Expanding the Current Boundaries of Nature-based Modeling and Computing: *Chem-Inspiration for Meta-Heuristics.*

Yovani Marrero-Ponce,[β]✉ Yasser B. Ruiz-Blanco,[£] Yuviny Echevarría,[§] Felix Martinez-Rios,[¥] Rafael Bello,[#] Carlos Morell,[#] & Ricardo Vivas-Reyes.[Ωπ]

[β]*Universidad San Francisco de Quito (USFQ), Grupo de Medicina Molecular y Traslacional (MeM&T), Colegio de Ciencias de la Salud (COCSA), Escuela de Medicina, Edificio de Especialidades Médicas, Av. Interoceánica Km 12 ½ —Cumbayá, Quito 170157, Ecuador.* https://orcid.org/0000-0003-2721-1142

[£]*Computational Biochemistry, Center of Medical Biotechnology, University of Duisburg-Essen, Essen 45141, Germany.* orcid.org/0000-0001-5400-4427.

[§]*Department of Informatics, University of Cienfuegos, Cuba.*

[¥]*Universidad Panamericana, Facultad de Ingeniería, Ciudad de México, México.* http://orcid.org/0000-0002-2346-5922

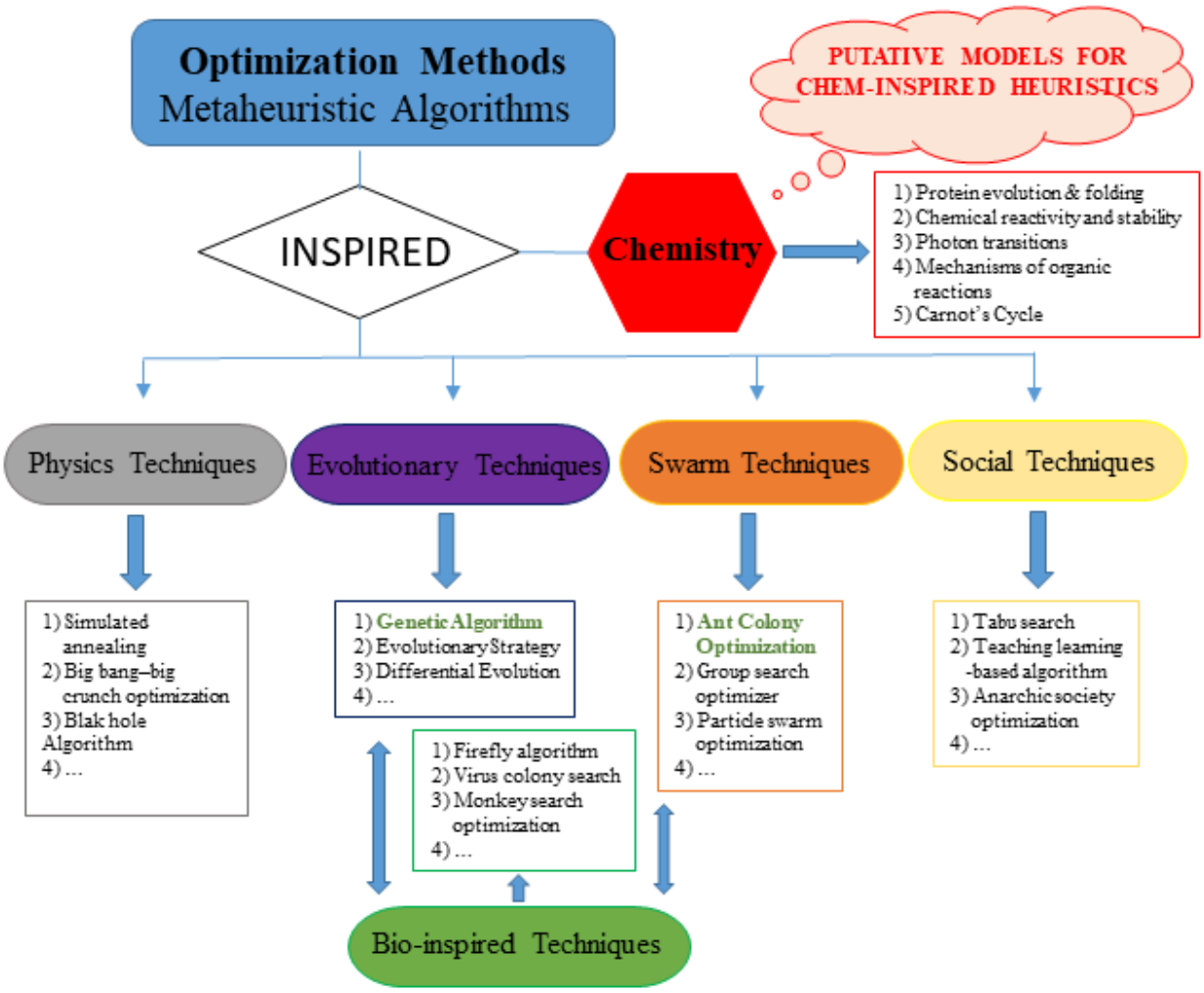
[#]*Department of Computer Science, Universidad Central de Las Villas, Santa Clara, Villa Clara, Cuba.*

[Ω]*Grupo CipTec, Fundacion Universitaria Tecnologico de Comfenalco, Facultad de Ingenierias, Fundacion Universitaria Tecnologico Comfenalco – Cartagena, Cr 44 D N 30A, 91, Cartagena, Bolivar, Colombia.*

[π]*Group of Quantum and Theoretical Chemistry, Faculty of Exacts and Naturals Sciences, University of Cartagena, Cartagena de Indias, Bolívar, 130001, Colombia.*

Corresponding authors (✉):
Y. Marrero-Ponce: ymarrero@usfq.edu.ec or ymarrero77@yahoo.es; **Tel.:** +593-2-297-1700 (ext. 4021). http://www.uv.es/yoma/ or http://ymponce.googlepages.com/home**; ORCID ID:** http://www.orcid.org/0000-0003-2721-1142.

TOC Graphic


Optimization Methods
Metaheuristic Algorithms
INSPIRED
Chemistry
PUTATIVE MODELS FOR CHEM-INSPIRED HEURISTICS
1) Protein evolution & folding
2) Chemical reactivity and stability
3) Photon transitions
4) Mechanisms of organic reactions
5) Carnot's Cycle
Physics Techniques
Evolutionary Techniques
Swarm Techniques
Social Techniques
1) Simulated annealing
2) Big bang–big crunch optimization
3) Blak hole Algorithm
4) ...
1) Genetic Algorithm
2) Evolutionary Strategy
3) Differential Evolution
4) ...
1) Firefly algorithm
2) Virus colony search
3) Monkey search optimization
4) ...
1) Ant Colony Optimization
2) Group search optimizer
3) Particle swarm optimization
4) ...
1) Tabu search
2) Teaching learning -based algorithm
3) Anarchic society optimization
4) ...
Bio-inspired Techniques

ABSTRACT

High-throughput methods in science have created a trend to generate massive amount of data that challenge our ability to mine and search through massive information spaces. Thus more efficient and effective solutions for data analysis and optimization are required continuously. The best solutions for many problems-solving approaches in science could have many sources of inspiration coming from diverse natural phenomena. In this context, most Artificial Intelligence (AI) approaches benefit from emulation natural processes for their information processing strategy. Among the AI protocols, meta-heuristic algorithms for learning model and optimization have exploited a number of biological phenomena leading to highly effective search and learning engines. Examples of these processes are the ant colony organization, brain function and genetics among others. The evolution has turned all these biological events in highly efficient procedures, whose basics principles have then provided an excellent ground of new computational algorithms The aim of this report is pave the way to a new class of nature-based meta-heuristic methods which shall be based on diverse chemical and biomolecular systems. We present five examples from different subjects of Chemistry like Organic Chemistry, Chemical Physics and Biomolecules; and introduce how computational models could be inferred from them. Besides, we develop one of these models, in detail, which is based on protein evolution and folding principles. We consider that the wealth of systems and processes related to Chemistry, as those described in the present communication, might boost the development of relevant meta-heuristic and classification algorithms in upcoming years.

