Artificial Intelligence Review **20:** 95–120, 2003.

Cooperative Metaheuristics for Exploring Proteomic Data

ROBIN GRAS[1*], DAVID HERNANDEZ[1], PATRICIA HERNANDEZ[1], NADINE ZANGGER[1], YOANN MESCAM[1,2], JULIEN FREY[1], OLIVIER MARTIN[1], JACQUES NICOLAS[2] & RON D. APPEL[1,3]
[1]*Swiss Institute of Bioinformatics, CMU, 1 rue Michel Servet, CH-1211 Geneva 4, Switzerland;* [2]*IRISA-INRIA, Rennes, France;* [3]*University of Geneva, Geneva, Switzerland*
*(*author for correspondence, e-mail: Robin.Gras@isb-sib.ch)*

Abstract. Most combinatorial optimization problems cannot be solved exactly. A class of methods, called metaheuristics, has proved its efficiency to give good approximated solutions in a reasonable time. Cooperative metaheuristics are a sub-set of metaheuristics, which implies a parallel exploration of the search space by several entities with information exchange between them. The importance of information exchange in the optimization process is related to the building block hypothesis of evolutionary algorithms, which is based on these two questions: what is the pertinent information of a given potential solution and how this information can be shared? A classification of cooperative metaheuristics methods depending on the nature of cooperation involved is presented and the specific properties of each class, as well as a way to combine them, is discussed. Several improvements in the field of metaheuristics are also given. In particular, a method to regulate the use of classical genetic operators and to define new more pertinent ones is proposed, taking advantage of a building block structured representation of the explored space. A hierarchical approach resting on multiple levels of cooperative metaheuristics is finally presented, leading to the definition of a complete concerted cooperation strategy. Some applications of these concepts to difficult proteomics problems, including automatic protein identification, biological motif inference and multiple sequence alignment are presented. For each application, an innovative method based on the cooperation concept is given and compared with classical approaches. In the protein identification problem, a first level of cooperation using swarm intelligence is applied to the comparison of mass spectrometric data with biological sequence database, followed by a genetic programming method to discover an optimal scoring function. The multiple sequence alignment problem is decomposed in three steps involving several evolutionary processes to infer different kind of biological motifs and a concerted cooperation strategy to build the sequence alignment according to their motif content.

Keywords: cooperative metaheuristics, evolutionary algorithm, motif inference, multiple sequence alignment, protein identification, proteomics, swarm intelligence

1. Introduction

Metaheuristics are generic methods for non-exact solving of difficult (NP-hard) combinatorial problems (Michalewicz and Fogel, 2000; Yagiura and Ibaraki, 2001). Their global strategy consists in an efficient exploration of

the search space in order to localize reasonably "good" solutions for a given objective function. They can be viewed as toolboxes of optimization methods that can be combined together and with specific heuristics to develop an exploration strategy dedicated to a particular problem. A large variety of such optimization methods is available; they can be classified according to different criteria reflecting a particular property to be emphasized. For example, they are often classified into deterministic and non-deterministic categories depending on the use (or not) of a stochastic process for the exploration of the search space. We are interested in another metaheuristics property leading to classify between cooperative and non-cooperative methods. The non-cooperative metaheuristics are those which explore a unique point of the search space at a given time, like hill climbing, simulated annealing, taboo search, etc. Cooperative metaheuristics correspond to a parallel exploration of the search space by a set of coexisting potential solutions; each solution cooperate with the others by information exchange in order to select new promising potential solutions. This aspect is strongly linked with the concept of building blocks (BB) (Goldberg, 1989, 2002), that are relevant sub-parts of solutions shared by most of the good solutions, because information exchange is a way to detect and transmit building blocks between solutions. We distinguish three sub-classes of cooperative metaheuristics according to the origin of the decision and the nature of information exchange. In the first one, referred to as "centralized cooperation", cooperation between entities is set by an external oracle that selects both the cooperating entities and the content of the exchange. For example, in a classical evolutionary algorithms, cooperation consists in sub-solution exchange performed by the crossover operator. More generally, the centralized cooperation approach can be viewed in a logical framework as a centralized multi-agent system where a master agent set and controls communication between agents (chromosomes/solutions of the evolutionary algorithm). The second sub-class, referred to as "individual cooperation", corresponds to a system in which entities set communication themselves however without any prior information from the other entities. For example, the parallel version of evolutionary programming, based on the injection island model (Fernandez et al., 2000; Golubski, 2002; Lin et al., 1994), is a two level cooperation metaheuristics. The first level is the centralized cooperation between chromosomes in each island and the second level is an individual cooperation between islands. In this second level, each island sends data to its own selection of other islands independently of the other agents needs. The island model not only provides efficient exploration ability but it is also a common way to preserve diversity (Punch, 1998). The swarm intelligence metaheuristics (Bonabeau et al., 1999) is another example of individual cooperation process since the communication between entities

(ants for example) is performed by an indirect vector, i.e., the pheromone. Each entity decides how much and where to deposit pheromones, with no knowledge of the states of other ants. This sub-class of communication can also be assimilated to the blackboard model of multi-agent systems. Finally, the third sub-class, referred to as "concerted cooperation", embodies a cooperation process in which each exchange depends on a mutual agreement between entities. Therefore, the direction and the content of the communication are defined dynamically, in order to optimize the gain of each participant. This model can be much more complex than those described in the other two sub-classes. It implies the definition of a strategy of cooperation for each entity based on knowledge regarding its proper state and possibility of improvement, the information content of the other entities, the strategies of the other entities, etc. However, the exploration of the search space is much more dynamic and "intelligent" due to the association of two factors: first, the diversity, which is preserved by isolation of potential solutions inside agents; second, the flexible and directed property of the information exchange mechanism between entities. Moreover, a hierarchy of multiple concepts can emerge spontaneously from the dynamic association of entities optimizing different objective functions.

Proteomics (Pennington and Dunn, 2001; Wilkins et al., 1997) can be defined as the study of the protein expression pattern of a given tissue or organism at a given time. This involves knowing about large number of different proteins, their possible variants (modifications, mutations, fragments ...), their corresponding amino acid sequence and potential interactions between these proteins. The commonly used technique for proteome analysis involves four steps: protein separation (for example by two-dimensional electrophoresis), protein digestion which produces a set of peptides, measurement of the peptides and peptide fragments masses by mass spectrometry, comparison of mass data with proteic or translated genomic sequence databases. Matched masses are used to identify proteins and their possible variants. The understanding of the possible biological functions of the identified new proteins, protein variants or protein complexes rely on various information sources (experimental data, sequence and/or annotation databases, literature ...). This approach must be automated or partially automated to analyze in real time the large amount of data generated in high-throughput environments commonly set in recent years.

In our research group, we develop different methods for the automation of proteomics data analysis. From our point of view, most of these analyzes can be regarded as difficult combinatorial optimization problems such as, for example, efficient learning of properties from data, classification of complex sets of information, extraction of grammatical structure from sequences, etc.

We apply cooperative metaheuristic approaches in a hierarchical way to manage a wide variety of proteomics problems.

2. Exploiting Tandem Mass Spectrometric Data for Protein Identification

Protein identification is a major issue in proteomics. In a global approach, the challenge is to identify all proteins present in a sample. In high-throughput identification projects, the identification tool should be fast, fully automated and robust. Alternatively, proteins of clinical interest can be targeted by differential expression between two samples. In this case, the identification tool must be particularly tolerant when identifying mutated or modified proteins.

