Monte Carlo Simulation and Population-Based Optimization

Alain Cercueil Olivier François
Laboratoire de Modélisation et Calcul,
Institut IMAG,
BP 53, 38041 Grenoble cedex 9,
FRANCE
Alain.Cercueil@imag.fr, Olivier.Francois@imag.fr

Abstract- This paper briefly reviews some properties of Monte Carlo simulation and emphasizes the link to evolutionary computation. It shows how this connection can help to study evolutionary algorithms within a unified framework. It also gives some practical examples of implementation inspired from MOSES (the mutation-or-selection evolution strategy).

1 Introduction

Monte Carlo simulation consists of generating samples from a probability distribution. This technique is popular in statistical mechanics where probability distributions are usually related to energy functionals via the Boltzmann-Gibbs formalism [24].

In optimization, evolutionary algorithms find global minima of a fitness function f on a search space S. The paradigm presented in this paper is that evolutionary algorithms can be viewed as Monte Carlo methods intended to sample from a probability distribution defined on the trajectories of their populations. This perspective helps unifying convergence theories for evolutionary algorithms and provides insights on convergence rates for these algorithms.

The paper is organized as follows. Section 2 briefly reviews Monte Carlo methods in statistical mechanics such as the Metropolis dynamics and Markov random fields. Section 3 extends this simulation setting to evolutionary computation, showing how classical algorithms can be placed into this framework. This section also revisits evolution strategies from a statistical mechanics viewpoint. In Section 4, two algorithms are proposed whose dynamics are controlled by cost functionals that are proportional to the number of fitness evaluations at each generation. Simulation results are presented in Section 5.

2 Monte Carlo methods in statistical mechanics

There is a deep connection between the behavior of complex systems in thermal equilibrium at finite temperature and multivariate optimization. In these systems, each configuration is weighted by its Boltzmann probability factor $\exp(-E/T)$, where E is the energy of the configuration and T the temperature. Finding the low-temperature state of a system when the energy can be computed amounts to solving an optimization problem. This connection has been used to devise the simulated annealing algorithm [15]. Simulated annealing is based upon the Metropolis procedure, which consists of a Markov Chain Monte Carlo method that produces a simulation of the system in equilibrium at a given temperature. In each step of the algorithm, the configuration is given a random perturbation and the resulting change in energy ΔE is computed. Two situations may happen. If $\Delta E \leq 0$, the modified configuration is more likely than the original one and the change is accepted. The case $\Delta E > 0$ is treated probabilistically. The probability that the modified configuration is accepted is $\exp(-\Delta E/T)$. By repeating the basic steps many times, the simulated configuration evolves into a Boltzmann distribution.

Many other Markovian strategies enable the Boltzmann distribution to be simulated. Of these strategies, Glauber's procedure has received the most attention [14, 16]. Again, it is based upon the computation of the change in energy after a small perturbation of the system configuration. However, both cases $\Delta E \leq 0$ and $\Delta E > 0$ are treated probabilistically. The acceptance probability is merely equal to

$$p_{accept} = \frac{1}{1 + \exp(\Delta E/T)}.$$

Monte Carlo procedures such as Metropolis or Glauber's dynamics provide approximate samples from the Boltzmann distribution. (Perfect simulation is not discussed here.) Choosing the strategy that yields the best approximation is difficult in general as this choice depends on the energy landscape and the temperature [12].

Another connection between statistical mechanics and Markov chain simulation arises from Markov random field (MRF) theory. As a simplified example, consider a sequence of atoms in one dimension whose spins x_i are allowed to point "up" ($x_i = +1$) or "down" ($x_i = -1$). The interaction between two adjacent spins can be written $-Jx_ix_{i+1}$. The contribution to the total energy is $-J$ if x_i and x_{i+1} are aligned, otherwise it is equal to $+J$. For any sequence of length r, the energy can be computed as

$$E = \sum_{i=1}^{r-1} C(x_i, x_{i+1}), \tag{1}$$

where $C(x_i, x_{i+1})$ corresponds to the interaction $-Jx_ix_{i+1}$. The one-dimensional structure of the sequence enables the Boltzmann distribution to be viewed as a MRF [13]. In this perspective, the cost $C(x_i, x_{i+1})$ is associated with a transition probability between states x_i and x_{i+1}. In the example, such a transition matrix can be described as

$$P = \begin{pmatrix} p(-,-) & p(-,+) \\ p(+,-) & p(+,+) \end{pmatrix}$$

Each component can be computed easily:

$$p(-,-) = p(+,+) \propto e^{+J/T}$$

and

$$p(-,+) = p(+,-) \propto e^{-J/T}$$

which implies

$$p(x_i, x_{i+1}) \propto \exp(-C(x_i, x_{i+1})/T).$$

Monte Carlo simulation from the transition matrix P can therefore be used to produce samples from the Boltzmann distribution associated to the energy E.

