AN OPTIMIZATION ALGORITHM BASED ON BACTERIA BEHAVIOR

Ricardo Contreras[1] and Rodrigo Neira and M. Angélica Pinninghoff and Homero Urrutia[2] and Ricardo Contreras[3]

[1]Department of Computer Science, University of Concepción, Concepción, Chile
[2]Biotechnology Center, University of Concepción, Concepción, Chile
[3]School of Natural Computing Sciences, Physics, University of Aberdeen, UK

ABSTRACT

Paradigms based on competition have shown to be useful for solving difficult problems. In this paper we present a new approach for solving hard problems using a collaborative philosophy. A collaborative philosophy can produce paradigms as interesting as the ones found in algorithms based on a competitive philosophy. Furthermore, we show that the performance - in problems associated to explosive combinatorial - is comparable to the performance obtained using a classic evolutive approach.

KEYWORDS

Bacterial conjugation, Travelling salesman problem, Evolving systems.

1. INTRODUCTION

Interacting mechanisms existing among living creatures have been an inspiration for developing computational programs capable of solving difficult problems. In the particular case of problems dealing with combinatorial features - in which exact solutions cannot be reached in polynomial time - several studies have been conducted inspired on the principle of life and evolution, e.g. [1][2][3]. In this paper we propose a new strategy based on a collaborative optimization principle that uses the metaphor of life and evolution to introduce a new type of algorithms.

The novelty of our approach is the use of a collaborative mechanism, in contra-position with the classic competitive ones. We claim that a collaborative mechanism performs as good as competitive mechanisms. We developed a prototype for evaluation purposes and our inicial results show that a collaborative mechanism tend to obtain best results, when applied to the same kind of problems, than the results obtained by competition.

The interaction between computer science and biology is an example about the way real life is transforming traditional solving methods in scientific and engineering disciplines. The idea of taking into account nature for problem solving is not new. We know that closer to life the artificial artifacts we build are, the better/faster the solutions we get.

Genetic algorithms and ant colony optimization have shown interesting features when applied to combinatorial problems [4][5]. Furthermore, genetic algorithms are often used on hard problems where other optimization methods fail or are trapped in suboptimal solutions. Genetic algorithms can be applied to a large number of domains, as long as a coherent genetic representation can be

DOI : 10.5121/ijaia.2014.5504

formulated or coupled to other complementary search methods to increase the quality of the solutions [6].

Ant colony optimization (ACO) offers an alternative for dealing with optimization problems. ACO is inspired by pheromone-based strategies of ant foraging. ACO algorithms were originally conceived to find the shortest route, typically by using a graph for modeling the space of solutions. In ACO, several ants travel across the edges that connect the nodes of a graph while depositing virtual pheromones. Different types of ACO algorithms have been developed, most of them based on the work of [7].

ACO can be classified as collaborative paradigm. It exhibits a behavior similar to a micro-organisms behavior, but with a different underlying concept. Ant-based algorithms emulate a specific activity of ants that is to find a path from the nest to the food source. Our proposal emulates the colony evolution, its development, its growing mechanisms, its stability seeking mechanisms, and the common weal. It is a global concept, as genetic algorithms propose, but replacing the competition by the collaboration.

The travelling salesman problem (TSP) is one of the most known and analyzed challenges in the field of optimization. TSP can be stated as follows: given $\boldsymbol{n}$ cities and their intermediate distances, find a shortest route traversing each city exactly once. Mathematically the travelling salesman problem generalizes the question for a Hamiltonian circuit in a graph [8]. Conceptually, in TSP, a travelling salesman must visit every city in his territory exactly once and then return home covering the shortest distance. TSP can be reformulated according different criteria, e.g. time, cost, partial order constraints, to obtain an optimal solution [9].