Keywords: Natural phenomena, Chemistry process, Computational algorithm, Artificial intelligence, Learning engine, Meta-heuristics, Protein folding.

Running Head: *Nature-based Modeling and Computing: Chem-Inspiration for Metaheuristics...*

1. INTRODUCTION

Complex optimization problems demand efficiency and robustness optimization algorithms.[1] Many researches in areas of Artificial Intelligence (AI), Operations Research and Engineering have been increasingly interested in this kind of problems. Besides, the field represents an attractive scientific challenge because it implies working under the paradigm of identifying a simple approach to achieve a complex solution.[2]

The word *heuristic* comes from the Greek term "heuriskein" that means "to find". The first time the term *heuristic* appears was by Polya 1957 in his classic report "How to Solve It?".[3] The heuristic methods became central in the solution of complex problems since the beginning of the research in AI.[4] Later, Simon defined a heuristic approach as a process that can solve a problem without guaranteeing the exact solution.[5] A modern definition was presented by Reeves as: "A heuristic technique (or simply heuristic) is a method which seeks good (*i.e.* near-optimal) solutions at a reasonable computation cost without being able to guarantee optimality, and possibly not feasibility".[6] Consequently, the success of heuristic algorithms depend integrally to the specific problem that is being solve.[7] This issue is expressed in a relevant theorem published by Wolpert and Macready in 90´s decade, which mention that *do not exist the heuristic method that could solve any kind of optimization problem with better solutions than the other method.*[8] This theorem constitutes the main driving force for the development of new and more efficient heuristics algorithms.

The term *meta-heuristics* was used by Glover, in 1986,[9] making reference to a *master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality*. *The heuristics guided by such a meta-strategy*

may be high-level procedures or a description of possible transformations that turn one solution into another, together with an associated evaluation rule.

The meta-heuristics techniques have applications in wide range of computer models. Among the most relevant applications, there are the design of chemotherapy schedules, optimization of bio-molecular systems, medical decision support and software engineering, among others. Significant examples of applications can be seen in references.[10, 11]

An important perspective in AI is the inspiration for the core information processing strategy of a given technique.[12] Computers can only do what humans instruct; the main ideas could come from a diversity of sources. Many computational models have inspiration in Biology, Physics or Social phenomena. Natural Computing is an interdisciplinary field concerned with the relationship of computation and biology, see for details.[13, 14]

A field of relevant nature-based inspirations is the meta-heuristic algorithms for model learning and optimization.[15, 16] Two important groups of meta-heuristic algorithms are the Evolutionary Computing (EC) and the Swarm Intelligence (SI). On one hand, EC refers to the homologous features of several computational algorithms with the natural-evolution process. On the other hand, the main idea of SI is the search for the optimal value of a given problem by sharing historical and social information between agents. The main representatives algorithms in SI are Particle Swarm Optimization[17, 18] and Ant Colony Optimization.[19] The popular Genetic Algorithms [20, 21] is an important member of EC with Evolution Strategies,[22] Evolutionary Programming [23] and Genetic Programming.[24] A relevant introduction to EC is presented in the review of Eiben and Smith.[25, 26]

For both EC- and SI-based strategies, the inspiration sources of novel nature-based algorithms do not stop in biology. The most recognized physic-based algorithm is Simulated

Annealing (SA).[27] Along the last two decades, this technique has shown the success of the paradigm of a simple nature-inspired computational model to solve large optimization problems.

Two studies have entered in what it is, to the best of our knowledge, an unexplored source of models for novel meta-heuristics methods, *i.e.* using a chemical process as background inspiration.[28, 29] The authors, Alatas and Lam *et al*., introduce the use of chemical reactions to develop new a new meta-heuristic approach.

Chemistry is a broad field of science interfacing Physics and Biology; therefore, it accounts with the wealth and diversity of the entire atomic and nano-scaled world. Chemical systems are naturally optimized; actually, the concept of *optimization* in chemistry can be evidenced in dissimilar forms. For instance, there is *chemical optimization* in the form of thermal equilibrium in the Brownian motion of particles in gas and liquid phases, as well as it is also present in high symmetry lattices of crystals. Despite the suitability of several chemical systems for serving as key principles of heuristic algorithms in AI, they have not received the same attention as biological processes. Consequently, the main aim of this report is to shed light into several chemical process and systems that can be suitable sources of inspiration for new meta-heuristics algorithm.

2. BACKGROUNDS

An optimization process intends to find an optimal solution to a particular problem where there are a wide solutions space and exists a discrimination criterion that can be used as *objective function* to assess each independent solution. The value of the variables could be affected for some restrictions.

Optimization problems can be determined by decision variables with discrete domains, which are called discrete optimization problems; and can be modeled generally as a search through a large, but finite number of combinatorial solutions. On the other hand, continuum optimization problems are a class where the decision variables possess an infinite domain of real numbers.

Optimization algorithms can be classified into two categories: *gradient-based methods*, in which derivative vectors are used to drive the search process, and *direct-search methods*, in which only the *objective function* values are used to drive the search.[30] Furthermore, the number of solutions stored in every step of the algorithm provide one of the most relevant features of a meta-heuristic method, *i.e. sequential or population-based algorithms*. The first methods work with one solution in every stage of the algorithm while the second use successive sets of putative solutions respectively. Some heuristics can work with a multi-objective framework based on the use of more than one decision variable. Multi-objective approach must keep out solutions with high performance in at least one of the objectives, but with poor performance in the rest of the objectives.[31]

The population-based methods show some advantages: a) continuous sampling of information from diverse regions in search space, b) the variance of the population provides good information about the extent of the potential search region [30]. There have been proposed four items in which population-based meta-heuristics can be tuned in order to increase their performance: (i) the selection of a subset of solutions from the population, (ii) the generation of new solutions using the selected subset, (iii) choosing inferior or spurious solutions for replacement, and (iv) updating the solution base with new or old solutions. Most of the differences among the meta-heuristic approaches relay on the second item, whose operators can

Table 1. Optimization Methods (OMs).

Method	Ref.
Swarm based OMs	
Artificial fish-swarm algorithm	[32-34]
Bacterial foraging optimization	[35-37]
Bat algorithm	[38, 39]
Bee colony optimization	[40]
Cat swarm optimization	[41, 42]
Firefly algorithm	[43]
Glowworm swarm optimization	[44-46]
Krill Herd algorithm	[47]
Monkey search optimization	[48, 49]
Cuckoo search	[50]
Spotted hyena optimizer	[51]
Lion optimization algorithm	[52]
Group search optimizer	[53]
Dolphin echolocation algorithm	[54]
Biogeography-based optimization	[55]
Whale optimization algorithm	[56]
Shark smell optimization	[57]
Salp swarm algorithm	[58]
Moth-flame optimization algorithm	[59]
Dragonfly algorithm	[60]
Crow search algorithm	[61]
Ant Lion optimizer	[62]
Flower pollination algorithm	[63]
Symbiotic organisms search	[64]
Swine influenza models based optimization	[65]
Virulence optimization algorithm	[66]
Virus colony search	[67]
CSA: based on chemotherapy science	[68]
Cuttlefish algorithm	[69]
Physics based OMs	
Artificial physics optimization algorithm	[70, 71]
Big bang–big crunch optimization	[72, 73]
Central force optimization	[74, 75]
Charged system search	[76]
Electromagnetism-like algorithm	[77, 78]
Galaxy-based search algorithm	[79]
Gases Brownian motion optimization	[80]
Gravitational search algorithm	[81, 82]
Grenade explosion method	[83]
Intelligent water drops algorithm	[84, 85]
Particle collision algorithm	[86]
Ray optimization	[87]
Variable mesh optimization	[88, 89]
Spiral dynamics inspired optimization	[90]
Thermal exchange optimization	[91]
Vortex search algorithm	[92]
Water cycle algorithm	[93]
Water wave optimization	[94]
Simulated annealing	[95]
Rain-fall optimization algorithm	[96]

Optics inspired optimization	[97]
Ions motion algorithm	[98]
Hydrological cycle algorithm	[99]
Fractal-based algorithm	[100]
Colliding bodies optimization	[101]
Charged system search	[76]
Atmosphere clouds model optimization	[102]
Social based OMs	
Imperialist competitive algorithm	[103]
Parliamentary optimization algorithm	[104]
Teaching learning-based algorithm	[105]
Anarchic society optimization	[106]
Shuffled frog leaping algorithm	[107]
Fish school search	[108]
Sperm whale algorithm	[109]
Grey wolf optimizer	[110]
Brain storm optimization algorithm	[111]
Cohort intelligence algorithm	[112]
Tabu search	[9, 113]

follow dissimilar background information. The Table **1** summarizes several meta-heuristic algorithms organized by their inspiration sources.

