

Stochastic Metaheuristics as Sampling Techniques using Swarm Intelligence

Johann Dreo and Patrick Siarry
*Universite Paris 12, Laboratoire Images, Signaux et Systemes Intelligents
 France*

1. Introduction

Optimization problems appear in many fields, as various as identification problems, supervised learning of neural networks, shortest path problems, *etc.* Metaheuristics [22] are a family of optimization algorithms, often applied to "hard" combinatorial problems for which no more efficient method is known. They have the advantage of being generic methods, thus do not require a complex tuning for each problem, and can be used as a kind of "black boxes". Recall that, generally, optimization algorithms search for a *point* into the *search space*, so as to optimize (*i.e.*, minimize or maximize) the *objective function* (also called fitness or goal function). Metaheuristics are often divided into two sets:

1. Algorithms handling a *single point*, making it evolve towards a solution.
2. Algorithms handling a *population*, *i.e.*, a finite set of points, and computing a new population at each iteration.

An essential observation is that the population of the second category is a stochastic *sampling* of the objective function. Although those classes are not disjoint (an algorithm can belong to both classes, according to the point of view), we only consider population metaheuristics, which are simply referred as *metaheuristics* hereafter.

An important contribution in this domain comes from the theory of self-organization [10, p.8], which allows to analyze the properties of several metaheuristics stemming from real-world metaphors, often biological ones. This theory (notably studied except the biology [47]) describes the conditions of appearance of complex phenomena from distributed systems, the agents of which are the object of simple, but numerous interactions. The theory puts in front concepts such as communication, feedback, amplification of fluctuations and emergence. In the metaheuristics field, swarm intelligence was so explicitly used on two main fronts: via an approach "self-organized systems" (having given place to ant colony algorithms) and via an approach "socio-cognitive systems" (having led to the particle swarm optimization).

We suggest putting the theory of the swarm intelligence in connection with the concept of adaptive learning search, which tries to describe key points of modern metaheuristics, notably by insisting on the role of the learning and the mechanisms of intensification and diversification. More generally, we think that the theory of self-organization combined with the adaptive learning search gives keys to design the basic components of metaheuristics, recovering from swarm intelligence.

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

2. Fundamental concepts

2.1 Adaptive Memory Programming

Adaptive Memory Programming (AMP) is a common framework to metaheuristics [53], described in Algorithm 1. It stresses out the concepts of *memory*, *intensification*, and *diversification*. In the literature of evolutionary algorithms, these two last notions are often replaced by the words *exploitation* and *exploration*, which have a similar meaning.

1. Memorization of a set of solutions, or of a data structure containing the characteristics of the solutions produced by the search.
2. Construction of a temporary solution based on the data memorized.
3. Improvement of the solution by an algorithm of local search.
4. Memorization of the new solution, or of the data structure associated.

Algorithm 1. AMP framework

We briefly detail each element of the AMP framework:

- *Memory* stands for the information collected by the algorithm on the objective function distribution. It can be represented either as a simple set of points, or as more complex structures, like pheromone tracks in ant colony algorithms. Memory can be defined as global (compared to the problem as a whole) or inter-individual (a solution relative to another one).
- *Intensification* exploits the information obtained, in order to improve the current solutions. This is typically a local search algorithm (for instance with the Nelder-Mead algorithm [45] or a taboo search).
- *Diversification* aims at collecting new information, by exploring the search space.

The three components presented are not always clearly distinct, and are strongly interdependent in an algorithm. An example of metaheuristic that fits well the AMP model is the method GRASP [50].

2.2 Objective Function Sampling and AMP

Metaheuristics share a certain number of properties. An essential one is that they handle a sampling of the objective function, via common processes.

The probabilistic sampling should ideally pick the best solutions with higher probability. However, in an optimization problem, the effective goal is not to sample the objective function, but to find the distribution's optimum. Thus, sampling must concentrate on the areas of interest, while converging gradually towards the optimum by means of "learning" algorithms. From the point of view of sampling, this convergence is carried out by a progressive fall of dispersion in these areas.

In the majority of metaheuristics, the sampling of the objective function is probabilistic (diversification, also named exploration, synonym used almost indifferently [51, p.292]). Ideally, this sampling should be performed with respect to an approximation of the distribution of the points, so as to locate an area of interest, and then converge towards the optimum (intensification, or exploitation).

Most of the metaheuristics do not have any *a priori* information on the distribution, thus *implicitly* learn it by diversification and intensification, such as ant colony algorithms, and "classical" metaheuristics. Conversely, some methods use an approximation of the distribution, and are called *explicit* methods (see [3]).