TSP can be formulated as a simple symmetric problem, where the distance between each pair of cities is the same in either direction, i.e. $dist(i,j) = dist(j,i)$. Note that similarly, the asymmetric distance for TSP is $dist(i,j) \neq dist(j,i)$. In this work we consider the symmetric problems. The size of the search space is $|S| = n! / (2n) = (n-1)!$, where n represents the amount of cities. Consider that a 10-city TSP has about 362880 possible solutions; and a 20-city TSP 10^{16} possible solutions. Because of the latter, we strongly believe that a heuristic approach offers the best alternative to obtain an optimal solution.

Bacteria are microorganisms that have successfully survived for millions of years. They have achieved a wide distribution in the environment and are considered as the most successful biological organisms [10]. One reason for this biological success is related to the way information is transmitted. While in animals and plants information transmission relies on a seed gene transfer, bacteria are also capable of gene transmission among unrelated bacteria. Most of the genes involved in this type of transfer provide to the recipient bacteria genotypic and phenotypic characteristics (originally contained in the donor bacteria). These characteristics give to recipient bacteria capabilities that maximize their ability for surviving under certain environmental conditions [11][12]. Most common examples of new attributes include antibiotic resistance [13], new metabolic and physiological capabilities [14].

Inspired on the collaborative behavior of bacterias, we developed a new Bacterial Conjugation Algorithm (BCA). We used the TSP problem to test the performance of our algorithm. We chose the TSP because it is probably the most well known optimization problem and there are documented experiments and results we can used to compare our findings. We considered the data available at the University of Waterloo[1]. Different approaches have been proposed to deal with TSP. In [15] the authors propose a genetic algorithm that obtains good results, time however,

[1] [http://www.math.uwaterloo.ca/tsp/world/countries.html]

increases along with number of cities. In addition changes in the mutation rate affect the performance of their approach and the quality of solutions is bias by the sizes of the populations. In [16] a hybrid method combining ant colony and beam search is proposed. It relies on the use of an effective local search procedure to improve previous results. In [17] an improved ACO is presented to solve the TSP. The proposal reduces the processing costs involved with routing of ants in the conventional ACO. The work in [1] introduces a hybrid nature inspired approach based on Honey Bees Mating Optimization. It successfully solves the Euclidean TSP. Finally, in [18] the authors propose four improved genetic algorithms using three local search methods: 2-opt search, a hybrid mutation and a combined mutation operator that are incorporated to the sequential constructive crossover.

The concept of bacteria as a model for problem solving is not new. The work in [19] presents a new approach for edge detection using a combination of bacterial foraging (i.e. the behavior bacterial organisms develop to obtain enough food to enable them to reproduce) and a technique derived from Ant Colony Systems. The authors claim that the foraging behavior of some species of bacteria like *Escherichia coli* can be hypothetically modeled as an optimization process. This work derives from the previous proposal in [20], where authors explain the social foraging behavior of *Escherichia coli* and *Myxococcus xanthus* bacteria and develop simulation models based on the principles of foraging theory. Bacterial foraging as an optimization mechanism is explained in detail in [21]. These last three works focus on foraging strategies, as a concept that emphasizes competition in the sense that animals that have successful foraging strategies are more likely to enjoy reproductive success.

The reminder of this paper is structured as follows. Section 2 introduces the fundamental concepts on computational evolutive systems and bacteria characteristics. In section 3 we describe our approach for the treatment of difficult problems. In section 4 we present the results os our approach. Finally in section 5 we present the conclusions and future work.

2. THEORETICAL FRAME

The mid-1980s witnessed a renaissance of diverse approaches to the understanding and engineering of intelligent systems. A range of newly born fields, such as embodied cognitive science, neuromorphic engineering, artificial life, behavior-based robotics, evolutionary robotics, and swarmintelligence, to mention only a few, questioned the validity of the assumptions and methods of mainstream artificial intelligence for creating artifacts that could approximate the operational characteristics and performance of biological intelligence.

Biology is making continuous progress in the description of the components that make up living organisms and the ways in which these components work together.

The theory of natural evolution rests on four pillars: population, diversity, heredity and selection. The premise for evolution is the existence of a population, i.e. a pool of two or more individuals. Diversity means that within a population the individuals vary to some extent from each other. Individual diversity -within and between species- has been observed and described for thousands of years. Heredity indicates that individual characters can be transmitted to offsprings through reproduction. Selection indicates that only a part of the population is capable of reproducing and transmitting its characters to future generations [6].