Over the last years, a significant number of hybridization between traditional meta-heuristic algorithms were reported. The main motivation behind the hybrid algorithm is to exploit the complementary characteristic of different optimization strategies [114]. Moreover, meta-heuristic algorithms, especially Genetic Algorithms, are key-part many soft-computing learning approaches to solve non-lineal problems, *i.e.* the Genetical-Fuzzy Systems.[115]

Recently a new problem-solving approach named hyper-heuristics (HH) has emerged. The main idea of this approach is the automatic design and adaptation of heuristic methods in order to solve hard computational search problems. About a decade ago HH were showed up as *heuristics to choose heuristics* in the context of combinatorial optimization. Burke as proposed a modern definition of the term: "A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems" [116]. The earlier point of view refers to methodologies for selecting *existing* heuristics. The second includes the heuristic

generation of *new* heuristics from components or pieces of others. Several studies have shown the application of HH in real optimization problems.[117-121]

In spite of the advances in nature-inspired heuristics, it is surprising that chemistry has not been so far a relevant source of models to design meta-heuristic approaches. The Chemical Reaction Optimization (CRO)[29] and the Artificial Chemical Reaction Optimization Algorithm (ACROA)[28, 122] are, to the best of our knowledge, the pioneers in a potentially promising kind of *chem-inspired meta-huristics*. The CRO has been used in some problems such as distribution network reconfiguration,[123] grid scheduling problem,[124] job shop scheduling problem[125] and stock portfolio selection.[126] Original CRO has been shown to perform well in many optimization problems of discrete and finite search space; although an update of this algorithms, called Real-code chemical reaction optimization (RCCRO), have been proposed intended to continuum optimization problems.

Although these chemical models for meta-heuristics have definitely opened a door to a next generation of approaches in AI, chemical inspiration remains limited. In the next sections, we intend to fill this vacuum by, first, introducing and describing a new meta-heuristic model based on one of the most popular open problems in Chemistry: *the protein-folding problem*. Then, four other chemical processes are presented by sketching their possible application for generating new meta-heuristic algorithms for artificial intelligence.

3. NOVEL MODEL INSPIRED IN PROTEIN EVOLUTION & FOLDING

Proteomics is the part of molecular biology intended for the study of proteins. The central dogma of this science suggests that the sequence of a protein determines its structure and this in turn determines its function.[127] Following this paradigm, it can be understood that the native state

of a protein is not only the optimum state given an amino acid sequence; but also within the space of sequences, the observed sequence is optimum for the structure. These two optimization problems are known as direct- and reverse-protein-folding problems.

There are several models that explain the protein folding process, examples of these are the hydrophobic collapse, the Diffusion – Collision[128] and Nucleation – Condensation[129, 130] models, and the theory of the free energy landscape;[131, 132] which is accepted as the more general model to describe the process. All these approaches have in common that the hydrophobic amino acids are occluded inside the protein, whereas polar residues remain mostly exposed on the surface. This general trend has been modeled using a simple model with only two kinds of amino acids, hydrophobic (H) and polar (P).[132-134] Here, we develop a computational model that employs a heuristic framework based on these basic principles of protein folding.

3.1 Model Definition

A problem of *n* variables, where all of them share the same domain, is represented by **protein** with a **sequence** of *n* amino acids, or **residues**, aa_1, aa_2, …, aa_n, that represent the variables. Every residue (aa_i) possess an associated classification as either **hydrophobic** (**H)** or **polar (P)**, which is randomly assigned initially. A **conformation** corresponds to a vector of length ***n*** where each aa_i adopts a value within its domain. Depending on the values of the amino acids (variables), each one is classified as **internal** (**I)** or **superficial (S)** as long as the **local density (ρ_i)** of aa_i is larger or lower than a cutoff value **ρ***, respectively. The parameter ρ_i represents the number of amino acids (variables) whose values are within the range: $aa_i \pm \boldsymbol{h}$, where ***h*** is a fixable parameter according the domain of the variables in the problem.

$$aa_i \begin{cases} I & \rho_i \geq \rho \\ S & otherwise \end{cases}$$

A **protein** is built as the combination of a **sequence** and its corresponding **conformation**, and represents a putative solution of the problem. In a protein, the attributes of an amino acid are: a value, a type (**H**, **P**) and a classification according its value (**I**, **S**), this leads to four possible classifications for a given amino acid (**HI**, **HS**, **PI**, **PS**), which depend on the value and the assigned type. The meta-heuristic algorithm described ahead operate by favoring the exchange of amino acids classified as **HS** and **PI** into **HI** and **PS**, in agreement with the general trend of a protein folding event. The final goal is increasing the **stability** of the protein, which is represented by the objective function of the problem.

In order to optimize the values of the variables (**conformation**) the algorithm uses auxiliary operators that acts over the type (**H** or **P**) of residues (**sequence)** and thus determine the subsequent change of the **conformation**.

3.1.1 Operators for sequences

Several biosynthesis process, mediated by peptidases and peptidiltransferases, determine the primary structure of a protein. These processes are used to search within the space of residue types (**H** or **P**) in a given sequence.

- **C-Peptidase**: This subtractive operator eliminates the last residue of a sequence.
- **N-Peptidase**: This subtractive operator eliminates the first residue of a sequence.
- **Peptidyltransferase-H**: This additive operator inserts a residue of type **H** to the end of a sequence.
- **Peptidyltransferase-P**: This additive operator inserts a residue of type **P** to the end of a sequence.
- **Circular Permutation**: this operator places the last *l* residues of a sequence to the first *l* positions of the chain.

Protein evolution is a process largely depending on Genomics, therefore this heuristic method is presented as a sort of hybrid heuristic that incorporate operators of GA to search among the space of **sequences** (distribution of amino acids of type H and P).

- **Mutagenesis**: This operator randomly changes the type (H or P) of any residue in the sequence.

Nonetheless is worth highlighting that the actual solutions of this heuristics are the **conformations** (vector of values of all the aa), the **sequence** is only used to determine how the **conformation** operators shall be applied.

This concept of using a heuristic method to tune the application of other heuristics over the solution vectors, it resembles the modern hyper-heuristics approaches. Such fact it highlights the potential of the natural processes involved in protein evolution and folding, as a promising framework to develop a new computational model like the one described ahead.

3.1.2 Operators for conformations

- **Folding Operator:** This operator changes the value of an amino acid aa_i, classified as **HS**, to a random value within the range: $aa_j \pm \boldsymbol{h}$, where the amino acid aa_j must have higher **local density** than aa_i. The trend of this operator is to transform HS residues to HI.
- **Unfolding Operator:** This operator changes the value of an amino acid aa_i, classified as **PI**, to a random value within the range: $aa_j \pm \boldsymbol{h}$, where the amino acid aa_j must have lower **local density** than aa_i. The trend of this operator is to transform PI residues to PS.
- **Nucleation Operator:** this operator changes the values of a group of amino acids of type **H**, into random values within the range: $aa_i \pm \boldsymbol{h}$, where aa_i should be the amino acid with larger **local density** in the selected group. The trend of this operator is to aggregate hydrophobic amino acids.