2.3 General Scopes

We assisted to several attempts of structuration in the scope of distribution sampling. For instance, Monmarche *et al.* proposed the model Probabilistic Search Metaheuristic [42, 43] (PSM), based on the comparison of the algorithms PBIL [2, 4], BSC [52], and the ant system algorithm [13]. The general principle of a PSM method is presented in Algorithm 2.. Notice the relation of this approach with the estimation of distribution algorithms. However, the PSM approach is limited to the use of probability vectors, while specifying an essential update rule for these vectors.

```

Initialize a probability vector  $p_0(x)$ 
Until stopping criteria:
    Build  $m$  individuals  $x_1^l, \dots, x_m^l$  using  $p_l(x)$ 
    Evaluate  $f(x_1^l), \dots, f(x_m^l)$ 
    Rebuild a probability vector  $p_{l+1}(x)$  while considering  $x_1^l, \dots, x_m^l$  and
         $f(x_1^l), \dots, f(x_m^l)$ 
End

```

Algorithm 2. The scope of the PSM method

```

Initialize a population  $P_0$  of  $n$  points
Until stopping criteria:
    Memorize the worst point  $\theta$ 
    Search an appropriate distribution  $D_i(X)$  from the population  $P_{i-1}$ 
    Build a population  $O_i$  of  $m$  points according to  $D_i(X)$ , with  $\forall O_i^j \in O_i :$ 
         $f(O_i^j) < f(\theta)$ 
    Create a population  $P_i$  from a part of  $P_{i-1}$  and a part of  $O_i$ 
    Evaluate  $P_i$ 
End

```

Algorithm 3. The IDEA approach

The EDA's were presented as evolutionary algorithms, with an explicit diversification [44]. They are undoubtedly the algorithms closest to a general scope. The Iterated Density Evolutionary Algorithms [7, 8, 9] (IDEA) are a generalization of those, presented in Algorithm 3..

IDEA uses a more general diversification than PSM, while not being limited to a probability vector as model, but specifying that the search for the best probability distribution forms an integral part of the algorithm. However, the fall of dispersion is carried out by selecting the best individuals, no precision on the use of different intensification principles is given.

2.4 I&D frame

A classical problem when designing metaheuristics is the difficulty to achieve the balance between intensification and diversification. This has lead Blum and Roli to propose the *I&D frame* [51], which emphasizes the fact that the different components of a metaheuristic cannot be categorized as performing *strict* intensification or diversification. They propose to consider that components could be spaced out between three poles, determined upon the origin of the information comes from:

- the objective function,
- a random process,
- other functions.

Furthermore, each component can be considered as *intrinsic* or *strategic*, depending whether the component is defined by the basic idea of the metaheuristic, or added to it to improve its performances.

However, this framework does not give precise indication on the algorithms design, nor on the relation between components and probabilistic sampling aspects.

2.5 Self-organization and swarm intelligence

As a field of research, swarm intelligence deals with the study of self-organization in natural and artificial swarm systems. The *self-organization* is a phenomenon described in many disciplines, notably in the fields of physics and biology. A formal definition has been proposed in [10, p.8]:

Self-organization is a process in which *pattern* at the global level of a system *emerges* solely from numerous interactions among *lower-level* components of the system.

Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the *global* pattern.

Two terms need clarification for a better understanding, "pattern" and "to emerge". Generally, the first one applies to an "organized arrangement of objects in space or time". Additionally, an *emerging* property of a system is a characteristic which appears unforeseen (not being *explicitly* determined), from the interactions among the components of this system.

Thus, the crucial question is to understand how the components of a system interact with each other to produce a complex pattern (in relative sense of the term, i.e. *more* complex than the components themselves). A certain number of necessary phenomena have been identified: these are the processes of *feedback* and the management of the *information flow*.

The *positive feedbacks* are processes which result in reinforcing the action, for example by amplification, facilitation, self-catalysis, etc. Positive feedbacks are able to amplify the *fluctuations* of the system, permitting the updating of even imperceptible informations. Such

processes can easily lead to an explosion of the system, if they are not controlled by applying *negative feedbacks*. Hence negative feedbacks act as stabilizers for the system. When they are coupled, such feedback processes can generate powerful models.

Within the framework of biological behavior, it is easy to understand that the interactions among the components of a system will very often give rise to *communications* processes i.e. transfer of information between individuals. Generally, individuals can communicate, either by means of signals, i.e. by using a specific means to carry information, or by means of indices, where information is carried accidentally. In a similar manner, information can come directly from other individuals, or pass via the state of a work in progress. This second possibility of exchanging information, by means of modifying the environment, is called the *stigmergy*.

Generally, all these processes are more or less inter-connected, allowing a system consisting of a large number of individuals to act together to solve problems that are too complex for a single individual.

Certain characteristics of the self-organized systems are very interesting, in particular their *dynamism*, or their capacity to generate *stable* patterns. Within the framework of the study of the behavior of the social insects, certain concepts related to the principle of self-organization deserve to be underlined: the intrinsic *decentralisation* of these systems, their organization in *dense heterarchy* and the recurring use of the *stigmergy*. Indeed, these concepts are sometimes used to view the same problem from different angles and partially cover the principles of self-organization.

In a swarm intelligence system, there is no decision-making at a given level, in a specified order and no predetermined actions. In fact, in a decentralized system, each individual has a *local* vision of his environment, and thus does not know the problem as a whole. The literature of the multi-agent systems (see [24] for an initial approach) often employs this term or that of "distributed artificial intelligence" [34]. However, generally this discipline tends to study more complex behaviors patterns, founded in particular in cognitive sciences. To be precise, the advantages of decentralized control are the *robustness* and the *flexibility* [6]. Robust systems are desired because of their ability to continue to function in the event of breakdown of one of their components; flexible devices are welcome, because they can be useful for dynamic problems.

2.6 Adaptive Learning Search

Adaptive Learning Search (ALS) is a framework for considering the structure of metaheuristics [21], relying on the AMP, the I&D frame and the notion of objective function sampling.