2.1. Genetic Algorithms

Genetic algorithms (GA) are a particular class of evolutionary algorithms, used for finding optimal or good solutions by examining only a small fraction of the possible space of solutions. GAs are inspired by Darwin's theory about evolution. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

The structure of a genetic algorithm consists of a simple iterative procedure on a population of genetically different individuals. Every individual (a potential solution) is represented by a code; this code is known as the genotype.

On the other hand, the code is intended to represent specific characteristics an individual has. The value of these characteristics that depend on the particular code for an individual is known as phenotype. Phenotypes are evaluated according to a predefined fitness function. The genotypes of the best individuals are copied several times and modified by genetic operators. The newly obtained genotypes are inserted in the population in place of the old ones. This procedure is continued until a \textit{good enough} solution is found [6]. The genetic operators are briefly described bellow:

Selection. Selection is the operation that reflects the fact that only part of the population is capable of reproducing and transmitting its characters to future generations. It considers that individuals with a best fitness value will have a higher probability to be chosen as parents.

Cross-over. Cross-over is used to exchange genetic material, allowing part of the genetic information of one individual to be combined with part of the genetic information of a different individual. Cross-over allows that new individuals can be created by combining in these new individual characteristics that are present in different individuals belonging to a previous generation.

Mutation. Is the mechanism that allows producing slight variations in individuals. Mutation is introduced into the population when new individual are created [6].

2.2. Ant Colony Optimization

Some ant families have the capability of finding the shortest path between their nest and the source of food.

Ants use the environment as a medium of communication. They exchange information indirectly by depositing pheromone, while they pass through a particular trial (or path). The information exchanged has a local scope, only an ant located where the pheromone were left has anotion of it. This property is called *stigmergy* and can be observed in many social animal societies.

The mechanism to solve a problem too complex to be addressed by single ants is a good example of a self-organized system. This system is based on positive feedback (i.e. the deposit of pheromone attracts other ants that will strengthen it themselves) and negative feedback (i.e. dissipation of the route by evaporation prevents the system from thrashing). Theoretically, if the quantity of pheromone remains the same over time on all edges, no route would be chosen. However, because of feedback, a slight variation on an edge will be amplified allowing thus the choice of an edge.

The ACO algorithm will move from an unstable state, in which no edge is stronger than another, to a stable state where the route is composed of the strongest edges. The basic philosophy of the algorithm involves the movement of a colony of ants through the different states of the problem. As a result of the above an ant incrementally constructs a solution to the problem.

2.3. General Bacteria Behavior

Bacteria were among the first life forms to appear on Earth and are present in most of its habitats. Bacteria inhabit soil, wager, acid hot springs, radioactive waste and the deep portions of Earth's crust. Bacteria also live in symbiotic and parasitic relationships with plants and animals. They are also known to have flourished in manned spacecraft.

Bacteria are asexual organisms that inherit identical copies of their parent's genes (i.e., they are clonal). Mutations come from errors made during the replication of DNA or from exposure to mutagens. Mutation rates vary widely among different species of bacteria and even among different clones of a single species of bacteria [22].

Genetic changes in bacterial genomes come from either random mutation during replication or stress-directed mutation, where genes involved in a particular growth-limiting process have an increased mutation rate [23].

Some bacteria also transfer genetic material among cells. This can occur in three main ways. First, bacteria can take up exogenous DNA from their environment, in a process called transformation. Genes can also be transferred by the process of transduction, when the integration of a bacteriophage introduces foreign DNA into the chromosome. The third method of gene transfer is conjugation, whereby DNA is transferred through direct cell contact.

In ordinary circumstances, transduction, conjugation, and transformation involve transfer of DNA between individual bacteria of the same species, but occasionally transfer may occur among individuals of different bacterial species. This may carry significant consequences, such as thetransfer of antibiotic resistance. In such cases, gene acquisition from other bacteria - or the environment - is called horizontal gene transfer and may be common under natural conditions [24]. Gene transfer is particularly important in antibiotic resistance as it allows the rapid transfer of resistance genes between different pathogens [25].