- **Random Movements:** The operator assign a random value to an amino acid classified either as **HS** or **PS**. This operator resembles the higher conformational freedom of residues on the surface of the protein.

A *sui-generis* operator is introduced based on the ancillary role of chaperone proteins in folding other chains. This operator, named **Chaperone operator**, modifies the classification as internal (**I**) or superficial (**S**) of each amino acid, but it does not affect the conformation directly, *i.e.* this operator does not vary the values of the variables. This operator shall work as follow: i) In an intermediate step of the algorithm the **protein P** is merged with another one that is called **chaperon**. ii) The arrays of residues from both proteins are concatenated to form a complex, represented as a new array with the double of the number of variables in a solution. iii) The resulting complex is used to re-compute the **local density** of each *aa* and re-assess their accessibility class (**I** or **S**). iv) The part of the complex that belong to the chaperone protein is then separated from **P**, obtaining an updated protein with the same values for all residues but with a different classification of **S** and **I** for every amino acid.

Consequently, the **Chaperone operator**, *per se*, do not create a new solution, but it modifies the classification of the residues, thus affecting the application of the conformational operators. The chaperone protein could be an external solution that is known to be close to an optimum solution.

The proposed model introduces some new features for search and optimization, the implementations of this approaches could exploit some ideas as linkage learning,[135] intensifications in important hyper planes in the search space and auto adaptive techniques with tuning on the fly.

3.2 Pseudo-code of the algorithm

Scheme 1. Description of the pseudo-code of the algorithm.

//Biosynthesis
Initialize the parameters: **h**, **d**, $\mathbf{p_f}$, $\mathbf{p_u}$, $\mathbf{p_b}$, **k**, **l**, **M**, $\mathbf{f_H}$ from input data;
Generate a protein **P** with a random sequence and a random conformation;
Evaluate the **objective function, OF,** in **P** to determinate its stability;
Compute the **local density** of each aa;
Classify each aa of **P** as internal **I** or superficial **S** (**local density []**, **d**);
Create a clone **P'** of **P**;
//End Biosynthesis
//Folding
Apply subroutine **Folding (P', $\mathbf{p_f}$, $\mathbf{p_u}$, $\mathbf{p_n}$, $\mathbf{p_b}$, k)** by applying conformation operators;
//End folding
While (**current cycle < M**)
    //Start Evolution
    Repeat
        Generate a random number **m** in the range [1, **number of aa**);
        Select, randomly, a fragment **O** of **P'** with length **m**;
        Select, randomly, a number **n** among [0, 1, 2, 3, 4, 5];
        Switch (**n**) *//One of the next options is selected*
                case 0: Apply the **Circular Permutation** operator on **O**;
                case 1: Apply the **C-peptidase** operator on **O**;
                case 2: Apply the **N-peptidase** operator on **O**;
                case 3: Apply the **Peptidyltransferase-H** operator on **O**;
                case 4: Apply the **Peptidyltransferase-P** operator on **O**;
                case 5: Apply the **Gene Mutation** operator on **O**;
        End Switch
    Until (**l** iterations)
    //End evolution
    //Folding
    Apply subroutine **Folding (P', $\mathbf{p_f}$, $\mathbf{p_u}$, $\mathbf{p_n}$, $\mathbf{p_b}$, k)**;
    //End folding
    //Application of Chaperone protein
    Generate a random number **r** in the range [0,1);
    If (**r** $\leq$ 0.5) apply the **chaperone operator** on **P'**;
    //End application of Chaperone protein
        //Folding
        Apply subroutine **Folding (P', $\mathbf{p_f}$, $\mathbf{p_u}$, $\mathbf{p_n}$, $\mathbf{p_b}$, k)**;
        //End folding
End While

```
Subroutine Folding (P', pf, pu, pn, pb, k)
    Repeat
        Generate a random number r in the range [0,1);
        If (r ≤ pf) apply the folding operator on P';
          Else if (pf < r ≤ pf + pu)
                Apply the unfolding operator on P';
                Else if (pf + pu < r ≤ pf + pu + pn)
                      Apply the unfolding operator on P';
                      Else
                         Apply the random movement operator on P';
        Evaluate the OF of P';
        If (OF(P') is better than OF(P)) P = P';
              Else
                Generate a random number r in the range [0,1);
                If (r > pb) P = P';
    Until (k iterations)
End Subroutine
```

Table 2. Summary of the parameters of the code.

Parameter	*Description*
h	Cutoff of the neighborhood of each amino acid ($aa_i \pm$ **h**) to determine the local density.
d	Cutoff of local density to define Internal and Superficial amino acids
p_f	Probability of folding*
p_u	Probability of unfolding*
p_n	Probability of nucleation*
p_b	Probability of stepping back (After the evaluation of a given solution, the algorithm continues with it in spite of being worse than the best solution found to the moment).
M	Maximum number of cycles.
f_H	Fraction of aa of type H during the biosynthesis.
k	Number of folding iterations in each cycle.
l	Number of evolution iterations in each cycle.

*The sum of p_f, p_u, p_n must be equal or lower than 1.

In this section, we present the pseudo-code of the previously introduced optimization algorithm based on protein evolution and folding principles.

This method, in accordance with the protein folding process, is presented as a sequential algorithm. However, this can be transformed in a population-based heuristic by simply applying the conformation operators directly to the complex protein – chaperone, where the chaperone might be any other protein of the population. The possibility of transforming the heuristics based

on such simple and even natural consideration highlights the potential impact and applicability of the novel Nature-based heuristic method.

4. PUTATIVE MODELS FOR CHEM-INSPIRED HEURISTICS

Chemistry is a field science that gathers many dissimilar systems which vary among: organic, inorganic, atomic, molecular, crystalline and polymers. Additionally, it comprises a series of principles that rules the evolutions of atomic and molecular systems and processes. Ahead in the section, we describe basic principles of chemical systems that can serve as the bases for four computational models of novel chem-inspired heuristics.

4.1 Chemical reactivity principles

Chemical reactivity and stability have been defined in terms of several magnitudes such as: entropy, free energy (or energy for very small systems), chemical potential, electronegativity equalization, polarizability and chemical hardness. The maximum harness (MHP) [136, 137] and minimum polarizability (MPP) [138] principles are rather equivalent concepts that generalize the rules of reactivity among organic compounds. Similarly, the electronegativity (or chemical potential) equalization principle (EEP) is another reactivity tendency.[138] A stable system, should achieve an equivalent chemical potential throughout the molecule, and specifically between bonded atoms. That is, there should not have a net forces to transfer electron density from one point to another.

Chemical hardness (η) and polarizability are conversely related and can be estimated by means of the difference between the energies of the Highest Occupied Molecular Orbital (HOMO) and the Lowest Unoccupied Molecular Orbital (LUMO), $\eta = (HOMO - LUMO) / 2$. The lower de energy gap, between HOMO and LUMO, the lower is the chemical hardness and

higher is the polarizability. The chemical potential is considered to be the opposite of absolute (Mulliken) electronegativity and can be approximated by $\mu = -(HOMO + LUMO) / 2$, equivalent to the mean energy of both orbitals.

In order to define, on the basis of the above reactivity principles, a computational heuristic model to search a space of N variables for an optimal solution, could be considered the next homologations: i) Be the energy of the system defined by the objective function of the problem. ii) Be the pair HOMO-LUMO a couple of variables selected in a given way (for instance they could be the pair of variables with minimum difference between their values among a random sample of the problem's variables). Therefore, an operator based on the MHP could modify the values of a selected pair of variables in order to increase the difference between them. Another operator that can be defined is based on the EEP, which could modify the values of two pairs of variables such that their mean values (*i.e.* their chemical potentials) tend to be equalized. In conjunction, both operators could guide a meta-heuristic search through the solutions space of a given problem.