Instead of considering only a memorization process, as in AMP, we propose to consider a *learning* phase. Indeed, the memory concept is quite static and passive; in a sampling approach, it suggests that the sample is simply stored, and that the metaheuristic only takes into account the previous iteration, without considering the whole optimization process. We emphasize on the fact that the memorized data is not only a raw input, but provides *information* on the distribution, and thus on the solutions.

Thereby, we propose to consider three terms to describe the characteristic processes in a population metaheuristic: learning, diversification and intensification. Metaheuristics progress in an iterative way, archetypally by alternating phases of intensification, diversification and learning, or mixing these notions in a more narrow way. The state of

departure is often randomly chosen, the algorithm running until a criterion of stop is reached. A simple ALS algorithm could thus be organized as presented in Algorithm 4.

Initialize a sample;

Iterate until stopping criteria:

Sampling: either explicit, implicit or direct,

Learning: the algorithm extracts information from the sample,

Diversification: it searches for new solutions,

Intensification: it searches to improve the existing sample,

Replace the previous sample with the new one.

End

Algorithm 4. ALS algorithm

The diversification indicates the processes harvesting information about the optimized problem. The intensification aims at using the information already harvested to define how much an area is interesting. The memory is the support of the learning, which allows the algorithm to take into account only zones where the global optimum may be, so avoiding the local optima. The notions of intensification and diversification are important in the design of metaheuristics, which have to reach a delicate balance between these two dynamics of search. Both notions are not thus contradictory, but additional, and there are numerous strategies mixing at the same moment both of the aspects.

We use here a terminology similar to the one used for the I&D frame, but slightly modified to be easier to comprehend and manipulate. Notably, we have chosen to assign the terms to archetypal processes:

Intensification: the sampling only uses informations from the objective function (local search, determinist selection operators, etc.),

Diversification: the sampling is purely random (noise, uniform mutation operator),

Learning: use of a distribution constructed from the whole set of solutions sampled from the start of the algorithm.

Moreover, in ALS, we proposed to split up metaheuristics in three categories, according to the way the sampling is managed:

Implicit: an implicit probability density function (PDF) is used to draw the sample (e.g. evolutionary algorithms),

Explicit: a specific PDF is used (e.g. estimation of distribution algorithms),

Direct: an approximation of the objective function is used as a PDF (e.g. simulated annealing).

The implicit methods permit to avoid the hard choice of the PDF model to use, but are difficult to control and understand. Explicit methods permit to control their components almost independently, but are pledged to the choice of a model. The direct algorithms use the "ideal" model (the objective function itself), but make the intensification difficult.



Figure 1. The three classes of metaheuristics proposed in adaptive learning search are defined according to the sampling of the objective function. They combine themselves with the three archetypal components, that form the set of possible behaviours

3. Metaheuristics

Metaheuristics form a wide class of methods, among which the more interesting are often stochastic algorithms manipulating a sample of points (also called a "population" of "individuals"). In this section, we will briefly introduce some of the best known metaheuristics, from the simulated annealing (which does not use swarm intelligence) to ant colony algorithms (a method well known for using swarm intelligence). Each metaheuristic is here described along with its position regarding adaptive learning search and swarm intelligence.

3.1 Simulated Annealing

The simulated annealing [37, 11] was created from the analogy between a physical process (the annealing) and an optimization problem. As a metaheuristic, it is based on works simulating the evolution of a solid towards its minimal energetic state [41, 30].

The classic description of simulated annealing presents it as a probabilistic algorithm, where a point evolves in the search space. The method uses the Metropolis algorithm, recalled in Algorithm 5., inducing a markovian process [1, 38]. The simulated annealing, in its usual version ("homogeneous"), calls this method at each iteration.

```

Initialize a starting point  $x_0$  and a temperature  $T$ 
For  $i = 1$  to  $n$ :
    Until  $x_i$  accepted
        If  $f(x_i) \leq f(x_{i-1})$ : accept  $x_i$ 
        If  $f(x_i) > f(x_{i-1})$ : accept  $x_i$  with a probability  $e^{-\frac{f(x_i) - f(x_{i-1})}{T}}$ 
    End
End

```

Algorithm 5. Sampling with the Metropolis method

It is possible to see the simulated annealing as a population algorithm. Indeed, the Metropolis algorithm directly samples the objective function using a degenerated parametric Boltzmann distribution (of parameter T). Hence, one of the essential parameters is the temperature decrease, for which many laws were proposed [54]. There also exist some versions of the simulated annealing more centred on the handling of a points population [32, 55, 40, 33].

Here, the Metropolis method represents the diversification (coupled with the learning), while the temperature decrease is controlling the intensification process. Note that other methods than Metropolis' may be used [14, 48].

Algorithm 6. presents a synthesis of the simulated annealing. The learning step is not present in basic versions, but many existing variants have tried to link the temperature to certain characteristics of the sampling obtained through the Metropolis method [25, 46, 19].

Sampling: direct.

Learning: relational mechanisms between the set of points and the temperature.

Diversification: sampling of the objective function through the Metropolis method.

Intensification: temperature decrease.

Algorithm 6. ALS model for the simulated annealing

Simulated annealing cannot be considered as a metaheuristic using swarm intelligence operators. Indeed, the behavior of the system is defined by a global rule (the Metropolis method), without any use of local interactions. Finally, the simulated annealing is mainly characterized by its direct sampling of the objective function. The mechanism behind this algorithm is one of the most common to all the metaheuristics and should thus be underlined.

