A Multiobjective Evolutionary Fuzzy System for Promoter Discovery in *E. coli*

Rocío C. Romero Zaliz
Department of Computer Science
FCEyN - University of Buenos Aires
Buenos Aires, Argentina
E-mail: rromero@dc.uba.ar

Oscar Cordón, Cristina Rubio, Igor Zwir
Department of Computer Science and A.I.
E.T.S.I. Informática. University of Granada
18071-Granada. Spain
E-mail: {ocordon,crubio,zwir}@decsai.ugr.es

***Abstract*— In this contribution, the biological problem of extracting promoters (composed of two nucleotide sequences, TTGACA and TATAAT, separated by among 15 and 22 pairs of bases) from *E. coli* DNA sequences is tackled. Classical approaches for this problem, based on considering probabilistic models of the promoter motifs, fail at performing accurate predictions due to the difficulty of properly integrating the modeled sub-motifs because of the uncertainty existing in the distance between them. However, our methodology solves this problem by applying a multiobjective evolutionary algorithm to extract the promoters, thus being able to discover promoters where the sub-motifs are located at different distances. As we consider the sub-motifs to be modeled by fuzzy logic tools, and evolutionary algorithms are also used to tune these fuzzy models, the resulting technique becomes a multiobjective evolutionary fuzzy system. Some experiments to extract previously known and unknown promoters from *E. coli* DNA sequences are reported to show its good performance when compared to classical techniques. This method is available for public use in http://gps-tools.wustl.edu.**

I. Introduction

Advances in molecular biology and new computational techniques are enabling us to systematically investigate the complex molecular process underlying biological systems. The continued development of large, sophisticated repositories of knowledge and information has facilitated the accessibility to vast amounts of biological data (e.g., cis-regulatory features, metabolic pathways, regulatory networks). However, paradoxically, the usefulness of these databases is partially limited by the inability to search them in terms that match the needs and experience of their users. For example, researchers usually get lost when trying to identify the distinguishing features that describe their target systems in highly interconnected databases. Moreover, available databases always provide insights of previously described biological systems, but conceal a mechanism to make inferences from stored knowledge into new queries and to make predictions about them [1]. Due to these reasons, there is an increasing interest on applying knowledge discovery and intelligent data analysis techniques to this area.

Soft computing [2], [3] is a problem solving methodology of the latter kind that provides a computational framework to address design, analysis and modeling problems in the context of uncertain and imprecise information. Its constituents fuzzy logic, neural networks, probabilistic computing and evolutionary algorithms are considered as complementary and synergistic partners rather than competing methodologies.

In particular, *genetic fuzzy systems* (GFSs) [4], [5] have been showed as promising hybrid techniques in the realm of soft computing, as they combine the ability of fuzzy logic to deal with uncertainty and fuzziness, designing approximate reasoning models, with the learning and adaptation capabilities of evolutionary algorithms (EAs) [6]. More specifically, the combination of multi-objective EAs [7], [8] and fuzzy systems have obtained very good results [9].

In a previous work [10], [11], we made use of the latter good properties of GFSs in order to solve a complex biological problem: the discovering of promoters in prokaryotic DNA sequences, more specifically for *E. coli*, where the promoters are composed of two nucleotide sequences, TTGACA and TATAAT, separated by among 15 and 22 pairs of bases. To do so, our approach, called *Generalized Analysis of Promoters* (GAP) and based on generalized clustering, considered the use of fuzzy logic to model the two sub-motifs (sequences of nucleotides) composing the promoter, as well as the distance between them, all of which are characterized by their uncertainty in nature. The main novelty of our methodology was the use of a multiobjective EA as the tool to extract the existing promoters in the set of DNA sequences, thus being able to discover promoters where the two sub-motifs are located at different distances, a difficult task for classical, probabilistic techniques [12].

Although the designed GFS performed properly in the problem solving, leading to a better performance than classical techniques in the identification of true positive solutions, it also reported however a higher amount of false positive results. Multiple occurrences of promoters in the same regulatory region of one gene can be found (e.g. different promoters can be used for gene activation and repression, or can interact with different regulatory factors from the same regulatory pathway [13], [14]), and unless mutagenesis is performed, each site has the chance to be the place chosen by the RNA polymerase to bind the DNA. Although this approach agree with the biological requirements performed by the experts, computationally, it produces an uncompensated amount of false positive results. To solve this problem, in this contribution we extend our methodology by incorporating a new EA to it with the aim of tuning the parameters of the sub-motif and distance fuzzy models in order to increase the system accuracy. Both the fuzzy model membership functions and, specially, their membership thresholds are adjusted, with the latter being the most important task as these thresholds will finally determine which of the DNA sub-sequences matching to some degree with the promoter fuzzy models are actually considered to belong to the modeled sub-motif.

To do so, the paper is structured as follows. Section II briefly introduces the problem tackled, describing the composition of the promoters that are looked for, as well as the classical approaches considered to extract them. Section III presents the way in which the biological feature (the promoter) is modeled using fuzzy logic by deriving the three fuzzy models for the two sub-motifs and the distance existing between them. Then, the multiobjective EA (based on Scatter Search [15]) to extract the promoters from the DNA sequences using the fuzzy models is introduced in Section IV. The genetic tuning process to adjust the parameters of the sub-motif and distance fuzzy models is then showed in Section V. The experiments developed on *E. coli* sequences are reported in Section VI, comparing the results obtained by our approach, with and without evolutionary tuning, with those from classical techniques. Finally, some concluding remarks and future works are showed.