2.4. Bacterial Conjugation

Bacterial conjugation is the transfer of genetic material (plasmid) between bacterial cells by direct cell-to-cell contact or by a bridge-like connection between two cells [26]. Discovered in 1946 by Lederberg and Tatum [27], conjugation is a mechanism of horizontal gene transfer as are transformations and transduction, although the latter two mechanisms do not involve cell-to-cell contact [28].

Bacterial conjugation is often regarded as the bacterial equivalent of sexual reproduction or mating since it involves the exchange of genetic material. During conjugation the donor cell provides a conjugative or mobilizable genetic element that is most often a plasmid or transposon. Most conjugative plasmids have systems ensuring that the recipient cell does not already contain a similar element.

The genetic information transferred is often beneficial to the recipient. Benefits may include antibiotic resistance, xenobiotic tolerance or the ability to use new metabolites. Such beneficial plasmids may be considered bacterial endosymbionts. Other elements, however, may be viewed as bacterial parasites and conjugation as a mechanism evolved by them to allow for their spread.

The prototypical conjugative plasmid is the F-plasmid, or F-factor. The F-plasmid is an episome (a plasmid that can integrate itself into the bacterial chromosome by homologous recombination) with a length of about 100 kb (kilo base pairs). It carries its own origin of replication, the oriV, and an origin of transfer, or oriT. There can only be one copy of the F-plasmid in a given bacterium, either free or integrated, and bacteria that possess a copy are called F-positive or F-plus. Cells that lack F plasmids are called F-negative or F-minus and as such can function as recipient cells [26][29].

Conjugation is a convenient means for transferring genetic material to a variety of targets. In laboratories, successful transfer have been reported from bacteria to yeast, plant, mammalian cells and isolated mammalian mitochondria. Conjugation has advantages over other forms of genetic transfer including minimal disruption of the target's cellular envelope and the ability to transfer relatively large amounts of genetic material. In plant engineering, Agrobacterium-like conjugation complements other standard vehicles such as tobacco mosaic virus (TMV). While TMV is capable of infecting many plant families these are primarily herbaceous dicots (dicotyledons). Agrobacterium-like conjugation is also primarily used for dicots, but monocot (monocotyledon) recipients are not uncommon.

3. THE PROPOSAL

Through a modeling process, that considers the biological characteristics previously mentioned, we hypothesize that it is possible to build an algorithm that taking into account these biological characteristics can help to solve a set of difficult problems, more specifically, optimization problems.

The concept of horizontal evolution in bacterias is supported by the whole genetic evolution of a complete group of individuals in a specific environment, trying to reach a stable state through time. Bacterias achieve evolution due to some particular characteristics, summarized as follows: *Horizontal evolution*, that implies assimilation and transmission of genetic material in real time; *Quorum sensing*, that allows the communication through chemical substances, and that can trigger the coordination and collective genetic expression; and *Colony stability*, that allows the self regulation in production and resources consumption, aiming to a common and stable state for the individuals. To illustrate these ideas, our first approach considers horizontal evolution and colony stability. Quorum sensing has been taken into account in a separate experience.

As in most of evolving approaches, we are dealing with a population of individuals, each individual can represent a particular solution for a specific problem, and the nature of the problem may help to address the way in which to conduct the codification.

The first step is to establish a codification in coherence with the solutions in the search space, and a mechanism for evaluating the goodness for every codification. Solutions are identified with genetic individuals' codes that can be part of the population. Three basic operations are considered; as illustrated in the algorithm schema. The first one, *Transform code*, carries out a transformation process that modifies part of the genetic information. This operation is intended for escaping from a probable local optimum.

The next operation, *Challenge*, compares the goodness of a genetic code that belongs to a particular individual, with the genetic code of a potential enemy.