3.2 Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDA) were first created as an alternative to evolutionary algorithms [44]: the main difference is that crossover and mutation steps are replaced by the choice of random individuals with respect to an estimated distribution obtained from the previous populations. The general process is presented in Algorithm 7.

The main difficulty is how to estimate the distribution; the algorithms used for this are based on an evaluation of the dependency of the variables, and can belong to three different categories:

1. Models without any dependency: the probability distribution is factorized from univariant independent distributions, over each dimension. That choice has the defect not to be realistic in case of hard optimization, where a dependency between variables is often the rule.
2. Models with *bivariant* dependency: the probability distribution is factorized from bivariant distributions. In this case, the learning of distribution can be extended to the notion of *structure*.

3. Models with *multiple* dependencies: the factorization of the probability distribution is obtained from statistics with an order *higher* than two.

```

 $D_0 \leftarrow$  Randomly generate  $M$  individuals.
 $i = 0$ 
While stopping criteria:
     $i = i + 1$ 
     $D_{i-1}^{Se} \leftarrow$  Select  $N \leq M$  individuals in  $D_{i-1}$  using the selection method.
     $p_i(x) = p(x | D_{i-1}^{Se}) \leftarrow$  Estimate the probability distribution of the selected individuals.
     $D_i \leftarrow$  Sample  $M$  individuals from  $p_i(x)$ 
End

```

Algorithm 7. Estimation of distribution algorithm

For continuous problems, the distribution model is often based on a normal distribution. Some important variants were proposed, using for example "data clustering" for multimodal optimization, parallel variants for discrete problems (see [39]). Convergence theorems were also formulated, in particular with modeling by Markov chains, or dynamic systems. EDA algorithms in the ALS scope are modelled in Algorithm 8.

```

Using an explicit sampling.
Learning: extraction of the parameters of an explicit distribution;
Diversification: sampling of the distribution;
Intensification: selection (or dispersion reduction techniques).

```

Algorithm 8. ALS model for estimation of distribution algorithms

3.3 Particle Swarm Optimization

The particle swarm optimization ("Particle Swarm Optimization", *PSO*) [35, 36] evolved from an analogy drawn with the collective behavior of the animal displacements (in fact, the metaphor was largely derived from socio-psychology). Indeed, for certain groups of animals, e.g. the fish schools, the dynamic behavior in relatively complex displacements can be observed, where the individuals themselves have access only to limited information, like the position and the speed of their closer neighbors. For example, it can be observed that a fish school is able to avoid a predator in the following manner: initially it gets divided into two groups, then the original school is reformed, while maintaining the cohesion among the school.

The authors, who proposed the method of particle swarm optimization, drew their original inspiration by first comparing the behaviors in accordance with the theory of socio-psychology for data processing and the decision-making in social groups, side by side. It is an exceptional and remarkable achievement that this metaheuristic was originally conceived for the continuous domain, and, till date, majority of its applications are in this domain. The method conceives a large group of *particles*, in the form of vectors, moving in the search space. Each particle i is characterized by its *position* \vec{x}_i and a vector of change in position (called *velocity*) \vec{v}_i . In each iteration, the movement of the particle can be characterized as: $\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t-1)$. The core of the method consists in the manner in which \vec{v}_i is chosen, after each iteration. Socio-psychology suggests that the movements of the individuals (in a socio-cognitive chart) are influenced by their last behavior and that of their neighbors (closely placed in the social network and not necessarily in space). Hence, the updating of the position of the particles is dependent on the direction of their movement, their speed, the best preceding position \vec{p}_i and the best position \vec{p}_g among the neighbors:

$$\vec{x}_i(t) = f(\vec{x}_i(t-1), \vec{v}_i(t-1), \vec{p}_i, \vec{p}_g)$$

The change in position, in each iteration, is thus implemented according to the following relation:

$$\begin{cases} \vec{v}_i(t) = \vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)) \\ \vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t-1) \end{cases}$$

where the φ_n parameters are drawn randomly from the discourse $U_{[0, \varphi_{max}]}$ and are influential in striking a balance between the relative roles of the *individual experience* (governed by φ_1) and of the *social communication* (governed by φ_2). Uniform random selection of these two parameters is justified from the fact that it does not give any a priori importance to any of the two sources of information. The algorithm also employs another parameter, V_{max} , to limit the rapidity of movement in each dimension, so that it can prevent any "explosion" of the system, in case there are too large amplifications of the oscillations.

The algorithm could implement an effective compromise between intensification and diversification. The only problem arises when the points p_i and p_g move apart, in that case the particles will continue to oscillate between these two points without converging. An interesting characteristic of this algorithm is that, if a new optimum is discovered after the algorithm converged (i.e., after a phase of intensification), the particles will explore the search space around the new point (i.e. a phase of diversification).

The ALS modelling of this generic scheme is presented in Algorithm 13.

Sampling: implicit.

Learning: weight accorded to the best solutions and spreading of this information in the swarm.

Diversification: expansion of the swarm.

Intensification: contraction of the swarm around a point.