II. Problem: Discovering Promoters in DNA Sequences

Biological sequences, such as DNA or protein sequences, are a good example of the type of complex objects that may be described in terms of meaningful structural patterns. Availability of tools to discover these structures and to annotate the sequences on the basis of those discoveries would greatly improve the usefulness of these repositories that currently rely on methods developed on the basis of computational efficiency and representation accuracy rather than on terms of structural and functional properties deemed to be important by molecular biologists.

An interesting example of biological sequences are prokaryotic promoters. Many compilations gathered these sequences data and analyzed it [16], [17], [18], revealing the presence of two well conserved sequences or sub-motifs separated by variable distances and a less conserved sequence. The variability of the distance between the sub-motifs and their own fuzziness, in the sense that they present several mismatches, hinder the existence of a clear model of prokaryotic core-promoters.

The most representative promoters in *E. coli* (i.e. σ^{70} subunits) are described by the following conserved patterns (see Figure 1):

1) *TTGACA:* This pattern is an hexanucleotide conserved sequence whose middle nucleotide is located approximately 35 pairs of bases upstream of the transcription start site. The consensus sequence for this pattern is TTGACA and it is often called *-35 region*. Different compilations have different nucleotide distributions for this pattern [18], [17]. For example, in [18] the following nucleotide distribution is showed: $\mathrm{T}_{69}\mathrm{T}_{79}\mathrm{G}_{61}\mathrm{A}_{56}\mathrm{C}_{54}\mathrm{A}_{54}$, where for instance the first T is the most seen nucleotide in the first position of the pattern and is present in 69 % of the cases. However, in [17] the nucleotide distribution is $\mathrm{T}_{78}\mathrm{T}_{82}\mathrm{G}_{68}\mathrm{A}_{58}\mathrm{C}_{52}\mathrm{A}_{54}$.
2) *TATAAT:* This pattern is also an hexanucleotide conserved sequence, whose middle nucleotide is located approximately 10 pairs of bases upstream of the transcription start site. The consensus sequence is TATAAT and it is often called *-10 region*. Again, the nucleotide distribution varies from compilation to compilation. For instance, in [18] the following one is showed: $\mathrm{T}_{77}\mathrm{A}_{76}\mathrm{T}_{60}\mathrm{A}_{61}\mathrm{A}_{56}\mathrm{T}_{82}$, while in [17] the nucleotide distribution is $\mathrm{T}_{82}\mathrm{T}_{89}\mathrm{G}_{52}\mathrm{A}_{59}\mathrm{C}_{49}\mathrm{A}_{89}$.
3) *CAP Signal:* In general, a pyrimidine (C or T) followed by a purine (A or G) compose the CAP Signal. This signal constitutes the transcription start site (TSS) of a gene.
4) *Distance(TTGACA, TATAAT):* The distance between the TTGACA and TATAAT consensus follows a data distribution approximately between 15 and 22 pairs of bases. This distance is critical in holding the two sites at the appropriate distance for the geometry of RNA polymerase [16], [17].

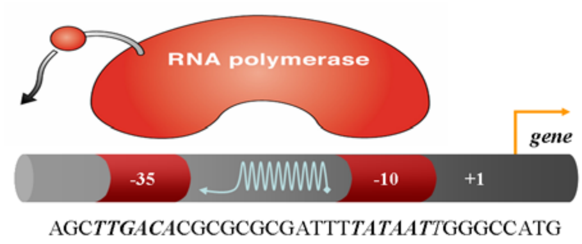


Fig. 1. RNA polymerase binding sites

The identification of the former RNA polymerase or promoters sites becomes crucial to detect gene activation or repression, by the way in which such promoters interact with different regulatory proteins (e.g. overlapping suggest repression and distances of approximately 40 base pairs suggest typical activation). Moreover, combining the promoter sites with other regulatory sites [19] can reveal different types of regulation, harboring RNA polymerase alone, RNA polymerase recruiting other regulatory protein, or cooperative regulations among more than one regulator [20].

Different methods have been used to identify promoters [21], [22], [23], [17], but several failed to perform accurate predictions because of their lack of flexibility, by using crisp instead of fuzzy models for the sub-motifs (e.g., TATAAT or TTGACA [24]), or, even dealing with probabilistic models for the two sub-motifs, restricting distances between them to fixed values (e.g., 17 base pairs [12]). The vagueness of the compound promoter motifs and the uncertainty of identifying which of those predicted sites correspond to a functional promoter can be completely solved only by performing mutagenesis experiments [20].

Thus, more accurate and interpretable predictions would be useful in order to reduce the experiment costs and ease the researchers work. This constitutes the main goal of the current work, and will be put into effect by a multiobjective GFS.

III. Our Promoter Fuzzy Logic-based Description Method

As seen in the previous sections, to address the promoter prediction problem we take advantage of the ability of representing imprecise and incomplete motifs, by considering the fuzzy set-based representations flexibility and interpretability, as well as of the ability of the multiobjective EA to properly combine the two fuzzy model sub-motifs with the distance one in order to be able to discover promoters located at different distances (as we will see in the next section).