An enemy is any individual, different to the bacteria, that defies the bacteria (an enemy defies bacteria when it is located in the neighborhood of the bacteria). If the bacterium has a better[2] genetic code than the code from the enemy, the bacteria survive and the organism playing the role of the enemy dies. In the other case, i.e. enemies's code is better, the bacterium dies and the enemy survives. This operation allows improving the genetic code of bacterias. The third operation, *Transference* (in biologic terms it can be understood as an horizontal transference), consists in the complete genetic code transmission among two individuals when they belong to the same neighborhood. This operation allows spreading among the bacteria population improved codes. The bacterias population evolves according to these three elemental operations.

The genetic codification of the bacteria considers a specific number of genes. It does not change during algorithm execution, but depending on the problem can be used as a parameter, i.e., different problems may adapt the number of genes for a better solution representation. The world size is specified as a two dimension matrix, and it is strongly related to the initial population of bacterias. The initial population is randomly coded. The number of enemies, strongly related to the matrix size is, as the initial population, randomly created.

A very important difference between bacterias and their enemies is related to the code storage. An enemy is an individual that stores a single genetic code that remains unchanged until it dies. Bacteria can store one or more genetic codes. The number of codes bacteria can store depends on the memory size for those bacteria. Enemies and bacterias can regenerate the population after a percentage of loses that can be settled as a parameter. It can be considered as a fourth operation that helps to maintain the population size over a specific threshold. This additional operation is called *Regeneration* in the algorithm schema.

For bacterias the regeneration process is produced by random code allocation. For the enemies, it is assumednbthe value of the best genetic code found at this point in the process. The reason for this is that after a number of generations the random regeneration of enemies is not as important as it is at the beginning of the complete process (due to the improvement that bacterias experiment during the evolution). This encourages the born of better bacterias. It is interesting to note that the latter is analogous to the way in which real bacterias interact with antibiotics. After a time interval, they face more and more powerful enemies, and consequently these bacterias are able to evolve to overcome this problem.

Taking into account that it is not a complete the biologic model, but a simplified metaphor of it; the next set of considerations and procedures illustrate the underlying structure for the algorithm. Parameters are listed in the following and the algorithm schema is shown in Figure 1.

W_{size}: world size: size of the two dimensional matrix M. A particular microorganism or enemy is located on a specific $m_{i,j}$ entry.
$M_{initial\_pop:}$ microorganisms initial population, number of microorganisms to be initially created.
$E_{initial\_pop}$: enemies initial population, number of enemies to be initially created.
M_{size}: memory size of a microorganism.
$T_{condition}$: triggering condition; a percentage of *casualties* in microorganisms and enemies that trigger regeneration (two different parameters).

[2] Here, better is to be defined depending on the particular problem we are trying to solve

```
Algorithm 1: Algorithm overview
Data: W_size, M_inittial_pop, E_inittial_pop, M_size, T_condition
Result: Solution
/* Creation of a matrix M                                    */
New M(W_size);
/* Adition of enemies and microorganisms                     */
M ← M_inittial_pop, E_inittial_pop;
while non T_condition do
    /* Count world population                                */
    Count number of microorganisms and enemies;
    if Population below specific value then
        RegeneratesPopulation
    /* Randomisation                                         */
    RandomiseMicroorganismCode;
    if Random code better than original then
        new_code replaces old_code
    /* Challenge                                             */
    for each microorganism in M do
        if Enemy in neighborhood then
            Compares microorganism and enemy code;
            if microorganism code better than enemy code then
                enemy dies;
            else
                microorganism dies
    /* Horizontal Transfer                                   */
    if There is a neighbor (another microorganism, a friend) then
        Compare genetic codes;
        Best code is transferred to the friend;
Return solution;
```

Figure 1. Algorithm schema

4. RESULTS

We have selected the TSP problem for the evaluation of our algorithm. The bacteria codes have been adapted in such a way that every code represents a path candidate to solve the problem. The web site introduced in the Introduction contains the data set we used. This data set contains geographic locations, expressed as two dimensions coordinate representing different cities in different countries. The data set also includes values representing distances associated to some optimum paths through a set of cities. The values in the reference set were obtained by using Concorde TSP Solver[3] .