3 Population-based optimization and stochastic simulation

This section presents a Monte Carlo simulation framework for evolutionary optimization processes that parallels the statistical mechanics approach.

3.1 The simulation setting

Evolutionary optimization procedures simulate complex behaviors inspired from nature. Several attempts have been made in trying to relate statistical mechanics and evolutionary computation. Davis and Principe emphasized the relationship existing between the mutation probability and a temperature for the simple genetic algorithm [5]. Suzuki [26, 27] analyzed Markov chain models of this algorithm using the analogy with simulated annealing. Rudolph [21], and Mahfoud and Goldberg [18] compared massively parallel simulated annealing and genetic algorithms. Van Nimwegen *et al.* [19] studied meta-stability in the royal road genetic algorithm deeply with low mutation probabilities. The connection has been investigated by Cerf in a series of papers [3, 4] for a genetic algorithm with Boltzmann roulette selection.

Trying to associate energy functionals or random fields with fitness landscapes is a natural approach [25]. However, this corresponds to a sequential optimization perspective. As soon as the objective function is identified as an energy, Monte Carlo optimization usually involves the iterative improvement of a single solution. Yet it is unclear how Metropolis-like methods could be based upon multiple interacting solutions.

On the other hand, trajectories of evolutionary optimization processes can be described by means of few operators, such as mutation, selection, or crossover. This paper emphasizes the representation of transitions between successive generations thanks to energy functionals in the same way as it has been defined for successive spins in the MRF model. Therefore, evolutionary processes can be viewed as Monte Carlo procedures that merely consist of simulating the Markov chains associated with such functionals. In this perspective, some parameters must play the role of a temperature. Many parameters provide some analogy with thermal fluctuations in evolutionary algorithms. These parameters are the mutation probability p_{mut}, the crossover probability p_c, the population size, the variance for Gaussian mutation, and so forth.

Let X be the set of all populations of size n, and consider a sequence $x_1, \ldots, x_r$ in X specified by a sequence of costs $C(x_i, x_{i+1})$. Then, the cost of the sequence is defined as the sum of the elementary transition costs as in equation (1). The only difficulty that a Monte Carlo procedure has to solve in order to prescribe the simulation of an evolutionary process is building a

Markov chain such that

$$p(x_i, x_{i+1}) \propto \exp(-C(x_i, x_{i+1})/T).$$

According to the previous section, this amounts to defining Boltzmann distributions on sequences of populations. Often, an exact proportionality relationship can be hard to obtain. However, many results remain unchanged whenever a rough relationship (i.e., involving only log-scales) can be identified

$$\log p(x_i, x_{i+1}) \sim -C(x_i, x_{i+1})/T.$$

3.2 Previous works

This section and the next show how evolutionary algorithms can be placed into the Monte Carlo simulation framework. In a simple evolutionary algorithms, the mutation probability is usually considered as being the disorder parameter. It can be related to a temperature as follows

$$p_{mut} = e^{-1/T}.$$

Note that p_{mut} represents the "global" probability that a string or an individual undergoes a mutation, but bitwise probabilities could be considered as well. The algorithm studied by Cerf can be described as follows. Start from a population $x = x_0 \in X$ and repeat the following steps. For each individual x^i, $i = 1, \ldots, n$, create one offspring ζ^i by mutation with probability p_{mut}. Otherwise, take ζ^i equal to x^i. Then, create a sample $(y^i)_{i=1,\ldots,n}$, $y^i \in \{\zeta^1, \ldots, \zeta^n\}$ according to

$$\mathrm{P}(Y = \zeta^i) \propto \exp(-\theta f(\zeta^i)/T), \quad \theta > 0. \qquad (2)$$

Replace x^i by y^i, for all $i = 1, \ldots, n$.