Our method represents each promoter, composed of the -10 and -35 regions and the distance that separates them (the CAP signal is left aside due to its very short length, often leading to a great number of false positives in the promoter extraction), as three parametrized fuzzy models M_α^1, M_α^2, and M_α^3, where α represent an approximate model whose membership functions are learned from data distributions [25], [26].

Hence, in order to solve our promoter prediction problem, three different fuzzy models were developed. The first two of them, those corresponding to the -10 and -35 sub-motifs, were implemented by using their nucleotide consensus frequency as discrete fuzzy sets [25]. This way, the fuzzy model associated to the TTGACA pattern, M_α^1, was formulated as:

$$M_\alpha^1 = \mu_{TTGACA}(x) = \mu_1^1(x_1) \cap ... \cap \mu_6^1(x_6) \qquad (1)$$

where the discrete fuzzy set corresponding to each nucleotide composing the sub-motif is obtained from the probability distribution associated to it, i.e., from the probability of appearance of each of the four letters in this position. The intersection operator applied to compute the matching of the sequence to the model is modeled by the min t-norm [26].

	T	T	G	A	C	A
A	3	10	3	**58**	32	**54**
C	9	3	14	13	**52**	5
G	10	5	**68**	10	7	17
T	**78**	**82**	15	20	10	24

(a) TTGACA motif

	T	A	T	A	A	T
A	3	**89**	26	**59**	**49**	3
C	8	3	10	12	21	5
G	7	1	12	15	11	2
T	**82**	7	**52**	14	19	**89**

(b) TATAAT motif

TABLE I

Motif nucleotide distribution from [17]

For example, taking into account the nucleotide distribution showed in Table I, taken from [17], the discrete fuzzy set of the first nucleotide of the sub-motif was defined as $\mu_1^1(x_1) = A/0.03 + T/0.78 + G/0.10 + C/0.09$, and the other fuzzy sets corresponding to positions 2-6 were calculated in a similar way. Notice that the sum sign has not got any mathematical meaning at all but stands for a notation of the different elements belonging to the fuzzy set [25].

The second fuzzy model corresponding to the TATAAT pattern, M_α^2, was described as:

$$M_\alpha^2 = \mu_{TATAAT}(x) = \mu_1^2(x_1) \cap ... \cap \mu_6^2(x_6) \qquad (2)$$

where the discrete fuzzy set associated to the first nucleotide of the sub-motif was defined as $\mu_1^2(x_1) = A/0.03 + T/0.82 + G/0.07 + C/0.08$ as showed in Table IV, and the fuzzy sets corresponding to the remaining positions were calculated in the same way.

We decided to use this method to build fuzzy sets form probability distributions since it is a very simple and economic approach. There are other techniques for building fuzzy sets from probability distributions [25], [26]. Some of them are more complex but may achieve better results, hence they will be studied further in future works.

The third fuzzy model, i.e., the distance between the two previous patterns, was built as a fuzzy set, whose triangular membership function M_α^3 was also learned from the data distributions observed in known promoters [17] (Figure 6(a)).

IV. GAP: A Multiobjective Genetic Fuzzy System for Promoter Discovery

A. Basis of the GAP Approach

Our investigations are oriented toward the automated production of qualitative descriptions of complex DNA objects (e.g., transcription factor or RNA polymerize binding sites). The term "qualitative" is meant to indicate that we intend to identify substructures that match approximately –often measured by some numerical measure of degree of matching– an instantiated version of an idealized model derived from expert knowledge. The notion of "interesting feature" is formally defined by means of a family of parametrized models $M = \{M_\alpha\}$ specified by domain experts [27] who are interested in finding patterns such as epoch descriptors of individual or multiple DNA sequences. These idealized versions of prototypical models are the basis for a characterization of clusters as cohesive sets that is more general than their customary interpretation as "subsets of close points".

In addition, as a generalized clustering method, GAP considers the quality of matching with each promoter model (Q), as well as the size of the promoter extent, by means of the distance between the two sub-motifs, as the *multiple objectives* to be optimized. To do so, we used a Multiobjective Scatter Search (MOSS) EA, which obtains a set of multiple and optimal promoter descriptions for each promoter region. Moreover, the former matching is also considered by MOSS as a *multimodal* problem, since there is more than one solution for each region. GAP, by using MOSS, overcomes other methods used for DNA motif discovery, such as Consensus/Patser based on position weight probabilistic matrices (as we will see in Section VI), and provides the desired trade-off between accurate and interpretable solutions, which becomes particularly desirable for the end users.