Nowadays, the most widely used technique in protein identification is mass spectrometry (MS). After purification, each protein is digested using a specific enzyme. The masses of the resulting peptides are then measured. The obtained mass list, called a MS spectrum, may already be used for identification by "peptide mass fingerprint", but additional information on the protein sequence can improve the reliability of identification: each peptide is further fragmented (ideally, the fragmentation occurs on the peptidic bonds joining together the amino acids). The fragmentation process generates ionic fragments carrying one or several charges. The fragment masses are measured, leading to a MS/MS spectrum, which includes the mass of the source peptide molecular weight (the parent mass) and of a peak list representing the masses and the intensities of the detected ionic fragments. The identification consists in comparing by the mean of a scoring function the experimental MS/MS spectrum with theoretical peptides or a nucleic sequences referenced in a database (Gras and Muller, 2001).

All existing identification methods from MS/MS data are confronted with combinatorial problems and conveniently justify the use of heuristic strategies. The first and major source of combinatory is the MS/MS spectrum itself. The fragmentation process is hardly foreseeable and depends, among other things, on the amount of energy used by the mass spectrometer, on the number and the repartition of the charges carried by the peptide, and on its sequence. As a result, some positions on the peptide are possibly not fragmented. Moreover, the masses finally measured are modified by various factors, including the exact position of the fragmentation related to the peptidic bond, the number of charges on the ionic fragment, the possible loss of molecules (as water or ammonium), the isotopic pattern of the peak ... Another source of combinatory is possible modifications (adjunction of specific molecule on amino acid) or mutations in the source peptide, resulting in shifts in some of the measured masses.

2.1. *Swarm intelligence for multiple tag extraction*

We present here a new method of protein identification from MS/MS data, called Popitam (Hernandez et al., 2002), in which we deliberately chose to favor non-deterministic cooperative strategies when confronted with combinatorial problems. Our algorithm can be shortly described as follows: after a preprocessing step, in which a given number of peaks in the spectrum are selected according to their intensity, the MS/MS spectrum is transformed into a direct acyclic graph (Dancik et al., 1999) and is compared with theoretical peptides from a database, leading to a ranked list of scored candidate peptides. The graph allows structuring the MS/MS spectrum, and then capturing information about the peptide sequence coming from the relative positions of the peaks in the spectrum. The vertices, which are built from the peaks, represent masses of hypothetical "ideal" fragments. They are scored according to the credibility level that can be assigned to them. Vertices whose masses differ by the mass value of one or two amino acids are connected by an edge. Reference to Dancik et al. (1999) provides a more complete description of the "spectrum graph". As a result, the graph represents all possible complete sequences and sub-sequences that can possibly be built from the spectrum. The sequences can be constructed by moving from one vertex to another by following existing edges. Only a few sections among the huge possibilities of paths in the graph represent real sub-sequences of the source peptide. Identification methods based on tag search typically try to extract tags (or complete sequences) from the graph (Chen et al., 2001; Dancik et al., 1999; Schlosser and Lehmann, 2002; Taylor and Johnson, 1997) and used to select candidate sequences from the database (Mann and Wilm, 1994). In Popitam, on the contrary, the database is used to direct the search and to emphasize relevant sections in the graph from which the peptides can be scored. Several pieces of information are taken into account, as for example, the intensity of the peaks, the credibility score of the vertices, or the linear correlation of the error between the masses of the vertices and the masses computed from the theoretical peptides (Gras et al., 1999, 2000).

Although the search is directed by a theoretical sequence, the search space can be vast, particularly if modifications or mutations are allowed. In Popitam, the graph exploration process is performed by an Ant Colony Optimization (ACO) algorithm (even if the computing time is rather long compared to other algorithms). ACO algorithms, directly inspired from the ant foraging behavior, are well suited to "shortest paths" problems and can be easily adapted to a specific problem. The search is parallelized, as several ants work at the same time, and the exploring approach is stochastic. The method can be classified as constructive, because the solution is built step by step by moving from adjacent states (partial solutions) of the problem.

It can also be classified as cooperative, since ants communicate indirectly by the mean of the pheromone trail. For a given spectrum, the complete identification process consists in applying the ACO algorithm to each peptide of the considered sequence database, leading to a ranked list of theoretical peptides. A description of the ACO used is given in Dorigo and Di Caro (1999). In our application, the transition rule used to move from a vertex to another connected one takes into account four pieces of information: (1) the visibility, symbolizing the *a priori* desirability of the move, and represented by the credibility scores of the vertices, (2) the knowledge acquired in previous iterations, symbolizing the *a posteriori* desirability of the move, and represented by the amount of pheromone deposited on the edges, (3) a description of the theoretical sequence being compared, and (4) the memory of the parsed vertices.

A first version of Popitam, denoted as the "Full Path algorithm", has been implemented and tested. As suggested in the name, this version works by searching for complete paths (named L^k) in the graph when comparing a theoretical sequence with the experimental spectrum. In other words, the ants start on the initial vertex and can go forward as long as the current vertex has one or more successors. Once an ant has completed a path L^k and hence, built a complete sequence from the MS/MS spectrum, it evaluates the quality of the solution using a scoring function. This is a key process of Popitam's algorithm. The score represents the similarity S^k between the current peptide and the path used by ant k. The function also contributes to the final identification score of the current peptide (S^+). The goal is to include relevant information, represented by several sub-scores, in S^k (and S^+). We have, for the moment, four sub-scores: a coverage score *covS*, which is the coverage percentage between the theoretical peptide from the database and the sequence parsed in the graph; an intensity score *intS* corresponding to the mean of the peak intensities included in L^k; a relevancy score *relS*, computed from the credibility values associated with the vertices in L^k, and a regression score *regS*, measuring the linear deviation between the experimental masses of the peaks included in L^k and the corresponding theoretical masses computed from the theoretical peptide. Each sub-score is scaled so that it varies between 0 and 1. Still other information can be added, such as expert rules set by biologists used to studying MS/MS data.

2.2. *Discovery of an optimal identification scoring function by Genetic Programming*

We have used Genetic Programming (GP) (Koza, 1992) to find a scoring function that efficiently discriminates the good peptide from the others of the database. Evolutionary algorithms (EA) are stochastic search methods

inspired by natural mechanisms. GP applies EA to a population of computer programs. Programs are hierarchical structures of dynamically varying size and shape. A program can also be seen as a parsed tree with ordered branches in which the internal nodes are functions and the leaves are the so-called terminals of the problem. GP provides a way to search the space of all possible programs composed of functions and terminals to find a solution appropriate to the given problem. The evolutionary process starts with a population composed of $\boldsymbol{M}$ random programs. Then this population applies the Darwinian principle of survival of the fittest and genetic mechanisms borrowed from biology to breed a new population of programs. This breeding process is repeated during $\boldsymbol{G}$ generations in order to produce better and better approximations to an optimal solution of the given problem by exchanging "genetic information" (BB) of promising points of the search space. The evolution is guided by a fitness function that determines how each program in the population solves the problem.

There are several preparatory steps to use GP such as determining the set of terminals, the set of functions, the fitness measure and the parameters for controlling the run. For our problem, a program is an arithmetic function involving information from the identification process. The set of functions, named $\boldsymbol{F}$, is composed of standard arithmetic operators of addition, subtraction, division, multiplication and power. The set of terminals, named $\boldsymbol{T}$, consists of the four sub-scores previously described and constants, named C, generated between –5.00 and +5.00. Thus, $\boldsymbol{F} = \{+, -, \div, \times, \wedge\}$ and $\boldsymbol{T} = \{\text{intS}, \text{relS}, \text{regS}, \text{covS}, C\}$.