Algorithm 9. ALS model for particle swarm optimization

In this algorithm, the positive feedbacks are situated at the level of the particles attraction. The moves limitations of each particle form the negative feedbacks. There is a memory situated at the local level, between neighbor particles, as each one does only move according to the state of its closest neighbors, and not according to the whole system. The readers are redirected to read [36] to obtain a detailed, state of the art, understanding of the particle swarm optimization and the concepts associated with it and [12] for a synthesis.

3.4 Evolutionary Algorithms

Evolutionary algorithms [23] are inspired from the biological process of the adaptation of alive beings to their environment. The analogy between an optimization problem and this biological phenomenon has been formalized by several approaches [31, 26, 49], leading for example to the famous family of genetic algorithms [27]. The term *population* metaheuristics fits particularly well; following the metaphor, the successive populations are called *generations*. A new generation is computed in three stages, detailed below.

1. Selection: improves the reproduction ability of the best adapted individuals.
2. Crossover: produces one or two new individuals from their two parents, while recombining their characteristics.
3. Mutation: randomly modifies the characteristics of an individual.

One clearly identifies the third step with the diversification stage, while the first one stands for the intensification. We interpret the crossover as a learning from the previous information (*i.e.* from the ancestors). Several methods [52, 29, 28, 5] were designed for the diversification operators, which emphasize the implicit process of distribution sampling.

The ALS modelling of this generic scheme is presented in Algorithm 13.

<p>Sampling: implicit.</p> <p>Learning: crossover.</p> <p>Diversification: mutation.</p> <p>Intensification: selection.</p>

Algorithm 10. ALS model for evolutionary algorithms

In this family of metaheuristics, feedback processes are sometimes difficult to figure out, as there are many variants. Generally speaking, the positive feedbacks are situated on selection operators, whereas negative feedbacks are typically implemented in mutation operators. There is a form of local memory, as the evolution of each individual at each iteration is linked to the evolution of its neighbors.

3.5 Immune Systems

The term "artificial immune systems" (*AIS*) is applicable for a vast range of different systems, in particular for metaheuristic optimization, inspired by the operation of the immune system of the vertebrates. A great number of systems have been conceived in several varied fields e.g. robotics, the detection of anomalies or optimization (see [18] for a detailed exploration of various applications).

The immune system is responsible for the protection of the organism against the "aggressions" of external organisms. The metaphor from which the *AIS* algorithms originate harps on the aspects of *training* and *memory* of the immune system known as *adaptive* (in opposition to the system known as *innate*), in particular by discriminating between *self* and *non-self*.

1. Generate a collection of solutions P composed of an entire collection of cell memories P_M added to the present population P_r : $P = P_M + P_r$;
2. Determine the n best cells P_n from the population P , which is based on the measure of affinity;
3. Clone n individuals to form a population C . The number of clones produced for each cell is a function of affinity;
4. Implement a hyper-mutation process for the clones, which thus generates a population C^* . The mutation is proportional to affinity;
5. Select the individuals C^* to form the memory population P_M ;
6. Replace the worst individuals in P to form P_r ;
7. If a termination criterion is not reached, return to 1.

Algorithm 11. A simple example of the algorithm of *artificial immune system*

The principal ideas used for the design of this metaheuristic are the selections operated on the lymphocytes accompanied by the positive feedback, allowing the multiplication and the implementation of memory by the system. Indeed, these are the chief characteristics to maintain the self-organized characteristics of the system.

The approach used in the *AIS* algorithms is very similar to that of the evolutionary algorithms but was also compared with that of the neural networks. Within the framework of difficult optimization, the *AIS* can be regarded to take the shape of evolutionary algorithm, introducing particular operators. To operate the selection, it has to be based, for example, on a measurement of affinity (i.e. between the receiver of a lymphocyte and an antigen). The process of mutation takes place through an operator of hyper-mutation, resulting directly from the metaphor. In the final analysis, the algorithm developed is very close to a genetic algorithm (see algorithm 11.).

The ALS modelling of this generic scheme is presented in Algorithm 12.

Sampling: implicit.
Learning: memory.
Diversification: mutation.
Intensification: selection.

Algorithm 12. ALS model for immune systems

A description of the basic theory and many applications of the artificial immune systems can be found in [17], [18] and in [16], and also in a book of reference [15].

3.6 Ant Colony Algorithms

An elegant description of ant colony algorithms was proposed in [20], which can be applied to the (combinatorial) problems where a partial construction of the solution is possible. This description, although restrictive, makes it possible to highlight the original contributions of these metaheuristics (called *ACO*, for "Ant Colony Optimization", by the authors).

Artificial ants used in ACO are stochastic solution construction procedures that probabilistically build a solution by iteratively adding solution components to partial solutions by taking into account (i) heuristic information on the problem instance being solved, if available, and (ii) (artificial) pheromone trails which change dynamically at run-time to reflect the agents' acquired search experience.

A more precise formalization exists [20]. It develops a *representation* of the problem on the basis of a basic *behavior* of the ants and a general *organization* of the metaheuristic under consideration. Several concepts have also been laid down to facilitate the understanding of the principles of these algorithms, in particular the definition of the trails of pheromone as an *adaptive memory*, the need for an adjustment of *intensification /diversification* and finally, the use of a *local search*.