The extension of the original *Scatter Search (SS)* EA [15] uses the DNA regions where promoters should be detected as inputs and finds all optimal relationships among promoter and distance models. In order to extend the original SS algorithm to a multiobjective environment, we need to introduce some concepts [7], [8]:

Our multiobjective optimization problem is defined as:

$$\left.\begin{array}{lll} \mathit{Maximize} & Q_m(x, M_\alpha), & m = 1, 2, \ldots, |M|; \\ \mathit{subject\ to} & g_j(x) \geq 0, & j_g = 1, 2, \ldots, J; \\ & h_k(x) = 0, & k = 1, 2, \ldots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \ldots, n. \end{array}\right\}$$

where M_α is a generalized clustering model, $|M|$ corresponds to the number of models and Q_m is the quality of matching with the models M, which constitute the objectives to optimize, J to the number of inequality constraints, K to the number of equality constraints, and finally n is the number of decision variables. The last set of constraints restricts each decision variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound. Specifically, we consider the following instantiations:

- $|M| = 3$. We have three models: M_α^1 and M_α^2 are the models for each of the sub-motifs, TTGACA and TATAAT, respectively, and M_α^3 corresponds to the distance between these two patterns (recall Equations 1 and 2, and Figure 6(a)).
- $|Q| = 3$. We have three objectives consisting of maximizing the degree of matching of the DNA regions to the fuzzy models (fuzzy membership): $Q_1(x, M_\alpha^1)$, $Q_2(x, M_\alpha^2)$ and $Q_3(x, M_\alpha^3)$.
  Therefore, the objective functions Q_m correspond to the membership to the fuzzy models M_α showed in Section III.
- $J = 1$. We have just one constraint g_1: the distance between models can not be less than a and b as showed in Figure 6(a).
- $K = 0$. No equality constraints needed.
- The models can not be located outside the sequence searched, that is, it can not start at negative positions or be greater than the length of the query sequence.
- The optimization procedure, to be presented in the remainder of this section, keeps only valid solutions in each generation.

Definition 1: A solution x is said to dominate a solution y ($x \prec y$), if both following conditions are true: (1) x is not worse than y in all objectives: $f_i(x) \ntriangleright f_i(y)$, for all $i = 1, 2, \ldots, M$; (2) x is strictly better than y in at least one objective: $f_j(x) \triangleleft f_j(y)$, for at least one $i \in \{1, 2, \ldots, M\}$. If x dominates y, it is also customary to write that x is *non-dominated* by y.

B. The MOSS Algorithm

We modified the original SS EA to allow multiple-objective solutions by adding the *non-dominance* criterion to the solution ranking [7]. Thus, non-dominated solutions were added to the set in any order, but dominated solutions were only added if no more non-dominated solutions could be found. In addition to maintaining a good set of non-dominated solutions, and to avoid one of the most common problems of multiobjective algorithms such as multimodality [7], we also kept track of the diversity of the available solutions through all generations by using the same approach as the one used in the original SS algorithm [15]. That is, whenever the subsets of solutions is fully explored, new subsets are generated using a *Diversification Generation Method* (see step 6 in Figure 2) that incorporates individuals in the set taking into account the distance between the solutions already in the set and the new ones to be included. This diversity is achieved by using a suitable distance metric (see Equation 5). Finally, the initial populations were created randomly and unfeasible solutions corresponding to out of distance ranges between the two sub-motifs were checked at each generation. Figure 2 clearly illustrates the MOSS algorithm proposed in GAP.

```
 1: Start with $P = \emptyset$. Use the generation method to build a solution and the local
    search method to improve it. If $x \notin P$ then add $x$ to $P$, else, reject $x$. Repeat until
    $P$ has the user specified size.
 2: Create a reference set $RefSet$ with $b/2$ non-dominated solutions of $P$ and other
    $b/2$ solutions of $P$ more diverse from the previous $b/2$. If there are not enough
    non-dominated solutions to fill the $b/2$, complete the set with dominated solutions.
 3: $NewSolution \leftarrow$ true
 4: while Exists a Solution not yet explored (NewSolution = true) do
 5:   $NewSolution \leftarrow$ false
 6:   Generate subsets of $RefSet$ with at least one non-dominated solution each.
 7:   Generate an empty subset $N$ to store non-dominated solutions.
 8:   while there is a subset to examine do
 9:     Select a subset and mark it as examined.
10:     Apply the combination operator to the solutions in the set.
11:     Apply local search to each new solution $x$ found after the combination process
        as explained in Figure 4 and name it $x^b$.
12:     if $x^b$ is non-dominated by any $x \in N$ and $x^b \notin N$ then
13:       Add $x^b$ to $N$.
14:     end if
15:   end while
16:   Add solutions $y \in N$ to $P$ if there is no solution $z \in P$ that dominates $y$.
17:   $NewSolution \leftarrow$ true.
18: end while
```

Fig. 2. MOSS algorithm

C. Coding Scheme, Combination Operator and Local Search

We used a block representation to code each individual, where each block corresponds to one of the promoter patterns (i.e., TATAAT or TTGACA). Particularly, each block was represented by two integers, where the first number corresponds to the starting point of the sub-motif, and the second one represents the size of the pattern (see Figure 3).

Phenotype

```
         TTGACA                         TATAAT
ACGTAGACCTGTCTTATTGAGCTTTCCGGCGAGAGTTCAATGGGACAGGTCCAG
         ↑                              ↑
      char 9                         char 36
```

Genotype

Gen 1 | Gen 2
[(9,6)] | [(36,6)]
$f_1 = 0.497872$ $f_2 = 0.651163$ $f_3 = 0.333333$