4.2 Photon transitions

Transitions in the electromagnetic spectrum vary in a wide range from radiofrequencies, which induce transitions between the nuclear spin states, to gamma radiations with high ionization power.

Transitions commonly obey symmetry restrictions according the states involved in the process. A quantum of energy either can be absorbed or emitted, leading to an excited state or a more stable one respectively. Eventually every excited system tends to return to its ground state spontaneously, this phenomenon in the context of UV-vis radiation is called fluorescence.

This knowledge may be used to define a computational model in which the range of variation of every variable could be divided into several bins using specified threshold or percentiles values.

The different types of transitions could determine the operators of this model, such that the value of a given variable can change to near, or distant, regions in its domain according the selected operator, *e.g.* gamma, UV-VIS, IR, microwaves, etc. The acceptance of a modification could be restricted by secondary functions that rule the permissibility of the transition. Ultimately, the energy of the system may be represented by the objective function of the problem. Consequently, all fluorescent-permitted transitions (*i.e.* those leading to a better value of the function) could be accepted, while a given rule might be follow to accept those transitions involving an absorption of energy. For instance, a bad solution might be accepted for a limited number of iterations, after which the variable could be re-settled to the best solution found previously.

4.3 Mechanisms of Organic Reactions

Most of the reactions of aliphatic-organic compounds can be detailed by means of five basic mechanisms called: *Substitution Nucleophilic Unimolecular* (**S_N1**), *Substitution Nucleophilic Bimolecular* (**S_N2**), *Elimination Unimolecular* (**E1**), *Elimination Bimolecular* (**E2**) and *Addition* (**A**). The **S_N1** is a two-step mechanism in which the substrate loses a group, named *leaving group*, and then a *nucleophile* attacks to the intermediate structure; in between, the intermediate structure may undertake arrangements of groups near to the position of the *leaving group*. The **S_N2**, on the contrary, is a concerted transformation in which the *leaving group* and the *nucleophile* participate in a single step. The **E1** is a reaction that shares the same first step of the **S_N1**, but in the second step, the intermediate suffers the elimination of other group (a

hydrogen) next to the position of the *leaving group* giving rise to the formation of a double bond between these positions. Arrangements are also possible in this mechanism. The **E2** mechanism is a one-step transformation where a *leaving group* and an adjacent hydrogen are eliminated at the same time forming a double bond. Finally, the *Addition* reactions are conversed processes to the eliminations and therefore as the result two groups are added to an existing unsaturation. In this reaction, the arrangements are not common but the additions may undertake in two ways: *syn-addition* (**syn-A**) and *anti-addition* (**anti-A**). In the **syn-A** reaction, both groups are added in the same face of the plane of a double bond, whereas in the **anti-A** the groups are inserted in opposite faces of the double bond.

Such mechanisms can become a framework to computational heuristic models considering the values of the variables of a given problem as the groups of the substrate molecule:

A **S_N1** operator could be defined in the following way: i) a variable v_i is selected, ii) a second variable v_j is selected from the same solution, iii) a rearrangement probability is evaluated to decide assigning the value of vj to v_i (substitution).

A **S_N2** operator could be defined such that the value of a selected variable v_i is replaced by the value v'_i from a different solution.

An **E1** operator may follow the next steps: the same steps i) to iii) of the **S_N1** are carried out, then iv) if the **S_N1** probability is not accepted, then find a near-neighbor variable of v_j otherwise find a near neighbor of v_i, v) either v_j or v_i and its near neighbor, are modified in such a way the two variables adopt the value of their mean (formation of an unsaturation).

An **E2** operator may be defined as follow: i) a variable v_i is selected, ii) find variable of v'_I from a different solution, iii) the two variables are modified by assigning the value of their mean to each of them (formation of an unsaturation).

A **syn-A** operator could be proposed as such: i) a variable v_i is selected, ii) find a near-neighbor variable of v_i, iii) the two variables are modified by assigning values between their current values and the extremes of the domain of the variables (addition). The variable with lower value moves to the low-end of the domain, while the variable with higher value moves towards the top of the domain (syn addition).

Similarly, an **anti-A** operator could be proposed as such: i) a variable v_i is selected, ii) find a near-neighbor variable of v_i, iii) the two variables are modified by assigning values between their current values and the extremes of the domain of the variables (addition). The variable with lower value moves to the top-end of the domain, while the variable with higher value moves towards the low-end of the domain (anti addition).

4.4 Carnot's Cycle:

The Carnot's cycle is a model of a thermal machine that works with an ideal gas doing reversible transformations between two thermal sources of different temperature. The cycle is composed by two isothermal processes and two adiabatic ones, which are organized in an intercalated scheme:

Step 1: Isothermal expansion. The gas receives an amount of heat and it uses this energy to exert an equivalent amount of work while the internal energy of the system remains invariant.

Step 2: Adiabatic expansion. The gas is expanded without exchange of heat but decreasing its temperature and consequently its internal energy as well.

Step 3: Isothermal compression. The gas loses an amount of heat but it receives an equivalent amount of work while the internal energy of the system remains invariant.

Step 4: Adiabatic compression. The gas is compressed without exchange of heat but increasing its temperature and consequently its internal energy as well.

This ideal process achieves the optimum efficiency among any thermal machine working between the same heat sources. Such optimal behavior highlights this system as a potential background to an efficient optimization algorithm.

A heuristic model based on Carnot's cycle could be defined by considering the variables of the problem as particles of the gas. The **range of values** of the variables could determine the **volume** of a system (set of variables). Besides, the **mean value** of the variables could be associated to the **temperature** of the system. Within a system, the **movements** of the gas' particles could be modeled as **permutations** of the values of variables. The four operators related to each of the steps of Carnot's cycle could be defined as follow:

1. **Isothermal expansion**: Given an even subset of variables, i) a number of permutations are carried out. ii) A random value *k* is subtracted from the half of the variables with lower values, while the same value *k* is added to the other half of the variables. This operator expands the range of values (volume) but it keeps the mean value (temperature) intact.
2. **Adiabatic expansion**: Given an even subset of variables, i) a number of permutations are carried out. ii) A random value *k* is subtracted from the half of the variables with lower values. This operator increases the range of values (volume) at the same time that decreases the mean value (temperature).

3. **Isothermal compression**: Given an even subset of variables, i) a number of permutations are carried out. ii) A random value *k*, which must be smaller than the half of the range, is subtracted from the half of the variables with higher values, while the same value *k* is added to the other half of the variables. This operator decreases the range of values (volume) but it keeps the mean value (temperature) intact.
4. **Adiabatic compression**: Given an even subset of variables, i) a number of permutations are carried out. ii) A random value *k*, which must be smaller than the half of the range, is added to the half of the variables with lower values. This operator decreases the range of values (volume) at the same time that increases the mean value (temperature).

5. CONCLUSIONS AND PERSPECTIVES

The present article presents our view nature-inspired meta-heuristic algorithms. These methods have found dissimilar applications proving the suitability of nature-like algorithms to find solution for highly complex and multivariate problems. We highlight the lack of method exploiting chemical principles to generate novel algorithmic solutions. Hence, there are proposed five ideas to fill the current vacuum in the definition of chem-inspired algorithms. The ways in which chemical principles can be extrapolated to computational protocols is sketched for these five systems. Specifically, based on principles of protein folding, we detail the definition of one computational model, and the corresponding operators, for a novel meta-heuristic approach for search and optimization. With this article we aim to call the attention of the computational biology and medicine, bioinformatics, and computer science community of a relatively unexplored but promising field that can result in a new generation of more potent algorithms for

find solutions for highly complex computational problems in both social and biological aspect of the life.

Acknowledgements: Yovani Marrero-Ponce thanks to the program International Investigator Invited for a postdoctoral fellowship to work at Valencia University in 2019-2020. **Yovani Marrero-Ponce** acknowledges the support from USFQ "Collaboration Grant 2019-2020 (Project ID16885).