A matrix, representing the euclidian distance between every pair of cities was constructed. The matrix is used to evaluate the goodness of everycandidate path our algorithm generates. Every genetic code represents a possible solution path. Every gene in this code is a city in the path. For every problem, a set of parameters can be fixed before running the experiments: i) number of genes in a genetic code, ii) Size of the world: allows to specify the size of the matrix in which individuals are going to interact, it is strongly related to the following two parameters, that set the initial populations, iii) Initial micro-organisms population, i.e., the number of micro-organisms that are going to be generated, iv) Initial enemies population, i.e., number of enemies that are

[3] [http://www.math.uwaterloo.ca/tsp/concorde.html]

going to be generated, Size of micro-organism's memory, that set the number of probably different genetic codes a micro-organism can store, v) Triggering condition for regeneration, in our experiments, when loses of individuals reach the 1%, the process of regeneration is triggered. The value is the same for bacterias and their enemies, but in general it can be separately assigned and vi) Mutation rate, that specifies the number of times a genetic code in a micro-organism can, randomly, change.

We conducted experiments to evaluate the performance of our algorithm. We considered 27 sets of cases, were each case represents a specific parametrization. Table 1 shows same results for two problems selected from the set of problems available at the Waterloo, previously cited. Problem 1 is the TSP containing 29 cities that correspond to the geographic location of Western Sahara, Problem 2 is the TSP containing 38 cities that correspond to the geographic location of Djibuti (in front of Gulf of Aden, Arabian Sea).

Problem 1 is solved, by using our algorithm, reaching the optimum value in 50% of experiments, approximately. Problem 2 is solved by using our algorithm, reaching the optimum value in 30% of experiment cases, approximately. For Problem 1, best values are obtained with a population of 1000 bacterias and 1000 enemies. The memory of bacteria needs only to store two different codes. For Problem 2, best values need a population 1000 individuals (bacterias and enemies in only one case), and a population of 5000 bacterias and 5000 enemies, for the rest of successful experiments. In addition, Problem 2 requires the bacteria uses a larger memory to store different codes; the optimum value requires always storing 100 different codes.

Table 1. TSP results obtained by using BCA.

Experiment number	**Distance problem 1**	**Distance problem 2**
	27601 (Optimum value)	**6659 (Optimum value)**
1	27944	9152
2	27620	8580
3	**27601**	6841
4	**27601**	7115
5	27620	6848
6	**27601**	6710
7	27620	**6659**
8	**27601**	**6659**
9	**27601**	**6659**

We compare our results with the results from previous experiments. Just like in our work, they evaluate their approaches for TSP instances, having comparable sizes, and comparable hardware components. We found that our algorithm performs well in terms of number of optimum solutions and response time, i.e., our preliminary results obtain a larger number of optimum solutions than those obtained by using genetic algorithms, and very similar to the approaches based on ant colony.

Table 2 shows results reported by different authors, that focuse on genetic algorithms; in particular diiferent genetic operator schemes. In this table, **n** represent the number of cities, and **Err** represents the minimum value obtained for the error percentage. Nevertheless, it is important to mention that different tests considered different instances, and that time was not considered as an important result, because slightly different hardware components may lead to important differences in processing time.

Comparisons consider the work of Deep et al. [30], that introduce variations of the order crossover operator (Test1 in Table 2); the work of Ahmed [31], that introduces a new crossover operator: the Sequential Constructive crossover (Test2 in Table 2); the results of Mitchell et al. [32] that evaluate the effectiveness of a new genetic operator: GeneRepair, developed to correct invalid tours generated following crossover or mutation, and the results reported in the work of Rani et al.[33], that implement roulette wheel selection operation with different crossover and mutation probabilities.

Table 2. A comparison of experimental results.