Fig. 3. Example of the representation of an individual

The combination process was implemented as a one-point crossover operator, where the point is always located between both blocks. For example, given chromosomes with two blocks A and B, and parents $P = A_1B_1$ and $P' = A_2B_2$, the corresponding offsprings would be $S = A_1B_2$ and $S' = A_2B_1$. The *local search* was implemented as a search for non-dominated solutions in a certain neighborhood. The neighborhood operator generates a new solution where a chromosome block has been moved a specified number of nucleotides to the left or to the right side. The selection process considers that a new neighbor solution that dominates one of its parents will replace it, but if it becomes dominated by its ancestors, no modification is performed. Otherwise, if the new individual is not dominated by the non-dominated population found so far, it replaces its father only if it is located in a less crowded region (see Figure 4). To do so, $crowd(x)$ is calculated by counting the number of solutions in the neighborhood of x.

```
Randomly select which block $g$ of the individual $c$ will be adapted by local search.
Randomly select a number $n$ in $[-neighbor, neighbor]$ and move $n$ nucleotides
the block $g$. Notice that it can be moved upstream or downstream. The resulting block
will be noted by $g'$ and the resulting individual will be called $c'$.
if $c'$ meets the restrictions then
    if $c'$ dominates $c$ then
        Replace $c$ with $c'$
    end if
    if $c'$ does not dominate $c$ and $c'$ is not dominated by $c$ and $c'$ is not dominated
    by any solution in the Non-Dominated set then
        Replace $c$ with $c'$ if $crowd(c') < crowd(c)$.
    end if
end if
```

Fig. 4. Local search

V. A Genetic Tuning Process for Adjusting the Promoter Fuzzy Models

As seen in Section III, the three fuzzy models built to describe the promoters we want to identify in the DNA sequences have been directly derived from known, average data from previously identified promoters in different sequences. However, this did not ensure the best global performance in the prediction process, as a large number of false positive results (FPs) were obtained (see the results obtained in Section VI). A possible solution to this problem could be to adjust the membership functions, and specially the membership thresholds (a DNA sub-sequence will be considered to match the fuzzy model M^i_α when its membership value to the corresponding fuzzy set is greater or equal than this threshold) of the three fuzzy models to increase the prediction process accuracy, minimizing the number of FPs without greatly reducing the number of true positive solutions (TPs).

This way, the three membership functions of the three fuzzy models (the TTGACA sub-motif, the TATAAT sub-motif and the distance between them) are going to be adjusted, and a membership threshold for each of them will be determined to differentiate between TP and FP estimations.

To do so, a genetic tuning process (see [4], chapter 4) has been developed to optimize the models fuzzy membership function parameters and thresholds. Two different approaches were taken into account: (1) optimizing the models in isolation without the thresholds and, after a good solution is achieved, adjusting the corresponding thresholds; and (2) simultaneously optimizing both models and thresholds dynamically. The second approach provided considerably better solutions than the first and therefore the tuning algorithm was developed using this idea.

The tuning algorithm is based on a memetic algorithm [28] that evolves individuals composed of three different parts:

- The first sub-chromosome includes three floating point numbers, representing the threshold for each sub-motif (thr_{TTGACA}, thr_{TATAAT}, and $thr_{distance(TTGACA,TATAAT)}$).
- The second part encodes the parameters of the discrete membership functions associated to the two *sub-motif objects*, corresponding to a matrix of probability for each alphabet letter in each motif position as in Table I.
- Finally, the third part is for the triangular-shaped fuzzy membership function of the model specifying the distance between motifs, encoded by an array of three integer numbers (a, b and c), as in Figure 6. Any distance below a or above b has a zero membership degree, while the full membership is attained at c.

The initial population consists of a random set of solutions that keeps the main features of the original models from [17] showed in Table I. This means that if, for instance, in the TTGACA pattern the nucleotide T is more frequent than nucleotide C in position 1, then the individuals of the initial population also have the same relation. This happens for the first two models, while the third part of the chromosome is initialized always with the original distribution of [17] as showed in Figure 6(a).

The algorithm works over an input promoter set S comprised by different DNA sequences where several true and false promoters have been identified in advance. The fitness function considered by the genetic tuning process is showed in Equation 3, where $N(x)$ is the set of the individuals in the neighborhood of x that are not dominated by x. In this context, a solution x dominates another solution y as explained in Section IV. The neighborhood of an individual x is composed of all solutions within a small distance from x. Distance is calculated by Equation 5. In this implementation, all the chromosomes with distance ≤ 2 are considered to belong to the same neighbor.

$$Fitness = \sum_{i \in S} isnotdominated(i, N(i)) - \frac{|j \in N(i); j \nprec i|}{|N(i)|} \quad (3)$$

$$isnotdominated(x, P) = \begin{cases} 0 & \exists k \in P; k \prec x \\ 1 & otherwise \end{cases} \quad (4)$$

$$distance(x, y) = |x_1 - y_1| + |x_2 - y_2| \quad (5)$$

where x_1/y_1 is the start position of the first model (TTGACA) in chromosome x/y and x_2/y_2 is the start position of the second model (TATAAT) in chromosome x/y.

A classical, generational evolution scheme is considered, with a binary tournament selection mechanism [6]. As regards the genetic operators, crossover is applied to the first (TTGACA), the second (TATAAT) or the third (distance) sub-motifs on equal probability. For the first and the second ones, a standard one-point crossover is used where the selected point refers to a position in the pattern. In this particular case, positions 2, 3, 4 or 5 of each motif could be chosen and exchanged. On the other hand, for the distance motif, the genotype of one of the two parents is randomly selected to compose the offspring instead of using one-point crossover due to the mutation operator action, as we will see later.