***Conflict of Interest**:* The authors confirm that this article content has no conflict of interest.

Corresponding Authors' information: Professor ***Yovani Marrero-Ponce*** received the BS degree in *Pharmaceutical Sciences* (summa cum laude) from the Central University of Las Villas (UCLV), Santa Clara, Cuba, in **2001**, the MS degree in *Biochemistry* from Medical University "Dr. Serafin Ruiz-de Zarate Ruiz", Santa Clara, Cuba, in **2004**, and the PhD degree in *Chemistry* from Havana University, Havana City, Cuba, in **2005**. After post-doctoral fellowships at the ***University of Valencia***, Spain, he founded the Unit of Computer-Aided Molecular "Biosilico" Discovery and Bioinformatic Research (CAMD-BIR Unit, today is known as CAMD-BIR International Network) as a spin-off of the Department of Pharmacy at UCLV. At present, he is an Full Professor/Research of *Molecular Pharmacology* and *Pharmacotherapy* at the Universidad San Francisco de Quito (USFQ), and Head of "Grupo de Medicina Molecular y Traslacional (MeM&T)", Colegio de Ciencias de la Salud (COCSA), Escuela de Medicina, Edificio de Especialidades Médicas 170157, Pichincha, Ecuador. His research interests include molecular modelling and drug discovery, chem-bio-med-informatics, chemometrics, molecular

descriptor, chemogenomics, and mathematical, theoretical and computational chemistry. **ORCID ID:** http://orcid.org/0000-0003-2721-1142. **Scopus Author ID**: 55665599200. **ResearcherID**: H-5724-2011. **ResearchGate:** http://www.researchgate.net/profile/Yovani_Marrero-Ponce/, **Google scholar:** http://scholar.google.com/citations?user=rsbUYyEAAAAJ&hl=en, **Facebook:** http://www.facebook.com/ymarreroponce.

6. REFERENCES

1. Sabat, S. L.; Ali, L.; Udgata, S. K., Integrated Learning Particle Swarm Optimizer for global optimization. *Applied Soft Computing* **2011**, 11, 574-584.
2. Garfinkel, R. S. Motivation and Modelling. In *The Travelling Salesman Problem: A Guide Tour of Combinatorial Optimisation*, Lawler, E. L.; Lestra, J. K.; Kan, A. H. G. R.; Shmoys, D. B., Eds.; Wiley. Chichester: 1985, pp 17-36.
3. Polya, G., *How to Solve It*. Princeton University Press: 1957.
4. Russell, S. J.; Norvig, P., *Artificial Intelligent A Modern Approach*. Prentice-Hall ed.; Alan Apt: New Jersey, 2010; p 1132.
5. Simon, H. A., Experiments with a Heuristic Compiler. *Journal of the ACM* **1963**, 10, 493-506.
6. Reeves, C. R. Modem heuristic techniques. In *Modern Heuristic Search Methods*, Rayward-Smith, V. J.; Osman, I. H.; Reeves, C. R.; Smith, G. D., Eds.; Wiley: New York, 1996, pp 1-25.
7. Silver, E. A.; Vidal, R. V.; Werra, D. D., A Tutorial on Heuristic Methods. *European Journal of Operational Research* **1980**, 5, 153–162.
8. Wolpert, D. H.; Macready, W. G., No Free Lunch Theorems for Optimization. *IEEE Transactions of Evolutionary Computation* **1997**, 1, 67-82.
9. Glover, F.; Laguna, M., Tabu Search. **1997**.
10. Handbook of Approximation Algorithms and Metaheuristics. **2007**, 1351.
11. Mucherino, A.; Seref, O., Modeling and Solving Real Life Global Optimization Problems with Meta-Heuristic Methods. *Advances in Modeling Agricultural Systems* **2009**, 25, 1-27.
12. Brownlee, J., Clever Algorithms Nature-Inspired Programming Recipes. **2011**, 436.
13. Forbes, N., Imitation of Life: How Biology Is Inspiring Computing. **2005**.
14. Castro, L. N. d.; Zuben, F. J. V., Recent developments in biologically inspired computing. **2005**.
15. Salvador García, D. M., Manuel Lozano, Francisco Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Jornaul of Heuristics* **2009**, 15, 617–644.
16. Dhal, K. G.; Ray, S.; Das, A.; Das, S., A Survey on Nature-Inspired Optimization Algorithms and Their Application in Image Enhancement Domain. *Archives of Computational Methods in Engineering* **2018**, 1-32.
17. Kennedy, J.; Eberhart, R. C., Swarm Intelligence. **2001**, 512.

18. Kennedy, J.; Eberhart, R. C., In *IEEE International Conference on Neural Networks*, Piscataway, Ed.; New York, USA, 1995; Vol. 4, pp 1942-1948.
19. Dorigo, M.; Blum, C., Ant colony optimization theory: A survey. *Theoretical Computer Science* **2005**, 344 (2-3), 243-278.
20. Holland, J., *Adaptation in natural and artificial systems*. MIT Press: MI, 1975.
21. Goldberg, D., *Genetic algorithms for search, optimization, and machine learning*. Reading, Addison-Wesley: MA, 1989.
22. Rechenberg, I., *Cybernetic solution path of an experimental problem*. library translation number 1122 ed.; Royal aircraft establishment: Farnborough, 1965.
23. Storn, R.; Price, K., Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **1997**, 11 (4), 341-359.
24. Fogel, D., *Evolutionary Computation*. IEEE Press: NY, 1995.
25. Eiben, A. E.; Smith, J. E., Introduction to Evolutionary Computing. **2003**, 152.
26. Eiben, A. E.; Smith, J., From evolutionary computation to the evolution of things. *Nature* **2015**, 521, 476.
27. Kirkpatrick, S.; Gellat, C.; Vecchi, M., Optimization by simulated annealing. *Science* **1983**, 671-680.
28. Alatas, B., ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications* **2011**, 38(10), 13170–13180.
29. Lam, A. Y. S.; Li, V. O. K., Chemical-reaction-inspired metaheuristic for optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* **2010**, 14(3), 381–399.
30. Deb, K., A population-based algorithm-generator for real-parameter optimization. *Soft Computing* **2004**.
31. Yuan, Y.; Xu, H.; Wang, B.; Yao., X., A New Dominance Relation Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **2016**, 20, 16-37.
32. Li, X. L. New ıntelligent optimization-artificial fish swarm algorithm. Zhejiang University China, 2003.
33. Wang, C.; Zhang, Z.; Xu, R.; Li, M., In *6th International Symposium on Intelligence Computation and Applications, ISICA 2012*; Wuhan, 2012; Vol. 316 CCIS, pp 1-7.
34. Neshat, M.; Sepidnam, G.; Sargolzaei, M.; Toosi, A. N., Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial intelligence review* **2014**, 42, 965-997.
35. Agrawal, V.; Sharma, H.; Bansal, J. C., In *Proceedings of the International Conference on SocProS*; 2011, pp 227-242.
36. Passino, K. M., Biomimicry of bacterial foraging for distributed optimization and control. *EEE Control Systems Magazine* **2002**, 22(3), 52-67.
37. Saikia, L. C.; Sinha, N.; Nanda, J., Maiden application of bacterial foraging based fuzzy IDD controller in AGC of a multi-area hydrothermal system. *International Journal of Electrical Power and Energy Systems* **2013**, 45, 98-106.
38. Yang, X.-S., A New Metaheuristic Bat-Inspired Algorithm. *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* **2010**, 65-74.
39. Gandomi, A. H.; Yang, X.-S.; Alavi, A. H.; Talatahari, S., Bat algorithm for constrained optimization tasks. *Neural Computing and Applications* **2012**, 1-17.

