevant feature, but we use the nearest neighbour (NN) model [7]. The mathematical description for this measure is provided in (6).

$$Tm(x) = \Delta H^{\circ}(x) / \Delta S^{\circ}(x) + R \ln(|C_T|/4) \tag{6}$$

where x is the DNA sequence studied, R is a gas constant and $|C_T|$ is the total sequence concentration. ΔH^o and ΔS^o refer to predicted enthalpies and entropies. Those values were taken from [7].

3 Multiobjective Approach

We have generated reliable DNA sequences suitable for molecular computing by using an adapted version of the Strength Pareto Evolutionary Algorithm 2 (SPEA2), which is a population-based algorithm originally created by Zitzler et al. in [4]. The pseudocode of the proposed MOEA is shown in Algorithm 1.

Algorithm 1. Pseudocode of SPEA2

```
1: P ⇐ generateRandomPopulation (PSize)
2: A ⇐ ∅   //Archive (ArchiveSize)
3: while not stop condition satisfied do
4:    FitnessAssignment (P, A)
5:    EnviromentalSelection (A, P)   //Truncate A if necessary
6:    for i=1 to PSize do
7:       ind1, ind2 ⇐ tournamentSelection (A)   // ind1≠ind2
8:       P_i ⇐ recombination (ind1, ind2, Pcr)
9:       P_i ⇐ mutation (P_i, Pm)
10:   end for
11: end while
```

SPEA2 uses a regular population, P, of $PSize$ individuals, and an archive (external set, A). The process starts with the random generation of the initial population and the initialization of the archive set (lines 1, 2). Each individual in the population is a valid DNA library which represents the solution for the specific problem instance which is being considered. A solution is composed of a set of n sequences. Each DNA strand is composed of m bases each (sequence length). The number of sequences and the number of bases per sequence depend on the problem instance. The data structure contains the DNA strands used by the genetic operators of our MOEA along with the values for each biochemical design criteria.

In each iteration, all non-dominated solutions (the best solutions) of both, population and archive, are copied into a new population, truncating it when the size of the new population exceeds $PSize$ solutions (line 5). Previously, a fitness value that is the addition of its strength raw fitness and a density estimation is assigned to each individual in P and in A (line 4). The raw fitness is based on the concept of Pareto dominance. The raw fitness of a solution, $R(i)$, is determined by the strengths of its dominators in both archive and population. It is a measure to be minimized, so $R(i) = 0$ corresponds to a nondominated individual, while a high $R(i)$ value means that solution i is dominated by many individuals. A particular solution is of more quality than another if it is dominated by fewer solutions. A solution dominates another if it is better, at least, in one of the objectives and it is not worse in any of the others. In case of individuals having identical raw fitness, it is used a density estimation technique which is based on the distance (in the objective space) to the kth nearest solutions. SPEA2 uses binary tournament selection, crossover at two levels (at individual and sequence levels) and random mutation (lines 7-9) for improving the population in each generation.

4 Experimental Evaluation and Results

The algorithm developed has been adjusted to obtain optimal results by performing a complete set of experiments. The value of each parameter (population size *PSize*, archive size *ArchiveSize*, crossover probability *Pcr*, mutation probability *Pm* and parent selection strategy) has been fixed after executing 30 independent runs to ensure statistical significance. Table 1 shows the algorithm configuration. All experiments were performed by using a 2.3GHz Intel PC with 1GB RAM. The algorithm was compiled using gcc 4.4.5 compiler. For comparison with other authors [6], we have used the same population size and stop condition for the algorithm (3000 individuals and 200 iterations respectively).

Table 1. Algorithm configuration

SPEA2 configuration	
Archive size (*ArchiveSize*)	*PSize*/2
Crossover probability (*Pcr*)	0.3
Mutation probability (*Pm*)	0.5
Parent selection strategy	Binary tournament

We have used three different-sized sets of DNA sequences proposed by different authors [8], [9], and [10] which have been used for reliable DNA computing. This fact ensures that our algorithm works with several types of instances which have been tested to be used for bio-molecular computing. Moreover, we compare our results with sequences generated by Shin *et al.* [6], which use a multiobjective approach with the same data sets. We examine the quality of each design criterion for a set of sequences taken from the median Pareto front generated by our SPEA2. The comparison is not performed in terms of any multiobjective metrics, such as hypervolume, because unfortunately no studies have taken multiobjective indicators so far. Biochemical constraints and parametrical adjustments for the design criteria used in our study were established as explained in the literature [6]. Thus, for H-measure (H) and similarity (S), we set lower limits for the continuous case equal to six bases and 17% for the discontinuous case. For continuity (C), the threshold value was 2. Hairpin (P) formation requires at least six basepairings and a six base loop. The melting temperature (T_m) was calculated with 1 *M* salt concentration and 10*nM* DNA concentration. Furthermore, the T_m and the GC ratio are considered constraints whose values were taken from the literature. For the results in [8] and in [9], sequences have the GC ratio restricted to 50% and the melting temperature between 46 and 53 degrees. On the other hand, for the work in [10], the range of the GC ratio is between 40% and 50% and the melting temperature between 31 and 39 degrees. Shin *et al.* [6] uses the same restrictions. Comparative results are given in Fig. 1 for the three data sets under study. Furthermore, in Table 2, we show the comparison of sequences generated in [8], sequences generated in [6] and an example taken from the median Pareto front of the sequences generated by our approach. Due to the limit in the number of pages, we cannot show a similar table for the other two instances (but Fig. 1 summarizes these comparisons).