The fitness function is the driving force of the evolution. It measures the adequacy of a program to the given problem. For our problem, the fitness indicates how a scoring function encoded by a program is able to discriminate, for a given MS/MS spectrum, the correct peptide from the others of the database. The fitness of a program is computed by using its scoring function to identify a learning set of MS/MS spectra with the "full path" algorithm. For each learning spectrum a fitness value is computed from the ranked list of results of the identification. This fitness value is computed with a fitness function based on the optimization method of SmartIdent (Gras et al., 1999). The fitness of a program is the average of the fitness values obtained for each spectrum of the learning set.

Genetic operators are the engines of the evolution. They allow producing offspring by combining and modifying the "genetic information" contained in the individuals of the population. We have used three different operators named crossover, mutation and permutation. They operate on parent trees that are selected from the population with a tournament selection method of size $\boldsymbol{N}$ (Blickle and Thiele, 1995). The crossover operates on two parent trees. The

process begins by independently selecting a crossover point in each parent tree. Then the sub-trees, which roots are a crossover point, are exchanged, giving rise to two offspring trees. The mutation operator consists in replacing a sub-tree of a parent tree by a new sub-tree randomly generated. Finally the permutation operator works by selecting a random internal node of a parent tree and permuting the order of its arguments. The best run was obtained with the following main parameters: $\boldsymbol{M} = 40$, $\boldsymbol{G} = 50$, $\boldsymbol{N} = 3$. The set of learning spectra used to evaluate the fitness of programs is composed of 174 MS/MS spectra. Each of these learning spectra has been identified by spectrometry experts. The optimized scoring function obtained by GP is the following:

$$(\text{cov}S \cdot (x)^x)^{4,9}, \text{ with } x = (\text{int } S + \text{cov } S) \cdot 4, 9^{perS} \tag{1}$$

2.3. *Results*

Popitam's Full Path algorithm was tested with a set of 271 MS/MS spectra obtained from nucleolar proteins purified by SDS-PAGE and 2D electrophoresis, digested with trypsin and analyzed with a Q-TOF mass spectrometer (Scherl et al., 2002). A quality value p has been computed for each of them, according to (Pevzner et al., 2001). This value represents the average of the ion type b and y frequencies observed in the spectra. Popitam was able to correctly identify 86.7% of the spectra. Table 1 shows the success rate of Popitam according to the value p. We consider here a spectrum as identified if its parent peptide is in the first position of the peptide list of results. These results have been obtained with the optimized scoring function (equation 1) and are better than those obtained with an empirical scoring function defined by a spectrometry expert (83%).

3. A Hierarchical Cooperative Multiple Sequence Alignment Combining Local Similarities

3.1. *Introduction*

In this section, we describe a cooperative multiagent strategy that takes advantage of concerted cooperation to achieve a fully automated clustering of biological sequences. Clustering leads to the emerging construction of a multiple sequence alignment based on regular single motifs (section 3.2) and linked dyad motifs (section 3.3). During evolution, DNA sequences are subject to mutation. These mutations may have very different outcomes depending on where they occur. A point mutation that does not affect the

Table 1. Success rate of Popitam according to the value of p

p	spectrum number	% success
[0;0.1[	0	—
[0.1;0.2[	7	0%
[0.2;0.3[	22	45,5%
[0.3;0.4[	59	84,7%
[0.4;0.5[	97	91,8%
[0.5;0.6[	61	100%
[0.6;0.7[	23	100%
[0.7;0.8[	2	100%
[0.8;0.9[	0	—
[0.9;1]	0	—
[0;1]	**271**	**86.7%**

survival of an organism is likely to remain and be passed on future generations. On the other hand, a mutation on a regulatory site (TATA-box of a gene promoter for example) may have dramatic consequences on the survival of the organism. Thus, this mutation is not likely to propagate to future generations. Considering a set of biological sequences known to be related (a set of promoters or a family of protein sequences for instance), common features or similar sub-words shared by all the sequences can be determined. We suspect that these similar sub-words have been kept by an evolutionary pressure, because they are involved in a biological process. Then the problem can be set as follows: given a set of sequences, extract one word (of a constrained length) per sequences such that all extracted words share a maximum global similarity. These extracted words will then constitute a local multiple alignments and the corresponding motif is members of regular motifs of Chomsky's hierarchy. This problem is addressed in section 3.2.

The three-dimensional shape of a protein bears on the protein function. This shape is constrained by physico-chemical interactions, which determine to the secondary and tertiary structures. Physico-chemical interactions are known, but it is very difficult to precisely predict their involvement in the folding of a linear sequence of amino acids into a structured protein. Motifs used in biology to characterize families of proteins are often single words that represent conserved regions in the corresponding set of related sequences. However, this kind of motif may not be descriptive enough because protein properties depend greatly on physico-chemical interactions between distant regions. Thus, a descriptive motif should at least be made up of regions linked by some dependence. This problem is addressed in section 3.3.

MSA is a very difficult problem ever present in bioinformatics. The alignment of a collection of biological sequences significantly contributes to the field of protein characterization: it applies to phylogenetic analysis, structural modeling or functional annotation transfer from characterized to new sequences (Mullan, 2002). Several heuristic MSA algorithms have been developed so far, ranging from traditional progressive methods (Feng and Doolittle, 1987; Higgins and Sharp, 1989; Thompson et al., 1994) to computationally expensive score optimization strategies (Gotoh, 1996; Morgenstern, 1999; Notredame et al., 2000; Stoye, 1998). The quality of alignments produced is highly dependent on the initial data (Lassmann and Sonnhammer, 2002). None of these methods performs well in the variety of situations raised by biological data, for example low sequence identity, complex or inverted sequence similarity architecture, or variable sequence length.

3.2. *A strategy to explore the local multiple alignment search space*

Local multiple alignment consists in choosing exactly one word per sequence, so that all chosen words are maximally conserved. Several tools already exist. They differ in several points: the selected strategy that explores the search space, and the objective function to be optimized. We can distinguish three main categories. First, methods dedicated to infer gapped-words, which have exact occurrences in a given number of sequences. These methods are essentially Pratt (Jonassen et al., 1995) and Splash (Califano, 2000). The second category looks for a set of occurrences at a Hamming distance of k or at most k of an external unknown word (the hypothetic common ancestor). We can cite Smile (Marsan and Sagot, 2000), Winnover (Pevzner and Sze, 2000), Projection (Buhler and Tompa, 2002), and Multiprofiler (Keich and Pevzner, 2002). The third category contains methods that look for a set of occurrences strongly conserved from the point of view of information content. This problem has been shown to be NP-hard (Akutsu et al., 2000). The main ones are the Gibbs Site Sampler (Lawrence et al., 1993), Meme (Bailey and Elkan, 1994), and Consensus (Hertz and Stormo, 1999). Our interest is in this third category. Our method called MoDEL (Motif Discovering with Evolutionary Learning) (Hernandez et al., 2002) is based on a centralized cooperative metaheuristics that is used to find good seeds. Its efficiency rests on the use of two complementary search spaces: (1) the search space made up by all possible words of a given length, (2) the search space made up of all possible alignments of this given length. An evolutionary algorithm samples the word search space, in order to find goods seeds in the alignment search space. These good seeds are then optimized using a simple hill-climbing operator. We show significant improvement in the exploring capacities over other cited methods.