As in crossover, mutation is applied to each of the three motifs on equal probability. Mutation of the first two sub-motifs involves selecting a position from 1 to 6 of the hexamer pattern, adding a random noise in $\{-10, 10\}$ to the chosen probability value, and adjusting the remaining probabilities of the other letters of the alphabet in the same position of the motif to maintain a meaningful probability distribution. For the distance sub-motif, one of the three integers of the tuple is selected with equal probability and a random noise in $\{-3, 3\}$ is added, checking that a valid triangular membership function is obtained. We should notice that this mutation process for the distance motif produces very similar results as the ones achieved with one-point crossover. This is why crossover is simplified to reduce run times without degradation of the final solutions quality.

Finally, a local search method is used only on the first part of the chromosome, that encoding the thresholds of each sub-motif fuzzy model. On equal probability, any of the three threshold values can be chosen after crossover and mutation. The neighborhood operator involves adding a small random noise to the selected integer, and the process is iterated several times till the resulting solution decreases its fitness value. The best solution achieved in this process is then returned.

VI. Experiments Developed and Analysis of Results

The S set of promoters used for both the GAP method and the genetic tuning process is the one described in the Harley & Reynolds compilation [17]. In this work, 272 known promoter sequences from *E. coli* organism are reported along with the exact positions of the submotifs described in Section II. Besides, *alternative* promoters were identified in that work, that is, it is possible to find two or more possible promoters in a same sequence, as illustrated in Figure 5 for the Ada gene. As was previously stated, this is important for the experts to have a broad view of the optimal binding sites as the most probable hypothesis where to perform the experimental mutagenesis process.

A total number of 79 alternative promoters are reported, totalizing 351 promoter locations. The fitness function used in both algorithms was calculated taking into account not only the set S but also penalizing any other good solutions found near each of them.

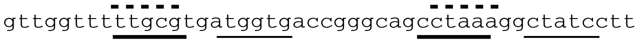


Fig. 5. Different solutions for the Ada sequence

In order to compare the results obtained by the different approaches considered for the promoter discovery on the previous set of DNA sequences, some classical statistic measures are used. We calculate the number of *true positives*, *false positives*, *true negatives*, *false negatives*, *sensitivity*, *specificity* and *accuracy* (see Equations 6, 7 and 8). Moreover, we compute two other statistic measures called *positive predictive value (PPV)* and *overall performance (OP)* [29], showed in Equations 9 and 10.

$$sensitivity = \frac{TP}{TP+FN} \tag{6}$$

$$specificity = \frac{TN}{TN+FP} \tag{7}$$

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

$$PPV = \frac{TP}{TP+FP} \tag{9}$$

$$OP = \frac{accuracy+PPV}{2} \tag{10}$$

The next three subsections are respectively devoted to analyze the performance of the MOSS algorithm in the multiobjective problem solving, the composition of the sub-motif fuzzy models obtained after the tuning process, and the performance of the GAP approach, when applied both over the original and the tuned fuzzy models. The results obtained by the classical Consensus/Patser method [30] will be considered as a baseline for comparison purposes.

A. MOSS Performance

Although the main proposal of the current contribution is not to analyze the quality of the Pareto-optimal solutions derived by the MOSS algorithm, the core of the GAP technique proposed in [10], [11], for the promoter discovery problem, we will briefly show the behavior of this multiobjective EA by comparing it to two well known and state-of-the-art algorithms, SPEA2 [31] and NSGAII [32], using the C metric described in [33]. This metric compares two sets of Pareto-optimal solutions, X and X', giving as result a number in the interval [0,1] that gives the idea of the proportion of solutions in the second population (X') are dominated by the first population (X). In Table II we can see that the MOSS approach is the best one as regards this metric.

$C(X,X')$	MOSS	SPEA2	NSGAII
MOSS	-	0.08481	0.05089
SPEA2	0.00856	-	0.00700
NSGAII	0.02883	0.06358	-

TABLE II

MOSS METRICS

B. Genetic Tuning of the Sub-motif Fuzzy Models

The tuning process was run using the parameter values showed in Table III(a). The adjusted parameters obtained for the TTGACA and TATAAT fuzzy models in our experimentation are showed in Table IV. A brief comparison with the original models of Table I shows that for the TTGACA motif, positions 3,4 and 6 (TT*GA*C*A*) experimented a noticeable change. For them, a higher probability is calculated for the most frequent nucleotide in each position in contrast with the original reports from [17]. The other positions of the motif only suffered minor changes, small increments or decrements. For the TATAAT motif, a not very high discrepancy is observed, only minor changes appeared.