40. Haddad, O. B.; Afshar, A.; Mariño, M. A., Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. *Water Resources Management* **2006**, 20, 661-680.
41. Chu, S.; Tsai, P.; Pan, J., Cat swarm optimization. *Lecture Notes in Computer Science* **2006**, 4099, 854–858.
42. Pradhan, P. M.; Panda, G., Solving multiobjective problems using cat swarm optimization. *Expert Systems with Applications* **2012**, 39(3), 2956–2964.
43. Yang, X. S., In *Stochastic algorithms: Foundations and applications*; Lecture Notes in Comput Science: 2009; Vol. 5792, pp 169-178.
44. Krishnanand; N., K.; Ghose, D., Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multiagent and Grid Systems* **2006**, 2(3), 209–222.
45. Zhang, H.; Fu, P.; Liu, Y., Parameter settings analysis for glowworm swarm optimization algorithm. *Journal of Information and Computational Science* **2012**, 9, 3231-3240.
46. Ghandehari, N.; Miranian, E.; Maddahi, M., In *Lecture Notes in Electrical Engineering; International Joint Conference on Advances in ITC2102 and PEIE2012*; BangalorE, 2013; Vol. 150 LNEE, pp 83-89.
47. Gandomi, A. H. A., A.H., Krill Herd Algorithm: A New Bio-Inspired Optimization Algorithm. *Communications in Nonlinear Science and Numerical Simulation* **2012**.
48. Mucherino, A.; Seref, O., In *AIP Conference Proceedings 953, Data Mining, System Analysis and Optimization in Biomedicine*; 2007, pp 162-173.
49. Yi, T.-H.; Li, H.-N.; Zhang, X.-D., A modified monkey algorithm for optimal sensor placement in structural health monitoring. *Smart Materials and Structures* **2012**, 21.
50. Yang, X.-S.; Deb, S., Cuckoo search: recent advances and applications. *Neural Computing and Applications* **2014**, 24, 169-174.
51. Dhiman, G.; Kumar, V., Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software* **2017**, 114, 48-70.
52. Yazdani, M.; Jolai, F., Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering* **2016**, 3, 24-36.
53. He, S.; Wu, Q. H.; Saunders, J. R., Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE transactions on evolutionary computation* **2009**, 13, 973-990.
54. Kaveh, A.; Farhoudi, N., A new optimization method: Dolphin echolocation. *Advances in Engineering Software* **2013**, 59, 53-70.
55. Simon, D., Biogeography-based optimization. *IEEE transactions on evolutionary computation* **2008**, 12, 702-713.
56. Mirjalili, S.; Lewis, A., The whale optimization algorithm. *Advances in engineering software* **2016**, 95, 51-67.
57. Abedinia, O.; Amjady, N.; Ghasemi, A., A new metaheuristic algorithm based on shark smell optimization. *Complexity* **2016**, 21, 97-116.
58. Mirjalili, S.; Gandomi, A. H.; Mirjalili, S. Z.; Saremi, S.; Faris, H.; Mirjalili, S. M., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* **2017**, 114, 163-191.
59. Mirjalili, S., Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems* **2015**, 89, 228-249.

60. Mirjalili, S., Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications* **2016**, 27, 1053-1073.
61. Askarzadeh, A., A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures* **2016**, 169, 1-12.
62. Mirjalili, S., The ant lion optimizer. *Advances in Engineering Software* **2015**, 83, 80-98.
63. Yang, X.-S.; Karamanoglu, M.; He, X., Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering Optimization* **2014**, 46, 1222-1237.
64. Cheng, M.-Y.; Prayogo, D., Symbiotic organisms search: a new metaheuristic optimization algorithm. *Computers & Structures* **2014**, 139, 98-112.
65. Pattnaik, S. S.; Bakwad, K. M.; Sohi, B. S.; Ratho, R. K.; Devi, S., Swine influenza models based optimization (SIMBO). *Applied Soft Computing* **2013**, 13, 628-653.
66. Jaderyan, M.; Khotanlou, H., Virulence optimization algorithm. *Applied Soft Computing* **2016**, 43, 596-618.
67. Li, M. D.; Zhao, H.; Weng, X. W.; Han, T., A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in Engineering Software* **2016**, 92, 65-88.
68. Salmani, M. H.; Eshghi, K., A metaheuristic algorithm based on chemotherapy science: CSA. *Journal of Optimization* **2017**, 2017.
69. Eesa, A. S.; Brifcani, A. M. A.; Orman, Z., Cuttlefish algorithm-a novel bio-inspired optimization algorithm. *International Journal of Scientific & Engineering Research* **2013**, 4, 1978-1986.
70. Xie, L.; Zeng, J.; Cui, Z., General framework of artificial physics optimization algorithm. *IEEE Nature & Biologically Inspired Computing* **2009**, 1321–1326.
71. Xie, L.; Zeng, J.; Formato, R. A., Convergence analysis and performance of the extended artificial physics optimization algorithm. *Applied Mathematics and Computation* **2011**, 218, 4000-4011.
72. Erol, O. K. y. E., I., A new optimization method: Big bang-big crunch. *Advances in Engineering Software* **2006**, 37 (2), 106-111.
73. Hasançebi, O.; Kazemzadeh Azad, S., An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. *Computers and Structures* **2012**, 110-111, 167-179.
74. Formato, R. A., Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research* **2007**, 77, 425–491.
75. Ding, D.; Qi, D.; Luo, X.; Chen, J.; Wang, X.; Du, P., Convergence analysis and performance of an extended central force optimization algorithm. *Applied Mathematics and Computation* **2012**, 219, 2246-2259.
76. Kaveh, A.; Talatahari, S., A novel heuristic optimization method: charged system search *Acta Mech* **2010**, 213, 267–89.
77. Birbil, S. I.; Fang, S. C., An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization* **2003**, 25, 263–282.
78. Yang, X.; Qiu, D.; Shen, J., In *Advances in Intelligent Systems and Computing*; International Conference on Communication, Electronics, and Automation Engineering, 2012;Xi'an, 2013; Vol. 181 AISC, pp 923-927.
79. Shah-Hosseini, H., Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering* **2011**, 6 (1/2), 132-140.

80. Abdechiria, M.; Meybodib, M. R.; Bahramia, H., Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO). *Applied Soft Computing* **2012**.
81. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S., GSA: A gravitational search algorithm. *Inform Sciences* **2009**, 179(13), 2232–2248.
82. Ajami, A.; Armaghan, M., Application of multi-objective gravitational search algorithm (GSA) for power system stability enhancement by means of STATCOM. *International Review of Electrical Engineering* **2012**, 7, 4954-4962.
83. Ahrari, A.; Atai, A. A., Grenade Explosion Method—A novel tool for optimization of multimodal functions. *Applied Soft Computing* **2010**, 10, 1132–1140.
84. Shah-Hosseini, H., The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation* **2008**, 1(1/2), 71–79.
85. Niu, S. H.; Ong, S. K.; Nee, A. Y. C., An improved Intelligent Water Drops algorithm for achieving optimal job-shop scheduling solutions. *International Journal of Production Research* **2012**, 50, 4192-4205.
86. Sacco, W. F.; de Oliveira, C. R. E. A new stochastic optimization algorithm based on a particle collision metaheuristic. In In 6th world congresses of structural and multidisciplinary optimization, 2005; 2005.
87. Kaveh, A.; Khayatazad, M., A new meta-heuristic method: Ray Optimization. *Computers and Structures* **2012**, 112-113, 283–294.
88. Cáceres, A. Y. P. Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejos. Universidad Central "Marta Abreu" de Las Villas, 2009.
89. BELLO, R.; PURIS, A.; GÓMEZ, Y.; FALCÓN, R., Feature selection through Dynamic Mesh Optimization. *Lecture Notes on Computer Sciences LNCS* **2008**, 5197, 348-355.
90. Tamura, K.; Yasuda, K., Spiral dynamics inspired optimization. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **2011**, 15, 1116-1122.
91. Kaveh, A.; Dadras, A., A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software* **2017**, 110, 69-84.
92. Do\ugan, B.; Ölmez, T., A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences* **2015**, 293, 125-145.
93. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M., Water cycle algorithm--A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures* **2012**, 110, 151-166.
94. Zheng, Y.-J., Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research* **2015**, 55, 1-11.
95. Van Laarhoven, P. J. M.; Aarts, E. H. L. Simulated annealing. In; Springer: 1987, pp 7-15.
96. Kaboli, S. H. A.; Selvaraj, J.; Rahim, N. A., Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science* **2017**, 19, 31-42.
97. Kashan, A. H., A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & Operations Research* **2015**, 55, 99-125.
98. Javidy, B.; Hatamlou, A.; Mirjalili, S., Ions motion algorithm for solving optimization problems. *Applied Soft Computing* **2015**, 32, 72-79.
99. Wedyan, A.; Whalley, J.; Narayanan, A., Hydrological cycle algorithm for continuous optimization problems. *Journal of Optimization* **2017**, 2017.