3.2.1. *Definitions*

Let $S = \{S_1, S_2, \ldots, S_N\}$ be a set consisting of N sequences of length T (to simplify, we suppose that all sequences have the same length). $S_{i,j}$ is the j^{th} symbol of the i^{th} sequence. S is defined over a fixed size alphabet $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$ with $K = |\Sigma|$ (4 for DNA and 20 for proteins). Let $M = \{m_1, m_2, \ldots, m_W\}$ be a word of length W defined on Σ. The whole possible M defines the search space $\mathcal{M}$ of size K^W. Let $P = \{p_1, p_2, \ldots, p_N\}$ be a position vector, which define occurrences in the sequence set. p_i is a position on S_i and defines the sub-word of length W beginning at p_i. Thus, P represent a set of words or a local multiple alignment of S. The whole possible P define the search space P, of size $(T - W + 1)^N$.

The purpose of our method is to find the point of P that maximizes an objective function. As an information content measure, we use the Kullback-Leibler distance (also known as the relative entropy). We first estimate a frequency matrix F from the alignment as follows: for $k = 1, \ldots, K$ and $j = 0, \ldots, W - 1$:

$$F_{k,j} = \frac{1}{N} \sum_{i=1}^{N} (k, S_{i,p_i+j}) \quad \text{with} \quad j(k, a) = \begin{cases} 1 \text{ if } a = \sigma_k \\ 0 \text{ else} \end{cases} \tag{2}$$

Information content is calculated from this matrix by:

$$I(F, F^0) = \sum_{i=1}^{K} \sum_{j=1}^{W} F_{i,j} \log \frac{F_{i,j}}{F_i^0} \tag{3}$$

where F^0 is the background frequency estimated once, during the initialization step, from S. For convenience, instead of using $I(F, F^0)$, we will use I(P).

We now define how the two search spaces are mapped together. Operators were defined to allow the projection of a point from $\mathcal{M}$ to $\mathcal{P}$ (MtoP) and inversely a point from $\mathcal{P}$ to $\mathcal{M}$ (PtoM). In the case of the MtoP operator, a position vector should be obtained from a word. This is achieved by aligning the word M with all the sequences of S. More formally; for $i = 1, \ldots, N$ and $j = 1, \ldots, T - W + 1$:

$$p_i = \underset{j}{\operatorname{argmax}} \sum_{k=1}^{W} H(S_{i,j+k-1}, m_k) \quad \text{with} \quad H(a, b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ else} \end{cases} \tag{4}$$

In this way, p_i corresponds to the position on S_i where the match count is maximal. This procedure often leads to a position vector with several ambiguous p_i. This is due to multiple maximum positions on a single

sequence. In this case, we use a greedy algorithm to clear up ambiguities. Let us stress an important property of MtoP operator. All selected positions on S will share a much greater similarity than positions selected by chance since positions on S originate from an alignment. Any point of $\mathcal{P}$ obtained by a projection from $\mathcal{M}$ will have a fitness value widely greater than a random point of $\mathcal{P}$. This feature allows us to directly find good points in $\mathcal{P}$ that will act as seeds. We also defined the complementary operator PtoM. In this case, a relation strip from the alignment space to the word space is defined as follow: considering the frequency matrix F (equation 2) we can derive from a point of $\mathcal{P}$ (an alignment) the corresponding word, which is the most likely word according to F. More formally, for i = 1, . . . , K and j = 1, . . . , W:

$$m_j = \sigma_k \quad \text{with} \quad k = \underset{i}{\operatorname{argmax}}\, F_{i,j} \tag{5}$$

It is important to note that these two operators are not symmetrical, a point of $\mathcal{M}$ projected to $\mathcal{P}$ and projected back to $\mathcal{M}$ will not necessarily be the same as the initial one. However, this feature will be used during the exploration. Finally, we define now $\mathcal{Q}$, a sub-space of $\mathcal{P}$, which is made up by all point in $\mathcal{P}$ that can be reached from $\mathcal{M}$ by a projection. The size of this sub-space is at most equal to $\mathcal{M}$'s one.

3.2.2. *Exploration*

The overall strategy of MoDEL is to produce reasonably good points of $\mathcal{P}$ to act as seeds for a hill-climbing optimization. An evolutionary algorithm is used to sample the $\mathcal{Q}$ search space in order to produce these seeds. We used a very similar Hill climbing approach, as the one used by the Gibbs Site Sampler (Lawrence et al., 1993). Given a position vector P = $\{p_1, p_2, \ldots, p_N\}$, only one p_i dimension is modified at once, keeping the others fixed. This dimension is then sampled on all possible values (corresponding to the T – W + 1 possible sub-words of S_i), and is updated to the value that maximizes the overall information content I(P). All dimensions are modified one after the other, either in a predefined order or in a random without replacement order. The whole process is performed until complete convergence, which usually takes less than ten cycles. This hill climbing process is coupled together with a phase shift correction (Lawrence et al., 1993) consisting in moving, several positions to the left or to the right, the whole alignment to ensure not being in a shifted sub-optimal position.

A classical evolutionary process is performed to generate seeds of $\mathcal{P}$ by exploring $\mathcal{Q}$. The particularity is that moves are made on $\mathcal{M}$ while fitness values are calculated on $\mathcal{P}$, through the MtoP operator. Chromosomes contain a word and a fitness value. The population contains 200 chromosomes. The evolutionary process uses a set of seven genetic operators, including several

original ones in addition to classical operators like crossover or mutation. Binary operators consist in three different crossovers: (1) the simple crossover (SCOP), (2) the point-crossover (PCOP), which allows swapping symbols independently with a probability of 0.35 for each position, (3) the slide-crossover (SCOP), which is more distinctive. If we consider two words that are converging in the same direction except a shift of one, two or more symbols, performing a classical crossover will not be constructive, because some non-homologous information will be exchanged. Instead of aligning each chromosome exactly in front of the other, a slide is performed to search for a layout that maximizes a similarity score. The aim is to increase the likelihood that swapped positions correspond to homologous information. The score of a particular layout will be given by the Hamming distance (sum of matches) of the two overlapped sub-words, divided by the total length of the alignment. Exchanges are performed on the overlapped sub-words, in the same manner as the point-crossover. Unary operators are the mutation and the slide. Slide operator (SOP): all symbols in the word are shifted one position to the left or to the right. The missing symbol is reinitialized. The mutation (MOP) simply reinitializes positions in the word. The two remaining operators (MPMHC & MPMPS) have a very local range. Whereas preceding operators perform a move from $\mathcal{M}$ to $\mathcal{M}$ search space, these two perform a move from $\mathcal{M}$ to $\mathcal{M}$ via the $\mathcal{P}$ search space. They consist, for a word of $\mathcal{M}$, in using a $\mathcal{P}$ optimization (hill climbing and phase shifts) on its corresponding position vector. MPMHC uses the hill climbing optimization, and MPMPS uses the phase shifts. The new word is produced by projecting the optimized $\mathcal{P}$ point back to $\mathcal{M}$. Due to the very local range of these two operators, their probability has to be low. We give in Figure 1 an overview of the interconnection of the two search spaces via the different operators.