Test case	n	Err
Test1	50 - 76	3.1%
Test2	51 -76	9.8%
Test3	16 – 51	1%
Test4	5 – 50	4.3
BCA	29 – 38	0%

5. CONCLUSIONS

In this work it has been proposed a novel strategy to face some optimization problems, and results show that it is a valid approach. Our preliminary results are very promising and we plan to develop a systematic mechanism for assigning values to the involved parameters.

In addition to the above there are additional issues we plan to address in our future research: *intensification* and *diversification*. In relation to the space of solutions and how it can be bound and the increment of the size of the world, we believe this will allow the algorithm to move across different spaces, avoiding the convergence to a unique solution. We also plane to take into consideration what was mentioned in section 3 related to quorum sensing. We strongly believe that these factors can positively impact the quality of our approach and thus, future results.

REFERENCES

[1] Marinakis, Y. & Marinaki, M. & Dounias, G. (2011) "Honey bees mating optimization algorithm for the Euclidean travelling salesman problem", Information Sciences 181(20) :4684-4698.
[2] Guzmán, M.A. & Delgado, A. & De Carvalho, J. (2010) "A novel multiobjective optimization algorithm based on bacterial chemotaxis", Eng. Appl. Artif. Intell. 523(3): 292-301.
[3] Pintea, C.M. (2014) Advances in Bio-inspired Computing for Combinatorial Optimization Problems, Springer.
[4] Haupt, R. & Haupt, S.E. (2004), Practical Genetic Algorithms, Wiley.
[5] Dorigo, M. & Stutzle, T. (2004) Ant Colony Optimization, MIT Press.
[6] Floreano, D. & Mattiusi, C. (2008) Bio-inspired Artificial Intelligence, MIT Press.
[7] Dorigo, M. & Maniezzo, V. & Colorni, A. (1996) "Ant system: Optimization by a colony of cooperating agents", Trans. Sys. Man Cyber. Part B, 26(1): 29-41.
[8] Biggs, N.L. & Lloyd, E.K. & Wilson, R.J. (1976) Graph Theory 1736 - 1936, Clarendon Press, Oxford.
[9] Michalewicz, Z. & Fogel, D.B. (2000) How to Solve it: Modern Heuristics, Springer.
[10] Womack, A. & Bohannan, B. & Green, J. (2010), "Biodiversity and biogeography of the atmosphere", ABC Philosophical Transactions of the Royal Society B. 365:3645-3653.
[11] Fournier, G. & Cogarten, P. (2008), "Evolution of acetoclastic methanogenesis in methanosarcina via horizontal gene transfer from cellulolytic clostridia", Journal of Bacteriology. 190(3): 1124-1127.
[12] Babic, A. & Lindner, A. & Vulic, B. & Stewart, E. & Radman, M. (2008), "Direct visualization of horizontal gene transfer", Science. 319:1533-1536.