This description corresponds to an evolutionary algorithm without crossover and using *Boltzmann roulette* selection. The vector ζ represents the intermediate population vector obtained after the mutation operator has been applied. The parameter θ represents the selection intensity and allows the respective weights of mutation and selection to be balanced. A Markov chain model can be associated to the cost

$$C(x,y) = \min_{\zeta} \sum_{i=1}^{n} 1 - \delta(x^i, \zeta^i) + \theta(f(y^i) - f(\zeta^*)),$$

when the transition $x \to y$ is admissible and $\delta(a,b)$ denotes the Kronecker symbol. (Admissibility depends on the way the mutation operator has been implemented.) The contribution $\sum_{i=1}^{n} 1 - \delta(x^i, \zeta^i)$ merely counts the number of offspring by mutation. Each selected individual y^i contributes to $\theta(f(y^i) - f(\zeta^*))$, and the cost is zero when the best offspring is selected (ζ^* is the best individual in the population ζ). Cerf's theory on the "Boltzmann" GA proposes temperature schedules that guarantee the convergence on an optimal solution [3].

Few authors have proposed the population size as representing the disorder parameter. Some algorithms exist that rely on adaptive population sizes [22], [2]. Such algorithms have proved to be efficient since computational efforts can be decreased as good solutions appear [7].

François [10] proposed an algorithm where the population size is random and can be controlled. The algorithm MOSES can be described as follows.

- Initialize a population x of n labelled individuals.
- Repeat
  1. Draw a random number N from the binomial distribution $bin(n, p)$.
  2. Select the best individual from the population
  $$x^* = \text{argmin}\ \{f(x^i);\ i = 1, \ldots, n\}. \quad (3)$$
  3. Create offspring of the N first individuals and replace the parents by the offspring. Replace the $n - N$ other individuals by x^*.
  4. Adapt the parameters p.

At generation t, the effective population size is $N_t + 1$, whose average size is

$$n_t = E[N_t + 1] = 1 + np_t.$$

This algorithm has been studied under the assumption that the search space is a finite set (that usually contains a huge number of solutions), and that this search space is endowed with a graph structure. This structure enables the mutation operator to be defined as a step of a random walk on the graph. Continuous problems can be transformed into discrete ones so that the discretization matches this formalism. Such a discretization may correspond to lattice structures when floating-point representation is used. But finite string representation using binary coding can be considered as well.

Choosing $p = e^{-1/T}$ enables the Boltzmann formalism to be considered again. Therefore, cooling schedules make the population converge to a single individual. The algorithm can be associated with the cost

$$C(x,y) = \sum_{i=1}^{n} 1 - \delta(x^*, y^i). \qquad (4)$$

This cost counts those individuals in y that differ from x^*, and corresponds to the number of fitness evaluations needed to implement the transition between populations x and y.

For small p, the mean hitting time of the optimal solution is in the worst case [10], [11]

$$\log E[\tau_*] \sim -d_* \log p, \quad (5)$$

where

$$d_* = \max_{a \neq a^*} \min_{b: f(b) < f(a)} d(a, b), \quad (6)$$

and $d(a, b)$ is the minimal number of mutation steps needed to reach a from b in the search space S. As a consequence of this result, the optimal decreasing schedule for population sizes can be obtained as follows

$$n_t = 1 + \frac{n}{t^{1/d_*}}, \quad t \geq 1.$$

3.3 A new look at the $(1, \lambda)$-ES

This section presents another example where statistical mechanics ideas can shed light on the behavior of a classical algorithm: the $(1, \lambda)$-evolution strategy. This algorithm can be defined as follows [20, 23].

- Initialize a single individual x in $X (= S)$.
- Repeat
    1. Create λ offspring according to a Gaussian distribution

    $$\zeta^i = x + \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, \lambda. \quad (7)$$

    2. Select the best individual ζ^* from the population $\{\zeta^1, \ldots, \zeta^\lambda\}$.
    3. Replace x by ζ^*.
    4. Adapt the parameter σ.