The problem is represented by a *set of solutions*, an *objective function* assigning a value for each solution and a *set of constraints*. The objective is to find the global optimum satisfying the constraints. The various states of the problem are characterized similarly to a sequence of components. It should be noted that, in certain cases, a cost can be associated to the states which do not belong to the set of solutions. In this representation, the ants build solutions while moving on a graph $G = (C, L)$, where the nodes are the components of C and the set L connects the components of C . The constraints of the problem are implemented directly in the rules of displacement of the ants (either by preventing the movements which violate the constraints, or by penalizing such solutions).

The movements of the ants can be characterized like a stochastic procedure of *building* constructive solutions on the graph $G = (C, L)$. In general, the ants try to work out feasible solutions, but if necessary, they can produce unfeasible solutions. The components and the connections can be associated with the trails of pheromone τ (establishing an adaptive memory describing the state of the system) and a heuristic value η (representing a priori information about the problem, or originating from a source other than that of the ants; it is very often the cost of the state in progress). The trails of pheromone and the value of the heuristics can be associated either with the components, or with the connections.

Each ant has a memory to store the path traversed, an initial state and the stopping conditions. The ants move according to a probabilistic rule of decision function of the local *trails* of pheromone, *state* of the ant and *constraints* of the problem. At the time of addition of a component to the solution in progress, the ants can update the trail associated with the component or the corresponding connection. Once the solution is built, they can update the trail of pheromone components or connections used. Lastly, an ant has the capacity of at least building a solution for the problem.

In addition to the rules governing the behavior of the ants, another major process is activated: the *evaporation* of the trails of pheromone. In fact, with each iteration, the value of the trails of pheromone is *decreased*. The goal of this reduction is to avoid a too fast

convergence and the trapping of the algorithm in local minima. This causes a gradual lapse in memory which helps in exploration of new areas.

According to the authors of the *AGO* formalism, it is possible to implement other processes requiring a *centralized* control (and thus not being able to be directly controlled by some ants), as additional processes. In our opinion, this is not desirable; in fact, one then loses the decentralized characteristic of the system. Moreover, the implementation of the *additional* processes with rigorous formalization becomes difficult, because one should be able to view any process there.

The use of the *stigmergy* is a crucial factor for the ant colony algorithms. Hence, the choice of the method for implementation of the trails of pheromone is significant to obtain the best results. This choice is mainly related to the possibilities of *representation* of the search space, each representation being able to bring a different way to implement the trails. For example, for the traveling salesman problem, an effective implementation consists in using a trail τ_{ij} between two cities i and j like a representation of the *interest* to visit the city j after the city i . Another possible representation, less effective in practice, consists in considering τ_{ij} as a representation of the interest to visit i as the j th city. In fact, the trails of pheromone describe the state of the search for the solution by the system in each iteration and the agents modify the way in which the problem will be *represented* and perceived by the other agents. This information is shared by the ants by means of modifications of the *environment*, in form of an indirect communication: the stigmergy.

The structure of ant colony metaheuristics comprises of an *intrinsic parallelism*. Generally, the good quality solutions emerge as a result of the indirect *interactions* taking place inside the system, not of an explicit implementation of exchanges. Here each ant takes only the local information about its environment (the trails of pheromones) into account; it is thus very easy to parallel such an algorithm. It is interesting to note that the various processes in progress in the metaheuristic (i.e. the behavior of the ants, evaporation and the additional processes) can also be implemented independently, the user has the liberty to decide the manner in which they will interact.

Sampling: implicit.

Learning: memory, construction of the model.

Diversification: random search for new solution components.

Intensification: evaporation, heuristics.

Algorithm 13. ALS model for ant colony algorithms

4. Conclusion

Population metaheuristics can be viewed as algorithms handling a probabilistic sampling of a probability distribution, representing the objective function of an optimization problem. These algorithms can be described either as implicit, explicit or direct, according to their way of sampling the objective function. These algorithms are iteratively manipulating the sample thanks to components that can be classified among three tendencies: learning,

intensification and diversification. These metaheuristics can thus be viewed as adaptive learning search algorithms.

A lot of the stochastic metaheuristics make use of swarm intelligence to design efficient components that can solve a large scale of different hard optimization problems. Among them, implicit metaheuristics like evolutionary computation or particle swarm optimization are the most known for their self-organized aspects.

These two theories are thus complementary and, from the point of view of the design of metaheuristics, there is a simple relation between them: the ALS describes the "goal" to be reached, and the theory of the swarm intelligence a "means" to reach this goal. So, an effective metaheuristic should, according to the adaptive learning search, set up mechanisms of learning, intensification and diversification, stays the question of the means to be used to set up these mechanisms. The swarm intelligence proposes a model of implementation: an algorithm on base of population defining simple interactions at the local level, allowing the emergence of a complex behavior at the global level.

Both presented theories should allow to better understand the functioning of existing metaheuristics and to direct the design of new ones. The concepts important to retain are the use by modern metaheuristics of learning, intensification and diversification, as well as the distributed aspect and the flexible hose of the swarm intelligence. However it is necessary to underline the difficulty to design a swarm intelligence system, what explains that the inspiration comes from the biology, where such systems are relatively common. The main difficulties are the following ones:

- Design sampling operators from which it is easy to extract the relevant information to direct the search,
- Set the balance between techniques of intensification, diversification and learning,
- Maintain the flexibility of the algorithm, so that it adapts itself to the problem.