Parameter	Value
Number of evaluations	100000
Crossover prob.	0.6
Mutation prob.	0.2
Local search prob.	1
Population size	50

(a) Parameter values for the genetic tuning process

Parameter	Value
Num. of evaluations	10000
Crossover prob.	0.6
Ref. set size	10
External Population	100

(b) Parameter values for the MOSS

TABLE III

PARAMETER VALUES FOR ALGORITHMS RUN

	T	T	G	A	C	A
A	7	18	3	**78**	21	**83**
C	13	1	8	19	**69**	0
G	5	0	**84**	3	7	7
T	**75**	**81**	5	0	3	10

(a) TTGACA motif

	T	A	T	A	A	T
A	5	**96**	4	**49**	**57**	0
C	17	4	30	20	11	2
G	0	0	12	15	11	2
T	**78**	0	**54**	16	21	**96**

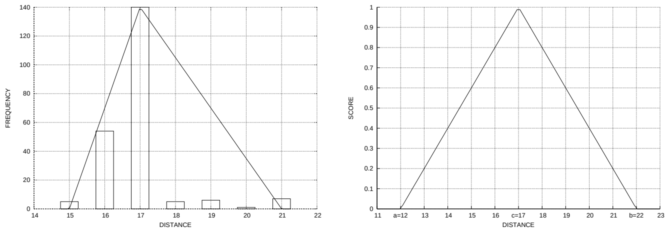
(b) TATAAT motif

TABLE IV

TUNED MOTIF NUCLEOTIDE DISTRIBUTION

On the other hand, membership threshold values of 0.280338, 0.477676 and 0.0797814 were obtained for the three fuzzy models, respectively.

Finally, the membership function derived for the "distance" fuzzy model is that showed on Figure 6(b). As can be seen, the function is centered in 17 and is completely symmetrical.



(a)

(b)

Fig. 6. Graphical representation of M_α^3

C. Experiments with the GAP Method: Tuned vs. Original Fuzzy Models

As said, two set of experiments were performed, one with the original models obtained from [17] and showed in Table I and Figure 6(a), and another with the optimized models learned by the proposed tuning algorithm showed in Table IV and in Figure 6(b), along with the learned thresholds.

Both experiments were done using the same objective functions described in Section IV and run 5 times with different seeds for each input sequence. From now on, a promoter is said to be found if it appears in, at least, one of the result sets. The parameters used in the experiments are listed in Table III(b).

The results obtained by the GAP method using the tuned models are showed in Table V, while those corresponding to the original models are comprised in Table VI. Comparing both tables, V and VI, we can see that, without any threshold at all, the number of FPs is very high. Although the number of TPs shows a small decrease in the tuned model, it clearly improves the general accuracy by obtaining substantially better PPV and OP values, from 0.46 and 0.44 to 0.72 and 0.74, respectively.

Measure	Value
TP	0.82
FP	0.32
TN	0.68
FN	0.18

Measure	Value
Sensitivity	0.82
Specificity	0.68
Accuracy	0.75
Positive Predictive Value (PPV)	0.72
Overall Performance (OP)	0.74

TABLE V

RESULTS FOR THE GAP METHOD USING THE TUNED FUZZY MODELS

Measure	Value
TP	0.85
FP	1.00
TN	0.00
FN	0.15

Measure	Value
Sensitivity	0.85
Specificity	0.00
Accuracy	0.42
Positive Predictive Value (PPV)	0.46
Overall Performance (OP)	0.44

TABLE VI

RESULTS FOR THE GAP METHOD USING THE ORIGINAL FUZZY MODELS

On the other hand, as regards the considered baseline, our method with or without tuning overcomes Consensus/Patser by detecting the 84.33% and 85.47% respectively of the available promoters, while this method, based on weight matrices, only identifies the 74%. More detailed statistics are showed in Table VII where we can observe that, although the total percentage of discovered promoters using the GAP method with the tuned models decrease in approximately 1% comparing to the GAP method without the tuned models, it is compensated by the great reduction of FPs.

	Original	Alternative	%originals	%alternatives	Total	%total
MOSS wo. tuning	246	54	90.44%	68.35%	300	85.47%
MOSS with tuning	244	52	89.71%	65.82%	296	84.33%

TABLE VII

MOSS VS. CONSENSUS/PATSER RESULTS FOR ALL SEQUENCES

VII. Concluding Remarks

GFSs —solving multivariable, multiobjective, optimization problems— provide effective tools to identify interesting features that help to understand complex objects such as DNA sequences. Our proposed promoter recognition method, which was tested by predicting *E.coli* promoters, combines the advantages of feature representation based on fuzzy sets and the searching abilities of multiobjective EAs to obtain accurate as well as interpretable solutions. Particularly, these kinds of solutions, by detecting multiple occurrences of promoters, shed light on different putative transcription start sites. This provides a complete description of diverse regulatory possibilities that can occur in the genome intergenic regions, allowing to predict distinct regulatory activities, harboring activation or repression. The present approach can be extended to identify other DNA motifs, which are also connected by distinct distances, such as binding sites of transcriptional regulators (e.g., direct or inverted repeats). Therefore, by combining multiple and heterogeneous DNA motifs (e.g., promoters, binding sites, etc.), we can obtain different descriptions of the cis-acting regions and, thus, different regulatory environments. The present implementation of our method is available for academic use in the GPS-TOOLS web site (http://gps-tools.wustl.edu).

Acknowledgments

This work was partially supported by the Spanish Ministerio de Ciencia y Tecnología under project TIC2003-00877 (including FEDER fundings).

References

[1] I. Zwir, R. R. Zaliz, and E. H. Ruspini, “Automated biological sequence description by genetic multiobjective generalized clustering,” in *Techniques in bioinformatics and medical informatics*, F. Valafar, Ed., vol. 980, 2002, pp. 65–82.

[2] P. P. Bonissone, “Soft computing: the convergence of emerging reasoning technologies,” *Soft Computing*, vol. 1, no. 1, pp. 6–18, 1997.