This algorithm has received a lot of theoretical study, and many adaptive rules have been proposed for the parameter σ^2 [6]. Of these adaptive schedules, the most famous is Rechenberg's $1/5$-rule that consist of modifying the variance according to the rate of success of the algorithm [20]. However, this rule relies upon the exact analysis of only two solvable problems (the sphere model and the corridor model) and has often been criticized.

In the statistical mechanics approach, the variance can be considered as a disorder parameter, and can be viewed as (half) a temperature

$$T = 2\sigma^2, \quad (8)$$

and new (and universal) adaptive schemes can be proposed. In the $(1, \lambda)$-ES, transitions between the successive solutions found by the algorithm can be expressed as follows

$$p(x, y) \propto \int_{\{\zeta : y = \arg\min f(\zeta_i)\}} \prod_{i=1}^{\lambda} \exp(-||\zeta_i - x||^2/T) d\zeta \quad (9)$$

where $||.||$ denotes the Euclidian norm in m dimensions (m is the problem dimension). Such a multidimensional integral can be estimated for small T thanks to the Laplace saddle-point method

$$\log p(x, y) \sim -C(x, y)/T \quad (10)$$

where

$$C(x, y) = \begin{cases} ||y - x||^2 & \text{if } f(y) < f(x) \\ ||y - x||^2 + (\lambda - 1)||y - x'||^2 & \text{otherwise,} \end{cases}$$

and x' is the closest solution to x whose fitness is higher than $f(y)$.

Assuming some discretization of the continuous search space as a multidimensional grid of small mesh and discarding unimodal functions, many results obtained in [11] for a non-ordered ES can be recovered easily.

Let d_* denotes the minimal Euclidian distance that separates any local (non global) minimum from a solution that outperforms this minimum. In the worst case, the mean hitting time of the optimal solution is such that

$$\log E[\tau_*] \approx d_*^2 / 2\sigma^2$$

for small σ^2. This result suggests that the optimal annealing schedule should be taken as follows

$$\sigma_t = \frac{d_*}{\sqrt{2 \log t}}, \quad t \geq 2. \quad (11)$$

4 Simulation algorithms

4.1 Non-ordered and ordered versions of MOSES

Metropolis and Glauber's dynamics are two different means of producing samples from a Boltzmann distribution. Similarly, starting from the cost representation of an evolutionary process, the way to implement an algorithm is not unique.

As an illustration, consider the costs $C(x, y)$ between populations x and y corresponding to the number of fitness evaluations required to produce y from x (the Hamming distance). In addition, assume that x^* can be produced from x at null cost (i.e., the population can be transformed into the uniform population $(x^*, \ldots, x^*)$ by a sequence of cost-less transitions). The algorithm MOSES presented in the previous section is associated with such a cost functional. This section presents two other algorithms that simulate the same energy on trajectories of an evolutionary process. The first variant can be described as follows.

- Initialize a population x of n individuals in X.
- Repeat
  1. Select the best individual x^* from the population.
  2. For $i = 1, \ldots, n$,
     - with probability p mutate individual x^i and replace x^i by its offspring,
     - otherwise replace x^i by x^*.

The second algorithm is a variant where individuals are ranked by increasing order of fitness.

- Initialize a population x of n individuals in X.
- Repeat
  1. Sort $x^1, \ldots, x^n$ according to their fitness.
  2. For $i = 1, \ldots, n$, let y^i be an offspring of x^i with probability p, and otherwise choose y^i randomly in $x^1, \ldots, x^i$.
  3. Replace x^i by y^i.

Regarding hitting times of the optimum for finite search spaces, the same results can be obtained for these variants as those described for MOSES. Such theoretical results hold as they rely on cost functionals only, and are concerned with estimates on a logarithmic scale. Regarding the non-ordered algorithm, the mean hitting time of an optimal solution can be described more precisely for small p: Let $(a^0, \ldots, a^0)$ be a uniform population of n individuals. Denote A_0 the subset of individuals whose fitness is lower than $f(a^0)$

$$A_0 = \{a \in S,\ f(a) < f(a^0)\}.$$

Denote by τ_0 the first entrance-time in A_0 starting from the uniform population (a^0). The expected entrance time can be described as follows

$$E[\tau_0] = \frac{1}{K_0} \frac{p^{d(a^0, A_0)}}{n} (1 + O(p)),$$

and its standard deviation:

$$\sigma(\tau_0) = E[\tau_0](1 + O(p))$$

where K_0 is a constant that depends on the graph structure induced by the mutation operator (a and b are neighbors iff b can be obtained from a as an offspring by mutation). The distance $d(a^0, A_0)$ is the minimal number of mutation steps needed to reach A_0 from a^0. To give a definition of K_0, consider B_0 the subset of A_0 defined as

$$B_0 = \{b \in S,\ d(a^0, b) = d(a^0, A_0)\}$$

and the subset of paths $\Gamma_{a^0, b}$ between a^0 and b for which the length is minimal. Then,

$$K_0 = \sum_{b \in B_0} \sum_{\gamma \in \Gamma_{a^0, b}} \prod_{a^i \in \gamma} \deg(a^i)^{-1}$$

where $\deg(a)$ is the degree of the vertex a in the graph.

The three algorithms are based upon a fraction of elitism at each generation. Although they share the same cost functionals and some similarities, they may exhibit quite different behaviors.

In MOSES, the selection pressure is weak for individuals with the lowest labels. This strategy enables improvements from sequences of mutations. MOSES has a hierarchical organization in which individuals with different labels perform local searches of different depths (individuals with the lowest labels undergo the longest random walks).

In the non-ordered variant, the selection pressure is high, and the algorithm favors short excursions. The population has no hierarchical structure, and more information is required from a practitioner to decide which parameter should be used.

The ordered version is less elitist. An individual that is not an offspring by mutation is not necessarily replaced by the best individual in the previous generation. Instead, the algorithm may recall individuals that were optimal in the past, and proceeds with re-exploration from these individuals.

5 Experiments

5.1 Test suite

Deciding which of the three algorithms performs the best is difficult in general. A number of test problems have been chosen to assess the performances empirically. The selection of five test problems follows the guidelines after [28].

Problem f_1 is the sphere model: It is unimodal and intended to provide an evaluation of the convergence speed of algorithms. Problem f_2 is the Rosenbroock problem: It is non-separable and usually considered as difficult for evolutionary strategies. Problem f_3 is the Rastrigin function: It is separable and highly multi-modal. Problem f_4 is the generalized Ackley function. Problems f_1, f_2, f_3, f_4 are considered in $m = 10$ dimensions. Problem f_5 is in two dimensions. In this low-dimensional problem, local minima are not arranged regularly. In each case, the fitness evaluation at the global minimum equals to zero.

$$f_1(x) = \sum_{i=1}^{m} x_i^2, \quad x_i \in [-5.12, +5.12],$$

$$f_2(x) = \sum_{i=1}^{m-1} (1 - x_i)^2 + 100(x_i^2 - x_{i+1})^2, \quad x_i \in [-30, +30],$$

$$f_3(x) = \sum_{i=1}^{m} x_i^2 - 10\cos(2\pi x_i) + 10, \quad x_i \in [-5.12, +5.12],$$

$$f_4(x) = -20\exp(-0.2\sqrt{\textstyle\sum_{i=1}^{m} x_i^2/m}) - \exp(\textstyle\sum_{i=1}^{m}\cos(2\pi x_i)/m) + 20 + e, \quad x_i \in [-32, +32],$$

$$f_5(x) = 1.03162 + 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x \in [-5, +5]^2.$$

5.2 Experimental setting

In the experiments, a continuous space version of the algorithms has been used. Individuals are represented by m-dimensional floating-point vectors. The mutation of an individual consists of adding a random variable U to a single coordinate, say a_i, chosen at random

$$b_i = a_i + U.$$

The random variable U is uniformly distributed over the interval $[-r, r]$, where r is called the *mutation radius*, and the offspring b is conditioned to stay in the definition subset. Parameter settings have been fixed to

max. pop. size	$n = 20$
mutation prob.	$p = 0.2, 0.1, 0.01$
mutation rad.	$r = 1.2$

and are common to the three algorithms. For example, with $p = 0.2$ the number of mutations in each generation is between two and six (a 0.95 confidence interval), and the average actual population size is equal to four. According to the experimental design analysis led in [9], the value $r = 1.2$ yielded good performances on a similar set of test problems. This value optimizes the trade-off between speed and accuracy for unimodal problems, and is close to the critical distance d_* for problems f_3 and f_4.