3.2.3. *Results*

We compared the exploration capacities of MoDEL and the Gibbs Site Sampler. As far as we know, no published method generates a sequence set containing a known maximum alignment, just above noise level (necessary condition for the problem being hard). We chose to evaluate these methods on random sequences. We realize that no alignment will be statistically significant, but our aim is to compare the exploration strategy of the two methods. 500 different sequences sets were generated. Each set contains 25 sequences of length 1000, cardinal = 4 and each symbol is equiprobable. The goal is to find the most conserved alignment of length 15. One may ask if these sequence sets are suitable for a neighboring exploration strategy (locally monotone). We can point that both MoDEL and the Gibbs Site sampler give much higher results compared to random exploration. We can there-

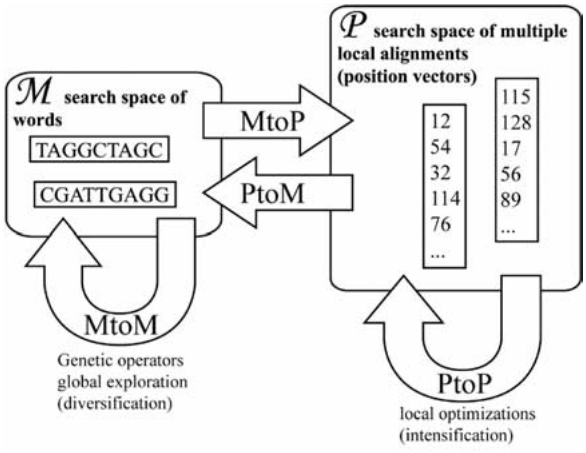

Figure 1. MoDEL uses two representations: (1) $\mathcal{M}$, the search space of words and $\mathcal{P}$ the search space of local multiple alignments. $\mathcal{M}$ is sampled by an evolutionary algorithm. Genetic operators are represented by the MtoM arrow. Randomly chosen good solutions of $\mathcal{M}$ are sent to the $\mathcal{P}$ search space to act as seeds. This is done through the MtoP projection operator. PtoP arrow represents the non-cooperative optimizations in $\mathcal{P}$ (hill climbing and phase shifts), which are applied to these seeds. PtoM projection is used by MtoMviaP operators (MPMHC et MPMPS).

fore assume that these search spaces present locally monotone properties and are suitable for a local heuristic exploration. We adjusted the parameters of the two methods to give a comparable CPU time to both of them. Model performed a number of 900 generations per run. The Gibbs Sampler parameters were chosen with the intention of maximizing its performance while allowing the same CPU-time for both methods. CPU-time in seconds was 529 for MoDEL and 636 for the Gibbs Site Sampler (Pentium4 1.5GHz). Model found a better result in 90% of the sequence sets, both methods were equal in 5%, and the Gibbs Site Sampler found a better result in 5%.

3.3. *Linked dyad motif discovery from combination of conserved regions*

3.3.1. *A covariation measure for structural motifs*

Since biological mechanisms leading to the three-dimensional protein folding are not sufficiently known yet to be used efficiently for our problem, we focused on the idea of using covariation to detect a statistical dependence between conserved regions in a set of sequences. Thus, we hope that regions correlated in this way have a real biological meaning (Fukami-Kobayashi et al., 2002; Rigden, 2002).

We define a linked dyad as a pair of conserved motifs (position vectors of section 3.2) with a significant covariation value between them. In linguistic

terms, linked dyads belong to the upper class of context dependent languages in Chomsky's hierarchy and thus are considered very difficult (NP-hard) to infer from examples. Our goal is to infer such motifs from a given sequence set, and because it is unrealistic to achieve that by an exact method, we will use an evolutionary cooperative metaheuristic based on concepts defined in MoDEL.

If we have a set of sequences on alphabet Σ and two position vectors of length N (with N the number of considered sequences), we define as the mutual information measure between the two vectors P1 = {P1[1], ..., P1[N]} and P2 = {P2[1], ..., P2[N]}. This definition can be extended that to measure the covariation between two conserved patterns $P_1^{1\ldots W}$ and $P_2^{1\ldots W}$ of same size W beginning at position vectors P_1 and P_2:

$$C_L(P_1, P_2) = C(P_1^{1\ldots W}, P_2^{1\ldots W}) = \sum_{i=1\ldots W} M(P_1^i, P_2^i) \text{ where}$$
$$P_1^1 = P_1 \text{ and } P_1^i[k] = P_1[k] + i - 1 \tag{6}$$

$$\text{We also compute: } \bar{C}_L(P_1, P_2) = \sum_{i=1\ldots L} M(P_1^i, P_2^{L-i+1}) \tag{7}$$

which takes into account the reverse order corresponding to "palindromic" covariation. We then select pairs of position vectors with high value of C_L or $\bar{C}_L$. However, we note a drawback for this measure: two non-conserved position vectors have a high C_L (and $\bar{C}_L$) value. This phenomenon has no effect in our search because we use MoDEL as a first step for extracting conserved regions as candidate for the covariation measurement.

3.3.2. *Evolutionary strategy for linked dyad discovery*

MoDEL works well to find independent conserved regions, but the diversity of the population obtained after several generations is poor: weaker chromosomes are discarded during the evolutionary process and the resulting solutions are always slight variations of the best one. To avoid that, we use the concept of ecological niches (Goldberg, 1989). Each chromosome fitness is balanced with a distance between the chromosome and the rest of the population. Thus the fitness of a "weak" chromosome with a few neighbors (similar chromosomes) in the population is as good as that of a "strong" chromosome with many neighbors. So the population will be distributed among many optima, and not only around one unique optimum like in MoDEL. With this modification the MoDEL evolutionary process finally yields several different conserved regions. If we use MoDEL as a generator of conserved regions, we can compute the covariation between all possible pairs of such regions and

keep the best pairs. Depending on the number of possible pairs, conserved regions can be combined via a heuristic step.

3.3.3. *Preliminary results*

In order to validate our hypothesis on covariation measurement, we have implemented a generator which produces sequences similar to those of the Pevzner's challenge (Pevzner and Sze, 2000), though, instead of regular motifs linked dyads are inserted. Given a consensus word, it is inserted in each sequence with a given number of errors. Then, for each of its occurrences, a second word, which is a morphic transformation of the first one, is inserted. Any mutation in the first word will be reflected onto the second one, creating two sets of words complying with our definition of a linked dyad. This morphism is a simple mathematical function from Σ to Σ that changes a word into another word of same length, letter by letter, from right to left or from left to right. Given Σ, we randomly generated this morphism. Some errors are allowed during the transformation. Thus the second part of the linked dyad can be less conserved than the first part. We have used this generator to create sets of 50 sequences of length 100, with inserted linked dyads of length 15 (for each sub-word), at most 4 errors in the first sub-word and at most 4 errors in the morphic transformation.

We used MoDEL (modified as described above) on these sets of sequences, and calculated the covariation values of all possible pairs composed of regions generated by the evolutionary process. The best covarying pair always corresponded to the inserted dyad (i.e., the position vectors were identical). The parameters applied were a run of 50 generations on a population of 200 chromosomes of length 15. Note that we gave the chromosomes the appropriate length; with a smaller (resp. greater) length, we would have retrieved regions that are included (resp. contained) in the inserted regions. The evolutionary process took 40s (on a Sun Ultra 10), using 10Mb of memory. The covariation computation is very time consuming (about 20 min) if we consider all the pairs, but is reduced to a few seconds with a heuristic approach selecting the most promising pairs.

3.4. *A cooperative multiple sequence alignment algorithm based on biological domain composition*

MSA requires a very flexible heuristic to produce a biologically meaningful alignment, keeping time and computation resources within reasonable ranges and preserving robustness with regard to the initial data set (the number of sequences, the degree of similarity between sequences, the variability in domain composition and organization). Several difficulties are encountered: the choice of a pertinent scoring system and the efficient exploration of the

search space, which is exponential to the number of sequences. MSA is therefore highly combinatorial and requires strong heuristic strategies since exhaustive methods are impracticable, even for reasonably sized data.