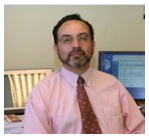
[13] Wiedenbeck, J. & Cohan, F.M. (2011), "Origins of bacterial diversity through horizontal genetic transfer and adaptation to new ecological niches", FEMS Microbiology Reviews. 35:957-966.
[14] Sjoestrand, J. & Tofigh, A. & Daubin, V. & Arvestad, L. & Sennblad, B. & Lagergren, J. (2014) "A Bayesian method for analysing lateral gene transfer", Systematic Biology.
[15] Philip, A. & Taofiki, A.A. & Kehinde, O. (2011), "A genetic algorithm for solving travelling salesman problem", International Journal of Advanced Computer Science and Application, 2(1).
[16] López-Ibáñez, M. & Blum, C. (2010), "Beam-aco for the travelling salesman problem with time windows", Computers & Operations Research, 37:1570-1583.
[17] Jun-man, K. & Yi, Z. (2012), "Application of an improved ant colony optimization on generalized travelling salesman problem", Energy Procedia, 17(A):319-325.
[18] Ahmed, Z.H. (2014), "Improved genetic algorithm for the travelling salesman problem", Int. J. Process Management and Benchmarking, 4(1):109-124.
[19] Verma, O.P. & Handmandlu, M. & Kumar, P. &. Chhabra, S. & Jindal, A. (2011) "A novel bacterial foraging technique for edge detection", Pattern Recognition Letters, 1187-1196.
[20] Liu, Y. & Passino, K. (2002), "Biomimicry of bacterial foraging for distributed optimization: Models, principles, and emerging behaviors", Journal of Optimization Theory and Applications, 115:603-628.
[21] Passino, K.M. (2002) "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine, 22(3): 5267.
[22] Denamur, E. & Matic, I. (2006) "Evolution of mutation rates in bacteria", Molecular Microbiology, 60(4): 820-827.
[23] Wright, B.E. (2004) "Stress-directed adaptive mutations and evolution", Molecular Microbiology, 52 (3): 643-650.
[24] Davison, J. (1999) "Genetic exchange between bacteria in the environment", Plasmid, 42(1): 73-91.
[25] Hastings, P.J. & Rosenberg, S.M. & Slack, A. (2004) "Antibiotic-induced lateral transfer of antibiotic resistance", Trends in Microbiology, 12(9): 401-404.
[26] Holmes, R.K. & Jobling, M.G. (1996) Medical Microbiology, University of Texas Medical Branch at Galveston, 4th edition.
[27] Lederberg, J. & Tatum, E.L. (1946), "Gene recombination in Escherichia coli", Nature, 158(4016): 158.
[28] Griffiths, A.J.F. & Wessler, S. & Carroll, S. & Doebley, J. (2012) Introduction to genetic analysis, W.H. Freeman.
[29] Ryan, K.J. & Ray, C.G. (2004) Sherris Medical Microbiology, McGraw-Hill.
[30] Deep, K. & Mebrahtu, H. (2011), "New variation of order crossover for travelling salesman problem", IJCOPI, 2(1): 2-13.
[31] Ahmed, Z.H. (2010), "Genetic algorithm for the travelling salesman problem using sequential constructive crossover operator", International Journal of Biometrics & Bioinformatics (IJBB), 3(6): 96-105.
[32] Mitchell, G.G. & O'Donoghue, D. & Barnes, D. & McCarville, M. (2003), "GeneRepair – A repair operator for genetic algorithms", GECCO Conference, Ilinois.
[33] Rani, K. & Kumar, V. (2014), "Solving travelling Salesman problem using genetic algorithm based on heuristic crossover and mutation operator", International Journal of Research in Engineering and Technology. 2(2):27-34.

Authors

Ricardo Contreras A. Is an associate professor in the Department of Computer Science at the University of Concepción. His research focuses on artificial intelligence and evolutionary computation. His current research interests include bio-inspired system for solving optimization problems. Contreras received an MSc in computer science from Pontifical Catholic University, Rio de Janeiro, Brazil.

Rodrigo Neira J. Is an Informatic Engineer, graduated at the University of Concepción. His research interests include Software Engineering and Artificial Intelligence. He received received an MSc in computer science from the University of Concepción.

M. Angélica Pinninghoff J. Is an associate professor in the Department of Computer Science at the University of Concepción. Her research interests include artificial intelligence, evolutionary computation, and intelligent optimization. Her current work includes image processing, in particular biological images, as a mechanism for improving images interpretation. Pinninghoff received an MSc in computer science from the University of Concepción.



Homero Urrutia B. Is researcher in the Center of Biotechnology at the University of Concepción. His research interests include: Structure, function and application in environmental biotechnology of microbial biofilms: fouling, biocorrosion, industrial waste treatment and aquaculture biotechnology, methane production, biobarriers. He received a PhD in Environmental Sciences from the University of Concepción.

Ricardo Contreras A. Is a researcher at the University of Aberdeen, UK. His current interests are Neural Networks, ECC, linear models, dynamic systems, chaos theory, granger causality, brain activity/models (MRI, EEG). His studies include an Informatics Engineering Degree. At Universidad del Bío-Bío, Concepción, Chile. He received the Master's Degree (M2R). from Université de Montpellier II, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Montpellier, France. He also received a PhD in Computer Science from. City University London, London, UK.