[3] L. A. Zadeh, “What is soft computing?” *Soft Computing*, vol. 1, no. 1, p. 1, 1997.

[4] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, ser. Advances in Fuzzy Systems - Applications and Theory. Vol. 19. World Scientific, 2001.

[5] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, “Ten years of genetic fuzzy systems: current framework and new trends,” *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5–31, 2004.

[6] T. Bäck, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. IOP Publishing Ltd and Oxford University Press, 1997.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., 2001.

[8] C. Coello, D. V. Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, ser. Genetic Algorithms and Evolutionary Computation. Kluwer, 2002.

[9] H. Ishibuchi and T. Yamamoto, “Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers,” in *Genetic and Evolutionary Computation (GECCO 2003)*, 2003, pp. 1077–1088.

[10] R. R. Zaliz, I. Zwir, and E. Ruspini, “Generalized analysis of promoters (GAP): A method for dna sequence description,” in *Applications of Multi-Objective Evolutionary Algorithms*, C. Coello and G. B. Lamont, Eds. Singapore: World Scientific Company, 2001.

[11] V. Cotik, R. R. Zaliz, and I. Zwir, “A hybrid promoter anaysis methodology for prokaryotic genomes,” *Special issue on “Bioinformatics”, Fuzzy Sets and Systems*, 2005, to appear.

[12] A. Huerta and J. C. Vides, “Sigma70 promoters in escherichia coli: specific transcription in dense regions of overlapping promoter-like signals,” *Journal of Molecular Biology*, vol. 333, no. 2, pp. 261–278, 2003.

[13] B. M. J. Collado Vides and J. Gralla, “Control site location and transcriptional regulation in escherichia coli,” *Microbiology Review*, vol. 55, no. 3, pp. 371–394, 1991.

[14] C. Mouslim, T. Latifi, and E. Groisman, “Signal-dependent requirement for the co-activator protein rcsa in transcription of the rcsb-regulated ugd gene,” *Journal of Biology Chemestry*, vol. 278, no. 50, pp. 50 588–50 595, 2003.

[15] M. Laguna and R. Martí, *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers, Boston, 2003.

[16] D. Hawley and R. McClure, “Compilation and analysis of escherichia coli promoter dna sequences,” *Nucleic Acids Research*, vol. 11, no. 8, pp. 2237–2255, 1983.

[17] C. Harley and R. Reynolds, “Analysis of e.coli promoter sequences,” *Nucleic Acids Research*, vol. 15, no. 5, pp. 2343–2361, 1987.

[18] S. Lisser and H. Margalit, “Compilation of e.coli mrna promoter sequences,” *Nucleic Acids Research*, vol. 21, no. 7, pp. 1507–1516, 1993.

[19] I. Zwir, P. Traverso, and E. Groisman, “Semantic-oriented analysis of regulation: the phop regulon as a model network,” in *Proceedings of the 3rd International Conference on Systems Biology (ICSB)*, 2003, pp. 282–283.

[20] M. Ptashne and A. Gann, *Genes and Signals*. Cold Spring Harbor Laboratory Press, 2002.

[21] A. Ulyanov and G. Stormo, “Multi-alphabet consensus algorithm for identification of low specificity protein-dna interactions,” *Nucleic Acids Research*, vol. 23, no. 8, pp. 1434–1440, 1995.

[22] C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton, “Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment,” *Science*, vol. 262, no. 5131, pp. 208–214, 1993.

[23] T. Bailey and C. Elkan, “The value of prior knowledge in discovering motifs with meme,” in *Proceedings of the International Conference on Intelligent Systems in Molecular Biology*, vol. 3, 1995, pp. 21–29.

[24] J. V. Helden and B. A. J. C. Vides, “A web site for the computational analysis of yeast regulatory sequences,” *Yeast*, vol. 16, no. 2, pp. 177–187, 2000.

[25] G. Klir and T. Folger, *Fuzzy sets, uncertainty, and information*. Prentice-Hall, Inc., 1987.

[26] W. Pedrycz, P. Bonissone, and E. Ruspini, *Handbook of fuzzy computation*. Institute of Physics, 1998.

[27] E. Ruspini and I. Zwir, “Automated generation of qualitative representations of complex object by hybrid soft-computing methods,” in *Pattern Recognition: From Classical to Modern Approaches*, S. Pal and A. Pal, Eds. Singapore: World Scientific Company, 2001.

[28] P. Moscato, “On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms,” Caltech Concurrent Computation Program, Tech. Rep. C3P Report 826, 1989.

[29] E. Benítez-Bellón, G. Moreno-Hagelsieb, and J. Collado-Vides, “Evaluation of thresholds for the detection of binding sites for regulatory proteins in escherichia coli k12 dna,” *Genome Biology*, vol. 3, no. 3, 2002.

[30] G. Hertz and G. Stormo, “Identifiying dna and protein patterns with statistically significant alignments of multiple sequences,” *Bioinformatics*, vol. 15, pp. 563–577, 1999. [Online]. Available: http://bioinformatics.oupjournals.org/cgi/reprint/15/7/563.pdf

[31] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” Computer Engineering and Networks Laboratory (TIK), Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 103, 2001. [Online]. Available: citeseer.ist.psu.edu/zitzler02spea.html

[32] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[33] E. Zitzler and L. Thiele, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.