A concerted cooperative multiagent approach, which is a distributed, dynamic and highly adjustable tool, is appropriate in this situation. The particularity of our strategy rests on the application of a novel distributed clustering stage based on biological domains to produce a hierarchy of clusters assembling naturally into a multiple alignment. The heuristics lies in splitting the problem into a variable number of agents that consider only a single domain at a time and achieve a dynamic assembly of sequences around this domain. Each agent strives towards a multi-objective goal, the maximization of the number of sequences on one hand, the optimization of the domain fitness on the other hand. Concerted cooperation is driving the process towards a high quality clustering of the data set, escaping local optima. This strategy, unlike traditional clustering methods, does not require to pre-set the number of clusters and tolerates overlapping clusters. The process is robust with regard to starting conditions. Since this approach is entirely founded on the notion of domains, it should be able to take advantage of any single piece of information contained in biological domains. The term 'biological domain' covers here not only regular motifs (section 3.2), but also linked dyad motifs (section 3.3). Moreover, domains found only in a sub-set of sequences should also be considered during the alignment process. Finally, the order in which the different domains are organized in the sequences should not impair on the procedure but rather bring additional information. Basic concepts, and a first draft of this novel MSA algorithm as well as preliminary results are presented in the following text.

3.4.1. *Clusters and sequences scoring system relies on domain information*
The dependency on pairwise similarity, either global or local, is a major weakness of the existing sequence clustering or alignment methods. The global measure is not optimal to describe the homogeneity of a group of biological sequences, since it does not reflect their domain architecture, which is fundamental to their biological function. We propose a new similarity criterion that does not rely on pairwise comparisons but includes the relative entropy of a regular motif inferred across all sequences on the one hand (section 3.2, equation (3)); and the co-variation measure of linked dyad motifs on the other hand (section 3.3, equations (6) and (7)). This two-component similarity criterion is local and hence it accounts for complex domain architecture with potential inversions. In addition, it is determined by the information content across all sequences and as such embodies transversal information that represents simultaneously the conservation of all considered

sequences. Therefore, it is more suitable than a pairwise score to characterize homogeneity within a group of sequences. Finally, the contribution of each sequence to this similarity measure is computed and serves as a basis for sequence exchange between cooperating agents or clusters during the optimization process.

3.4.2. *Cooperative agents exploration of the search space*

Exploration of the partition space, especially when overlapping clusters are authorized, is a NP-hard problem. To bypass this situation, we propose a cooperative heuristic that distributes the exploration among a variable number of agents, each agent being responsible for one cluster. As a result, the original NP-hard problem is split into several sub-problems. The optimization of a single sequence family can be properly addressed by a single agent. Each agent cooperates with other agents to perform a local optimization on the corresponding sequence family. Concerted cooperation is a decision-based communication procedure that results in sequence exchange between agents. Individual actions on sequences are also performed. This approach can be seen as a balance between two procedures: intensification and diversification of the search. Intensification tends to focus on promising points of the search space, whereas diversification directs the search towards yet unexplored points.

Intensification implies small moves in the search space. It is mostly achieved by concerted cooperation between agents. The cooperation is guided by two components: first, the agent's strategy, which is dependent on its own state and second, the state of its potential partner. More precisely, agents' size and fitness as well as sequence scores guide the choice for a partner and the type of sequence to be exchanged. Once an agreement is reached between two partners, the actions are executed. Diversification induces more radical moves in the search space. It encompasses birth of new agents, death of incompetent agents, fission of low-fitness agents or fusion of similar agents. Agents have also the possibility of picking new sequences from the initial set randomly, to maintain some diversity and avoid local optima. Inversely, they can trash bad sequences to avoid uncontrolled growth of the family and speed the convergence towards a stable sequence cluster.

Each agent pursues two goals simultaneously: the gathering of a maximum of sequences on the one hand, the optimization of the fitness on the other. Agents use a multi-objective optimization strategy (Coello Coello et al., 2002) in order to select actions that will reach their final objective as closely as possible, that is define a cluster of a biologically meaningful sequence family. Once the clustering process has reached an equilibrium state, then a multiple alignment can be constructed. The sequence family of each agent

embodies a building block of the whole cluster set. During the optimization process, agents self-organize into a hierarchical structure. The assembly of all families guided by the natural hierarchy present among agents will lead to an emerging multiple sequence alignment.

3.4.3. *First draft*

In the present chapter, we describe a first draft of the clustering algorithm involving cooperative agents. A parallel implementation has been performed using the 'message passing interface' package MPI (Pacheco, 1997). The global algorithm follows an asynchronous course where each agent performs a succession of actions independently of other agents' progress. Once the agents are initialized, they run a number of MoDEL generations to compute a preliminary best motif with its relative entropy, set as the agent's fitness. The contribution of each sequence to the agent's fitness is also computed. These scores and the best motif found so far are sent to all other agents so that they acknowledge each other. The agents check the information received from other agents, if any and update their data. After this communication step, if the fitness of an agent is low and promotes cooperation, a different communication procedure prompts the agent to decide which exchange should take place with which partner and actually sends the sequences. When this second communication is over, the agent looks for putative received sequences. It decides whether or not to pick new sequences or trash some poor scoring sequences. Finally, the agent updates its sequence set and starts again the succession of actions of the algorithm.

We describe here a simplified version of the decision algorithm accomplished by each agent when a cooperation process is initiated. Each agent computes the similarity between its own motif and the partner's motif. If this measure is above a set threshold, a highly scored sequence, absent from the partner's sequence set, is sent to the partner. If the two motifs are not similar, in 30% of cases a poorly scored sequence, absent from the partner's sequence set, is sent; in 70% of the cases no action is undertaken.

3.4.4. *Preliminary results*

Our clustering approach was tested on artificially-generated DNA (cardinal 4) sequence families in which a conserved motif of length 15 with 2 errors was inserted (Pevzner and Sze, 2000). A set of 100 sequences, 5 families (5 different motifs) of 20 sequences of length 600, was produced. A population of 14 agents received each a random subset of 46 sequences from the initial set. The subset was iteratively updated during the clustering process, by sequence exchange between agents as described in section 3.4.3. Each iteration implies 10 generations of MoDEL to focus on a promising motif of the current subset. A total of 50 iterations requires 71 seconds CPU-time

when distributed on a cluster of 14 Pentium4 1.5GHz processors. This whole clustering procedure was repeated 100 times on different data sets for statistical reasons. The results obtained were the following: among the 5 families contained in the initial set, a mean of 4.31 was identified by the agents after 50 iterations. At the end of the procedure, each cluster contained a mean of 52.2% of sequences from its identified family, the remaining was distributed among the 4 other families. Finally, each cluster contained on average 69.5% of the sequences from its identified family (14 sequences out of 20). Although this version of the algorithm still lacks many subtleties proposed in the conceptual method, it produced encouraging results, confirming the suitability of the scoring system. Indeed, the contribution of each sequence to the entropy of the inferred motif is appropriate to distinguish blocks of sequences with a common property and drives the process towards the emergence of accurate clusters.

4. Optimization of the Use of Genetic Operators for an Efficient Exploration

When it comes to the optimization of GA, the most common questions are: how to explore the search space more efficiently? How to avoid being trapped in local optima? When is it necessary to force the exploration in different regions of the search space, and when is it necessary to exploit the solutions already available by performing a local search? These questions are related to the control of the parameters of the GA, for example the number of chromosomes, the different probabilities of the operators or the duration of the simulation. Usually the values of these parameters are either kept constant or modified in a constant way, without knowing the effective influence (either positive or negative) of such a method on the results. Most of the time, the tuning of these values is made by hand. This approach can be tedious, and even intractable when too many parameters are taken into account. It would be more interesting to automatically adapt the parameters of the GA according to what is happening during the simulation. Therefore, whatever would be the values of the parameters, they would be changed all along the simulation to optimize the exploration and the exploitation of the search space. Davies (1991) introduced the discussion on the choice of operator parameters and on how to avoid the hand tuning of their values. The principle consists of changing the probabilities according to their contribution in building good solutions. This method has been (successfully) applied by Notredame and Higgins (1996) in a GA dedicated to multiple alignment.