Table 2. Comparison of the sequences in [8], [6] and sequences obtained by our proposal

Seq. (5'→ 3')	C	P	H	S	Tm	GC
Sequences obtained in [8]						
ATAGAGTGGATAGTTCTGGG	9	3	55	64	52.6522	45
CATTGGCGGCGCGTAGGCTT	0	0	69	51	69.2009	65
CTTGTGACCGCTTCTGGGGA	16	0	60	63	60.8563	60
GAAAAAGGACCAAAAGAGAG	41	0	58	45	52.7111	40
GATGGTGCTTAGAGAAGTGG	0	0	58	54	55.3056	50
TGTATCTCGTTTTAACATCC	16	4	61	50	48.4451	35
TTGTAAGCCTACTGCGTGAC	0	3	75	55	56.7055	50
Sequences obtained in [6]						
CTCTTCATCCACCTCTTCTC	0	0	43	58	46.6803	50
CTCTCATCTCTCCGTTCTTC	0	0	37	58	46.9393	50
TATCCTGTGGTGTCCTTCCT	0	0	45	57	49.1066	50
ATTCTGTTCCGTTGCGTGTC	0	0	52	56	51.1380	50
TCTCTTACGTTGGTTGGCTG	0	0	51	53	49.9252	50
GTATTCCAAGCGTCCGTGTT	0	0	55	49	50.7224	50
AAACCTCCACCAACACACCA	9	0	55	43	51.4735	50
Sequences obtained with SPEA2						
CAACAGATGAGTAACTCCCC	0	0	57	44	47.214	50
TTCCTTGTTCCTGCTTCCTC	0	0	41	57	49.576	50
CTTCTCTCCTTCTCTCCTTG	0	0	37	61	46.266	50
ATGGTTAGTGTAGGAGTGGG	0	0	58	42	48.126	50
TCTCGTCGTAGTAGTCTTCG	0	0	52	57	47.901	50
TTCAACCTGCTGTCTTCCCT	0	0	45	55	51.112	50
TTCTTGTGTTCTGCACTCCC	0	0	48	58	50.125	50

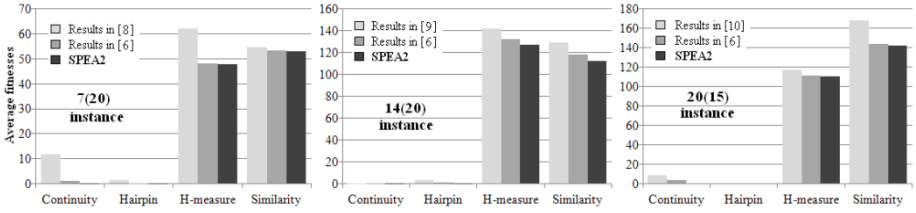

Fig. 1. Average fitness comparison between our approach (SPEA2) and other relevant works for the three instances tackled. *Y* axis indicates the average values of each fitness objective.

In [8], authors proposed a genetic algorithm to design good sequences for Adleman's graph. Shin *et al*., in [6], proposed NACST/Seq algorithm to improve those sequences. Results given in Table 1 and Fig. 1 show that our approach obtains sequences with lower similarity (S) and h-measure (H) values, while obtaining minimal values for hairpin (P) and continuity (C). This means that sequences obtained by our SPEA2 have higher probability to hybridize with its correct complementary sequences. Besides, secondary structures are virtually prohibited because values for hairpin and continuity are reduced to zero. Moreover, ranges for melting temperature and GC ratio are also better, which means more stable sequences. On the other hand, results obtained in [9] and in [10] generated sequences to solve other problems (travelling salesman problem and knight movement problem) by using other methods. Fig. 1 shows that for those instances our approach also obtains sequences with lower similarities and h-measures, while obtaining minimal continuities and hairpins.

This means that sequences obtained by SPEA2 are more reliable. Secondary structures are virtually prohibited because hairpin and continuity are reduced to zero. Moreover, ranges for GC ratio and T_m are also better, which means more stable sequences.

5 Conclusions and Future Work

In this paper, we present SPEA2 for the design of DNA sequences that can be applied to reliable molecular computing. SPEA2 can obtain high quality sets of sequences which simultaneously minimize similarity, h-measure, hairpin and continuity while controlling T_m and GC content. We have used three different real-world instances proposed by different authors to ensure the effectiveness of our approach. These data sets include different number of sequences, number of bases and bio-chemical restrictions, and all of them have been used for reliable computation. After our study, we can conclude that our version of SPEA2 can generate better sequences than other approaches previously published in the literature. As future work, we are studying other multiobjective approaches and restrictions which can contribute to generate more reliable sequences for DNA computing.

Acknowledgments. This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project).

References

1. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science 266, 1021–1024 (1994)
2. Brenneman, A., Condon, A.: Strand design for biomolecular computation. Theoretical Computation Science 287, 39–58 (2002)
3. Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic Algorithms and Evol. Computation. Kluwer (2002)
4. Zitzler, E., et al.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Proceedings of EUROGEN 2002, pp. 95–100 (2002)
5. Garzon, M.H., Deaton, R.J.: Biomolecular computing and programming. IEEE Trans. Evol. Computation 3, 236–250 (1999)
6. Shin, S.-Y., et al.: Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing. IEEE Trans. Evolutionary Computation 9(2), 143–158 (2005)
7. Santa Lucia Jr., J.: A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. Proc. Nat. Acad. Sci. U.S.A. 95, 1460–1465 (1998)
8. Deaton, R., et al.: Good encodings for DNA-based solutions to combinatorial problems. In: Proceedings of 2nd Annual Meeting on DNA Based Computers, pp. 247–258 (1996)
9. Tanaka, F., et al.: Toward a general-purpose sequence design system in DNA computing. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 73–78 (2002)
10. Faulhammer, D., et al.: Molecular computation: RNA solutions to chess problems. Proceedings of the National Academy of Sciences 97, 1385–1389 (2000)