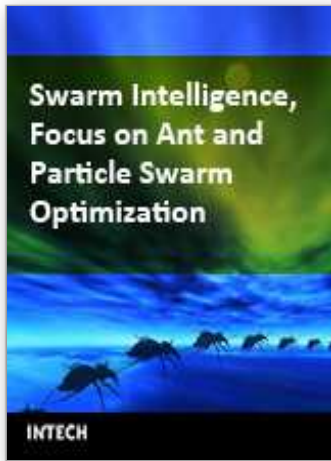
5. Bibliography

- E. H. L. Aarts and P. J. M. Van Laarhoven. Statistical cooling : a general approach to combinatorial optimisation problems. *Philips J. of Research*, 40:193-226, 1985. [1]
- S. Baluja. Population-based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. *Technical Report CMU-CS-94-163*, Carnegie Mellon University, 1994. [2]
- S. Baluja. Genetic Algorithms and Explicit Search Statistics. *Advances in Neural Information Processing Systems*, 9:319-325, 1997. [3]
- S. Baluja and R. Caruana. Removing the Genetics from the Standard Genetic Algorithm. In A. Frieditis and S. Russel, editors, *International Conference on Machine Learning*, pages 38-46, Lake Tahoe, California, 1995. Morgan Kaufmann. [4]
- S. Baluja and S. Davies. Fast probabilistic modeling for combinatorial optimization. In *Fifteenth National Conference on Artificial Intelligence, Tenth Conference on Innovative Applications of Artificial Intelligence*, Madison, Wisconsin, 1998. [5]
- E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence, From Natural to Artificial Systems*. Oxford University Press, 1999. [6]
- P. A. N. Bosnian and D. Thierens. An algorithmic framework for density estimation based evolutionary algorithm. Technical Report UU-CS-1999-46, Utrech University, 1999. [7]

- P.A.N. Bosnian and D. Thierens. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In M. Muehlenbein and A.O. Rodriguez, editors, *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2000*, pages 197–200, San Francisco, California, 2000. Morgan Kaufmann. [8]
- P.A.N. Bosnian and D. Thierens. IDEAs based on the normal kernels probability density function. Technical Report UU-CS-2000-11, Utrecht University, 2000. [9]
- S. Camazine, J.L. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2000. [10]
- V. Cerny. Thermodynamical approach to the traveling salesman problem : an efficient simulation algorithm. *J. of Optimization Theory and Applications*, 45(1):41-51, 1985. [11]
- M. Clerc. L'optimisation par essaim particulaire : principes, modeles et usages. *Technique et Science Informatiques*, 21:941–964, 2002. [12]
- A. Colomi, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In F. Varela and P. Bourguine, editors, *Proceedings of ECAL'91 -First European Conference on Artificial Life*, pages 134-142, Paris, France, 1992. Elsevier Publishing. [13]
- M. Creutz. Microcanonical Monte Carlo simulation. *Physical Review Letters*, 50(19):1411-1414, May 1983. [14]
- D. Dasgupta. *Artificial Immune Systems and their applications*. Springer Verlag, 1999. [15]
- D. Dasgupta and N. Attoh-Okine. Immune-based systems: A survey. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 369–374, Orlando, October 1997. IEEE Press. [16]
- L.N. De Castro and F. Von Zuben. Artificial Immune Systems: Part I: Basic Theory and Applications. Technical Report TR-DCA 01/99, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil, December 1999. [17]
- L.N. De Castro and F. Von Zuben. Artificial Immune Systems: Part II -A Survey of Applications. Technical Report DCA-RT 02/00, Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, State University of Campinas, Brazil, February 2000. [18]
- P. M. C. De Oliveira. Broad Histogram : An Overview, *arxiv :cond-mat/0003300v1*, 2000. [19]
- M. Dorigo and T. Stützle. *Handbook of Metaheuristics*, volume 57 of *International series in operations research and management science*, chapter The Ant Colony Optimization Metaheuristics: Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound, January 2003. [20]
- J. Dreoo, J.-P. Aumasson, W. Tfaili, and P. Siarry. Adaptive learning search, a new tool to help comprehending metaheuristics. *International Journal on Artificial Intelligence Tools*, 16(3), June 2007. [21]
- J. Dreoo, A. Petrowski, P. Siarry, and E. D. Taillard. *Metaheuristics for hard optimization*. Springer, 2006. [22]
- J. E. Eiben, A. E. Smith. *Introduction to Evolutionary Computing*. Springer Verlag, 2003. [23]
- J. Ferber. *Les systemes multi-agents. Vers une intelligence collective*. In-terEditions, 1997. [24]