4.1. *Self-adaptation of genetic operators*

We chose to work on self-adaptation of genetic operators, that is the possibility of adapting the probability of occurrence of the operator during the evolutionary process. We have developed a method allowing the exploration/exploitation ratio to be modified by changing the probabilities of the operators according to their potential in generating new chromosomes. The method has been applied to the discovery of motifs in biological sequences (Hernandez et al., 2002). Each operator is characterized by a simple redundancy score defined as 1 – (x/y), where y is the number of chromosomes on which the operator is applied, and x is the number of new chromosomes – never seen before across the whole simulation – generated by the operator. During the evolutionary process, goal redundancy values are fixed. The probability of applying each operator is adapted so that the operator redundancy converges toward the goal redundancy. Changing this goal redundancy therefore helped us in determining and changing the general direction of the GA. It led to slightly better performances than leaving the probabilities constant, but the computing time was also slightly increased.

4.2. *Representation of the explored space to optimize exploration*

We wish to study an improved way of exploiting building blocks properties of cooperative metaheuristics, and more specifically of GA. GA are believed to be successful mostly because they are manipulating short, highly fit subcomponents called building blocks (BB). BB are identified and assembled by the GA, and are recombined to form high performance solutions (Goldberg, 2002; Holland, 1975). BB are in competition, and the best ones take the advantage among the population. Some problems are hard for this BB concept, i.e., they are a challenge for the recombination process. Such problems are called deceptive, because they mislead the GA by imposing badly-fit, low order BB to be assembled in well-fit, higher order BB. An example of such problems is the Royal Road functions (Jones, 1993) (RRf).

RRf are multimodal and deceptive, which means that the global solution is neighbored by bad solutions, making the search harder than for more classical functions. The chromosomes are usually bit strings of 0 s and 1 s. To find the global solution (1 s in every position), the GA has to assemble BB that contributes to the fitness of the chromosome by an amount depending on the content in 1 s within BB. These functions are, by their multimodal properties, ideal to search for the different combinations of BB. Even if their use as test functions for GA has been discussed (Mitchell et al., 1995), they remain in our case a good challenge.

We define a representation of the explored space in order to localize BB and to guide the exploration process in promising areas of the search space by "wise" (or at least less stochastically driven) genetic operators. This representation is a tree structure, and every chromosome is stocked in this tree. By investigating its characteristics (like the number of times a certain part of a chromosome was seen in the simulation, or which part of the tree contains chromosomes with the highest fitness), it should allow the definition of a more accurate selection of parts of the sub-space that are more interesting at a certain time, according to a certain criterion.

Our first results show that even with a simple self-adaptation, both the number of evaluations and the computing time required to find the optimum are improved over classical deceptive functions (Pelikan et al., 1999). The results were obtained on an average in 100 independent runs. The simulations were stopped when the optimum was found. The size of the chromosomes was increasing from 30 to 90 bits and the various complexities were computed from regression curves. For the 3-deceptive function the complexity for the number of evaluations decreased from $O(n^{2.8})$ up to $O(n^{2.2})$, and the time complexity decreased from $O(n^4)$ up to $O(n^{3.2})$. For the 5-trap function the complexity for the number of evaluations decreased from $O(n^{3.1})$ up to $O(n^{2.3})$, and the time complexity decreased from $O(n^{4.3})$ up to $O(n^{3.3})$.

5. Conclusion

We presented in this paper a new classification of metaheuristics considering the property of cooperation between entities exploring the search space. This classification involves three sub-classes: centralized, individual and concerted cooperation depending on how the cooperation is accomplished. We gave new definitions and described new developments of these approaches, and tried to show their individual benefits. We emphasized the importance of a combination of these methods to maximally profit of their specific characteristics for complex structured problems. In this case, a hierarchical division of the problem into independent tasks can lead to spread and simplify optimization steps. The results of these optimizations are then associated to produce a global solution benefiting from the structures appearing in the different levels considered.

We applied all these techniques to two central problems of proteomics: automatic protein identification and multiple sequence alignment based on motif inference. We gave a short overview of the state of the art of these problems and some possible improvements to manage all their inherent difficulties. We presented our metaheuristic approaches for each proteomic problem taking into account a maximum of these difficulties. We gave also

some promising preliminary results obtained with our tools using real data for protein identification, learning of a discriminating score, motif inference or sequence clustering.

We are currently working on several enhancements of our methods. From the application point of view, we try to integrate some new biological expert knowledge in the parsing of the graph structure and in the protein identification scoring function. We also designed a new algorithm based on tag matching which will allow the detection of all possible modifications. We work on the extension of motif representation allowing variable length, gaps inside motif or multiple alphabet representation. A wide variety of improvements are also currently under study for the metaheuristic aspects. We work on a better management of the building block concept in swarm intelligence and in evolutionary algorithms. For example, if we can represent explicitly the possible building blocks in genetic programming then we can use a multi level parallel algorithm to discover and combine them. We will extend our explored space representation to control the use of the genetic operators and to define smarter new operators for genetic algorithms. We will develop adaptive strategies for agents that enable a mutual agreement and increase the diversification aspect of concerted cooperation using new operators like birth or death of agents. Finally, we plan to integrate a multi objective optimization approach (Coello Coello et al., 2002) to our methods because most of the biological problems imply several independent properties with inconsistent maximum values.

Acknowledgments

The authors wish to thank Frederique Lisacek, Markus Müller and Rumen Andonov for all helpful discussions about these works and Alexander Scherl and Pierre-Alain Binz for their useful contributions.

References

Akutsu, T., Arimura, H. & Shimozono, S. (2000). On Approximation Algorithms for Local Multiple Alignment. *Proceeding 4th Int. Conf. Computational Molecular Biology*, 1–7. Ref Type: Conference Proceeding.

Bailey, T. L. & Elkan, C. (1994). Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Biopolymers. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28–36, AAAI Press. Ref Type: Conference Proceeding.

Blickle, T. & Thiele, L. (1995). *A Comparison of Selection Schemes Used in Genetic Algorithms*. TIK 11. Ref Type: Report.

Bonabeau, E., Dorigo, M. & Theraulaz, G. (2002). *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press.

Buhler, J. & Tompa, M. (2002). Finding Motifs Using Random Projection. *J. Comput. Biol.* **9**: 225–242.

Califano, A. (2000). SPLASH: Structural Pattern Localization Analysis by Sequential Histograms. *Bioinformatics* **16**: 341–357.

Chen, T., Kao, M. Y., Tepel, M., Rush, J. & Church, G. M. (2001). A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *J. Comput. Biol.* **8**(3): 325–337.

Coello Coello, C. A., Veldhuizen, D. A. V. & Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publisher.

Dancik, V., Addona, T., Clauser, K., Vath, J. & Pevzner, P. A. (1999). De Novo Peptide Sequencing via Tandem Mass Spectrometry. *J. Comput. Biol.* **6**: 327–342.

Davis, L. (1991). *Handbook of Genetic Algorithm*. New York: Van Nostrand Reinhold.

Dorigo, M. & Di Caro, G. (1999). *The Ant Colony Optimization Meta-Heuristic*, ch. 2.

Feng, D. F. & Doolittle, R. F. (1987). Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees. *J. Mol. Evol.* **25**: 351–360.

Fernandez, F., Tomassini, M., Punch, III W. F. & Sanchez, J. M. (2000). Experimental Study of Multipopulation Parallel Genetic Programming. *Proceedings of the Third European Conference on Genetic Programming*, 283–293, Springer Verlag. Ref Type: Conference Proceeding.