100. Kaedi, M., Fractal-based algorithm: A new metaheuristic method for continuous optimization. *Int. J. Artif. Intell* **2017**, 15, 76-92.
101. Kaveh, A.; Mahdavi, V. R., Colliding bodies optimization: a novel meta-heuristic method. *Computers & Structures* **2014**, 139, 18-27.
102. Yan, G.-W.; Hao, Z.-J., A novel optimization algorithm based on atmosphere clouds model. *International Journal of Computational Intelligence and Applications* **2013**, 12, 1350002-1350002.
103. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In IEEE Congress on Evolutionary Computation, 2007; 2007; Vol. 7; pp 4661–4666.
104. Borji, A., A new global optimization algorithm inspired by parliamentary political competitions. . *Lecture Notes in Computer Science* **2007**, 4827, 61–71.
105. Venkata Rao, R.; Savsani, V. J.; Vakharia, D. P., Teaching-learning-based-optimization: A novel optimization method for continuous non-linear large scale problems. *Information Sciences* **2012**, 183 (1), 1-15.
106. Shayeghi, H.; Dadashpour, J., Anarchic society optimization based PID control of an automatic voltage regulator (AVR) system. *Electrical and Electronic Engineering* **2012**, 2, 199-207.
107. Lei, D.; Zheng, Y.; Guo, X., A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *International Journal of Production Research* **2017**, 55, 3126-3140.
108. Bastos Filho, C. J. A.; de Lima Neto, F. B.; Lins, A. J. C. C.; Nascimento, A. I. S.; Lima, M. i. P. Fish school search. In; Springer: 2009, pp 261-277.
109. Ebrahimi, A.; Khamehchi, E., Sperm whale algorithm: an effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering* **2016**, 29, 211-222.
110. Mirjalili, S.; Mirjalili, S. M.; Lewis, A., Grey wolf optimizer. *Advances in engineering software* **2014**, 69, 46-61.
111. Shi, Y. Brain storm optimization algorithm. 2011; pp 303-309.
112. Kulkarni, A. J.; Durugkar, I. P.; Kumar, M. Cohort intelligence: a self supervised learning behavior. 2013; pp 1396-1400.
113. Glover, F.; Laguna, M. Tabu search. In; Springer: 1998, pp 2093-2229.
114. Blum, C.; Puchinger, J.; Raidl, G. R.; Roli, A., Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing* **2011**, 11, 4135–4151.
115. Herrera, F., Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence* **2008**, 1, 27-46.
116. Burke, E. K.; Hyde, M.; Kendall, G.; Ochoa, G.; Ozcan, E.; Woodward, J. R., In *Handbook of Metaheuristics*, Gendreau, M.; Potvin, J.-Y., Eds.; New York, 2010; Vol. 146, pp 449-468.
117. Bilgin, B.; Demeester, P.; Misir, M.; Vancroonenburg, W.; Vanden Berghe, G., One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics* **2012**, 18, 401-434.
118. Anagnostopoulos, K.; Koulinas, G., Resource-constrained critical path scheduling by a GRASP-based hyperheuristic. *Journal of Computing in Civil Engineering* **2012**, 26, 204-213.

119. Tabataba, F. S.; Mousavi, S. R., A hyper-heuristic for the Longest Common Subsequence problem. *Computational Biology and Chemistry* **2012**, 36, 42-54.
120. Rafique, A. F.; He, L.; Kamran, A.; Zeeshan, Q., Hyper heuristic approach for design and optimization of satellite launch vehicle. *Chinese Journal of Aeronautics* **2011**, 24, 150-163.
121. Ochoa, G.; Walker, J.; Hyde, M.; Curtois, T., In; Taormina, 2012; Vol. 7492 LNCS, pp 418-427.
122. Alatas, B., A novel chemistry based metaheuristic optimization method for mining of classification rules. *Expert Systems with Applications* **2012**, 39, 11080–11088.
123. Wang, C., Distribution network reconfiguration based on chemical reaction optimization. *Dianwang Jishu/Power System Technology* **2012**, 36, 209-214.
124. Xu, J.; Lam, A. Y. S.; Li, V. O. K., Chemical reaction optimization for task scheduling in grid computing. *IEEE Transactions on Parallel and Distributed Systems* **2011**, 22, 1624-1631.
125. Li, J. Q.; Pan, Q. K., Chemical-reaction optimization for flexible job-shop scheduling problems with maintenance activity. *Applied Soft Computing Journal* **2012**, 12, 2896-2912.
126. Xu, J.; Lam, A. Y. S.; Li, V. O. K., Stock portfolio selection using chemical reaction optimization. *World Academy of Science, Engineering and Technology* **2011**, 77, 458-463.
127. Crick, F., Central Dogma of Molecular Biology. *Nature* **1970**, 227, 561-563.
128. Karplus, M.; Weaver, D. L., Protein folding dynamics: the diffusion-collision model and experimental data. *Protein science : a publication of the Protein Society* **1994**, 3, 650-668.
129. Fersht, A. R., Optimization of rates of protein folding: the nucleation-condensation mechanism and its implications. *Proceedings of the National Academy of Sciences of the United States of America* **1995**, 92, 10869-10873.
130. Wetlaufer, D. B., Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc Natl Acad Sci U S A* **1973**, 70, 697-701.
131. Fersht, A. R., Transition-state structure as a unifying basis in protein-folding mechanisms: contact order, chain topology, stability, and the extended nucleus mechanism. *Proceedings of the National Academy of Sciences of the United States of America* **2000**, 97, 1525-1529.
132. Abkevich, V. I.; Gutin, A. M.; Shakhnovich, E. I., Free energy landscape for protein folding kinetics: Intermediates, traps, and multiple pathways in theory and lattice model simulations. **1994**, 101, 6052-6062.
133. Berger, B.; Leighton, T., Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of computational biology : a journal of computational molecular cell biology* **1998**, 5, 27-40.
134. Dill, K. A., Theory for the folding and stability of globular proteins. *Biochemistry* **1985**, 24, 1501-1509.
135. Harik, G. R.; Goldberg, D. E. Learning Linkage. In *Foundations of Genetic Algorithms IV* Belew, R. K.; Vose, M. D., Eds.; 1996, pp 247-262.
136. Parr, R. G.; Chattaraj, P. K., Principle of Maximum Hardness. *J. Am. Chem. Soc.* **1991**, 113, 1854-1855.
137. Ruiz-Blanco, Y. B.; Alves, M. J.; Rodríguez-Borges, J. E.; Molina, R., Study based on electronic descriptors of the diastereoselective aza-diels-alder cycloaddition of [(1r)-10-(n,n-diethylsulfamoyl)isobornyl] 2h-zirine-3-carboxylate to e,e-1,4-diacetoxy-1,3-butadiene. *J. Chil. Chem. Soc.* **2013**, 58, 2243-2247.
138. Chattaraj, P. K.; Sengupta, S., Popular Electronic Structure Principles in a Dynamical Context. *J. Phys. Chem.* **1996**, 100, 16126–16130.