- A. M. Ferrenberg and R. H. Swendsen. Optimized Monte Carlo Data Analysis. *Phys. Rev. Lett*, 63:1195, 1989. [25]
- L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. Wiley, 1966. [26]
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley, 1989. [27]
- G. Harik. Linkage learning in via probabilistic modeling in the EcGA. *Technical Report 99010, IlliGAL*, 1999. [28]
- G. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. In *IEEE Conference on Evolutionary Computation*, pages 523-528, 1998. [29]
- W. K. Hastings. Monte Carlo sampling method using Markov chains and their applications. *Biometrika*, 57, 1970. [30]
- J. H. Holland. Outline for logical theory of adaptive systems. *J. Assoc. Comput. Mach.*, 3:297-314, 1962. [31]
- K. Hukushima and K Nemoto. Exchange Monte Carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.*, 65:1604-1608, 1996. [32]
- Y. Iba. Population Annealing: An approach to finite-temperature calculation. In *Joint Workshop of Hayashibara Foundation and SMAPIP*. Hayashibara Forum, 2003. [33]
- N. R. Jennings. Coordination Techniques for Distributed Artificial Intelligence. In G. M. P. O'Hare and N. R. Jennings, editor, *Foundations of Distributed Artificial Intelligence*, pages 187-210. John Wiley & Sons, 1996. [34]
- J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. on Neural Networks*, volume IV, pages 1942-1948, Piscataway, NJ: IEEE Service Center, 1995. [35]
- J. Kennedy, R.C. Eberhart, and Y. Shi. *Swarm Intelligence*. Evolutionary Computation. Morgan Kaufmann, April 2001. [36]
- S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671-680, 1983. [37]
- W. Krauth. *Advances in Computer Simulation*, chapter Introduction To Monte Carlo Algorithms. Springer-Verlag, 1998. [38]
- P. Larranaga and J.A. Lozano. *Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation*. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, 2002. [39]
- F. Liang and W. H. Wong. Evolutionary Monte Carlo: Application to C_p Model Sampling and Change Point Theorem. *Statistica Sinica*, 10, 2000. [40]
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087-1092, 1953. [41]
- N. Monmarche, E. Ramat, G. Dromel, M. Slimane, and G. Venturini. On the similarities between AS, BSC and PBIL: toward the birth of a new meta-heuristics. E3i 215, Universite de Tours, 1999. [42]
- N. Monmarche, N. Ramat, L. Desbarat, and G. Venturini. Probabilistic search with genetic algorithms and ant colonies. In A.S. Wu, editor, *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop*, pages 209-211, 2000. [43]

- H. Miihlenbein and G. Paa/3. From recombination of genes to the estimation of distributions I. Binary parameters. *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature*, PPSN IV:178-187, 1996. [44]
- J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965. [45]
- M. E. J. Newman and R. G. Palmer. Error estimation in the histogram Monte Carlo method. *arxiv:cond-mat/98043006*, 1998. [46]
- G. Nicolis and I. Prigogine. *Self-organization in Non-equilibrium Systems*. New York, 1977. [47]
- Y Okamoto and U. H. E. Hansmann. Thermodynamics of helix-coil transitions studied by multicanonical algorithms. *J. Phys. Chem.*, 99:11276– 11287, 1995. [48]
- I. Rechenberg. *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment Library Translation, 1965. [49]
- M.G.C. Resende. Greedy randomized adaptive search procedures (GRASP). Technical Report TR 98.41.1, AT&T Labs-Research, 2000. [50]
- Christian Blum Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268-308, September 2003. [51]
- G. Syswerda. Simulated Crossover in Genetic Algorithms. In L. D. Whitley, editor, *Second workshop on Foundations of Genetic Algorithms*, pages 239-255, San Mateo, California, 1993. Morgan Kaufmann. [52]
- E. D. Taillard, L. M. Gambardella, M. Gendreau, and J.-Y. Potvin. Adaptive Memory Programming: A Unified View of Meta-Heuristics. *European Journal of Operational Research*, 135(1): 1–16, 1998. [53]
- E. Triki, Y. Collette, and P. Siarry. A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. *European Journal of Operational Research*, 166:77-92, 2005. [54]
- O. Wendt and W. Konig. Cooperative Simulated Annealing: How Much Cooperation is Enough ? Technical Report 97-19, Institute of Information Systems, Goethe University, Frankfurt, 1997. [55]



Swarm Intelligence, Focus on Ant and Particle Swarm Optimization

Edited by Felix T.S. Chan and Manoj Kumar Tiwari

ISBN 978-3-902613-09-7

Hard cover, 532 pages

Publisher I-Tech Education and Publishing

Published online 01, December, 2007

Published in print edition December, 2007

In the era of globalisation, the emerging technologies are governing engineering industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from nature and implant them in the engineering sciences. This way of thinking led to the emergence of many biologically inspired algorithms that have proven to be efficient in handling computationally complex problems with competence, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms, the present book on "Swarm Intelligence: Focus on Ant and Particle Swarm Optimization" aims to present recent developments and applications concerning optimization with swarm intelligence techniques. The papers selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. In addition to the introduction of new concepts of swarm intelligence, this book also presented some selected representative case studies covering power plant maintenance scheduling; geotechnical engineering; design and machining tolerances; layout problems; manufacturing process plan; job-shop scheduling; structural design; environmental dispatching problems; wireless communication; water distribution systems; multi-plant supply chain; fault diagnosis of airplane engines; and process scheduling. I believe these 27 chapters presented in this book adequately reflect these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Johann Drezo and Patrick Siarry (2007). Stochastic Metaheuristics as Sampling Techniques using Swarm Intelligence, *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, Felix T.S. Chan and Manoj Kumar Tiwari (Ed.), ISBN: 978-3-902613-09-7, InTech, Available from:
http://www.intechopen.com/books/swarm_intelligence_focus_on_ant_and_particle_swarm_optimization/stochastic_metaheuristics_as_sampling_techniques_using_swarm_intelligence

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.