Fukami-Kobayashi, K., Schreiber, D. R. & Benner, S. A. (2002). Detecting Compensatory Covariation Signals in Protein Evolution Using Reconstructed Ancestral Sequences. *J. Mol. Biol.* **319**: 729–743.

Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers.

Golubski, W. (2002). *Genetic Programming: A Parallel Approach.* Lecture notes in computer science, vol. 2311.

Gotoh, O. (1996). Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinement as Assessed by Reference to Structural Alignments. *J. Mol. Biol.* **264**: 823–838.

Gras, R., Gasteiger, E., Chopard, B., Müller M. & Appel, R. D. (2001). *New Learning Method to Improving Protein Identification from Peptide Mass Fingerprinting, 2000*. 4th Siena 2D electrophoresis meeting. Ref Type: Conference Proceeding.

Gras, R. & Muller, M. (2001). Computational Aspects of Protein Identification by Mass Spectrometry. *Current Opinion in Molecular Therapeutics* **3**: 526–532.

Gras, R., Muller, M., Gasteiger, E., Gay, S., Binz, P. A., Bienvenut, W., Hoogland, C., Sanchez, J. C., Bairoch, A., Hochstrasser, D. F. & Appel, R. D. (1999). Improving Protein Identification from Peptide Mass Fingerprinting Through a Parameterized Multi-level Scoring Algorithm and an Optimized Peak Detection. *Electrophoresis* **20**: 3535–3550.

Hernandez, D., Gras, R., Lisacek, F. & Appel, R. D. (2002). *MoDEL: Inférence de motifs avec un algorithme évolutionniste, JOBIM 2002*, 265–267. Ref Type: Conference Proceeding.

Hernandez, P., Gras, R., Frey, J. & Appel, R. D. (2002). *Automated Protein Identification from Tandem Mass Spectrometric Data Using Ant Colony Optimization Algorithms*, 148–150. Proteomics in press, 5th Sienna meeting. Ref Type: Conference Proceeding.

Hertz, G. Z. & Stormo, G. D. (1999). Identifying DNA and Protein Pattern with Statistically Significant Alignment of Multiple Ssequence. *Bioinformatics* **15**: 563–577.

Higgins, D. G. & Sharp, P. M. (1989). Fast and Sensitive Multiple Sequence Alignments on a Microcomputer. *Comput. Appl. Biosci.* **5**: 151–153.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press.

Jonassen, I., Collins, J. F. & Higgins, D. G. (1995). Finding Flexible Patterns in Unaligned Protein Sequences. *Protein Science* **4**: 1587–1595.

Jones, T. (1993). *A Description of Holland's Royal Road Functions. 5th International Conference on Genetic Algorithms*. Ref Type: Conference Proceeding.

Keich, U. & Pevzner, P. A. (2002). Finding Motifs in the Twilight Zone. *Proc. 6th Int. Conf. Computational Molecular Biology*, 195–204. Ref Type: Conference Proceeding.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press.

Lassmann, T. and Sonnhammer, E. L. (2002). Quality Assessment of Multiple Alignment Programs. *FEBS Lett.* **529**: 126–130.

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. & Wootton, J. C. (1993). Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment. *Science* **262**: 214.

Lin, S. C., Punch, III W. F. & Goodman, D. (1994). *Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach*, 28–37. Sixth IEEE parallel and distributed processing. Ref Type: Conference Proceeding.

Mann, M. & Wilm, M. (1994). Error-tolerant Identification of Peptides in Sequence Databases by Peptide Sequence Tags. *Anal. Chem.* **66**: 4390–4399.

Marsan, L. & Sagot, M.-F. (2000). Extracting Structured Motifs Using a Suffix Tree – Algorithms and Application to Consensus Identification. In Minoru, S. & Shamir, R. (eds.) *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB 00)*, 210–219. Tokyo, Japan: ACM Press. Ref Type: Conference Proceeding.

Michalewicz, Z. & Fogel, D. (2000). *How to Solve It: Modern Heuristics*. Springer-Verlag.

Mitchell, M., Holland, J. H. & Forrest, S. (1995). *When Will a Genetic Algorithm Outperform Hill Climbing?* Morgan Kaufmann.

Morgenstern, B. (1999). DIALIGN 2: Improvement of the Segment-to-Segment Approach to Multiple Sequence Alignment. *Bioinformatics.* **15**: 211–218.

Mullan, L. J. (2002). Multiple Sequence Alignment – the Gateway to Further Analysis. *Brief. Bioinform.* **3**: 303–305.

Notredame, C. & Higgins, D. G. (1996). Sequence Alignment by Genetic Algorithm. *Nucleic Acids Res.* **24**: 1515–1524.

Notredame, C., Higgins, D. G. & Heringa J. (2000). T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. *J. Mol. Biol.* **302**: 205–217.

Pacheco, P. S. (1997). *Parallel Programming with MPI*. San Francisco: Morgan Kaufmann.

Pelikan, M., Goldberg, D. E. & Cantu-Paz, E. (1999). BOA: The Bayesian Optimization Algorithm. I. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, 525–532. San Francisco, CA: Morgan Kaufmann. Ref Type: Conference Proceeding.

Pennington, S. R. & Dunn, M. J. (2001). *Proteomics from Protein Sequence to Function*. BIOS Scientific.

Pevzner, P. A., Mulyukov, Z., Dancik, V. & Tang, C. L. (2001). Efficiency of Database Search for Identification of Mutated and Modified Proteins via Mass Spectrometry. *Genome Research* **11**: 290–299.

Pevzner, P. A. & Sze, S.-H. (2000). Combinatorial Approaches to Finding Subtle Signals in DNA Sequences. *Proceedings of the eighth International Conference on Intelligent Systems for Molecular Biology*, 269–278, San Diego. Ref Type: Conference Proceeding.

Punch, W. F. (1998). *How Effective are Multiple Populations in Genetic Programming. Genetic Programming 1998*, 308–313. Ref Type: Conference Proceeding.

Rigden, D. J. (2002). Use of Covariance Analysis for the Prediction of Structural Domain Boundaries from Multiple Protein Sequence Alignments. *Protein Eng.* **15**: 65–77.

Scherl, A., Coute, Y., Deon, C., Calle, A., Kindbeiter, K., Sanchez, J.-C., Greco, A., Hochstrasser, D. F. & Diaz, J. J. (2002). Functional Proteomic Analysis of the Human Nucleolus. *Mol. Biol. Cell*, published online.

Schlosser, A. & Lehmann, W. D. (2002). Patchwork Peptide Sequencing: Extraction of Sequence Information from Accurate Mass Data of Peptide Tandem Mass Spectra Recorded at High Resolution. *Proteomics* **2**: 524–533.

Stoye, J. (1998). Multiple Sequence Alignment with the Divide-and-Conquer Method. *Gene* **211**: GC45–GC56.

Taylor, J. A. & Johnson, R. S. (1997). Sequence Database Searches via de Novo Peptide Sequencing by Tandem Mass Spectrometry. *Rapid Commun Mass Spectrom* **11**: 1067–1075.

Thompson, J. D., Higgins, D. G. & Gibson, T. J. (1994). CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice. *Nucleic Acids Res.* **22**: 4673–4680.

Wilkins, M. R., Williams, K. L., Appel, R. D. & Hochstrasser, D. F. (1997). *Proteome Research: New Frontiers in Functional Genomics*. Springer-Verlag.

Yagiura, M. & Ibaraki, T. (2001). On Metaheuristic Algorithms for Combinatorial Optimization Problems. *Systems and Computers in Japan* **32**: 33–55.