For each algorithm and each problem, mean and standard deviations have been computed from 100 repetitions of the algorithm for each parameter setting. The performance measure computed from the simulations is the number of evaluations needed to produce an acceptable solution, i.e., the algorithm is stopped when the best individual fitness is below a threshold. Since the problems are of different difficulties, different thresholds have been used. They are given in Table 1.

	f_1	f_2	f_3	f_4	f_5
threshold	0.01	0.5	0.05	0.05	0.0001

Table 1: *Fitness values below which algorithms are stopped. These values specify the empirical hitting times for all algorithms.*

5.3 Results

The results are reported in Tables 2-4. The non-ordered and the ordered of MOSES obtain the best performances on this series of artificial test problems. In addition, the non-ordered version seems to perform slightly better than the one with ranking. This indicates that a strong selection pressure ensures a better convergence rate for these test problems. As p becomes small, the differences between algorithms vanish. This behavior is consistent with the above-formulated theory that stipulates the same cost functionals for the three algorithms.

Performances on f_4 and f_5 are very similar. This can be explained as these two problems are based upon the same structure: local minima are arranged on a regular integer lattice. The average performances on the test problem f_2 are poor. This is due to the choice of the mutation operator that performs coordinate-wise changes in individuals. This choice might be unadapted when the problem is strongly epistatic. Notice that variances are very large, and that the large mean value is due to few

	f_1	f_2	f_3	f_4	f_5
Moses					
av. hit.	$2,259$	18.8e+6	$12,819$	$10,756$	$2,793$
sd. dev.	652	39.6e+6	$3,753$	$2,736$	$2,286$
Non-ordered					
av. hit.	912	9.2e+6	$4,340$	$4,233$	$1,171$
sd. dev.	222	21.1e+6	857	959	808
Ordered					
av. hit.	$1,163$	9.6e+6	$4,969$	$5,130$	$1,271$
sd. dev.	223	18.7e+6	$1,394$	988	862

Table 2: *Parameter setting:* $p = 0.2$. *Hitting times of solutions with fitness lower than the threshold given by Table 1 averaged over 100 repetitions.*

outliers. In 20% of the runs, the threshold can be attained within 200,000 fitness evaluations. This suggests that it might be better to maintain short parallel runs of the algorithm instead of a single long run.

Overall, the results indicates that using large population sizes causes an extra computational cost, as the best results are obtained for $p = 0.01$. This bias is due to the almost optimal choice of the mutation radius $r = 1.2$. Furthermore, this choice favors algorithms that exhibit strong selection pressures. In solving real-world problems, the optimal parameter setting is however generally unknown. Algorithms that use hierarchical search have proved their efficiency in solving more realistic test problems [9].

	f_1	f_2	f_3	f_4	f_5
Moses					
av. hit.	$1,500$	17.8e+6	$8,600$	$8,038$	$2,072$
sd. dev.	449	30.5e+6	$2,797$	$1,760$	$1,378$
Non-ordered					
av. hit.	698	15.2e+6	$3,714$	$3,181$	$1,033$
sd. dev.	194	28.9e+6	$1,251$	634	703
Ordered					
av. hit.	838	10.2e+6	$4,177$	$3,839$	$1,134$
sd. dev.	213	21.3e+6	$1,307$	792	869

Table 3: *Parameter setting:* $p = 0.1$. *Hitting times of solutions with fitness lower than the threshold given by Table 1 averaged over 100 repetitions.*

	f_1	f_2	f_3	f_4	f_5
Moses					
av. hit.	669	18.2e+6	$3,852$	$3,906$	$1,024$
sd. dev.	197	30.8e+6	$1,310$	905	735
Non-ordered					
av. hit.	611	NA	$3,228$	$2,639$	955
sd. dev.	189	NA	$1,092$	605	757
Ordered					
av. hit.	593	NA	$3,378$	$2,672$	920
sd. dev.	182	NA	$1,085$	625	679

Table 4: *Parameter setting:* $p = 0.01$. *Hitting times of solutions with fitness lower than the threshold given by Table 1 averaged over 100 repetitions.*

6 Conclusion

This paper presented a Monte Carlo simulation perspective on evolutionary computation: Evolutionary algorithms simulate energy functionals on trajectories of populations in a search space.

This point of view provides a unified theory that embodies simulated annealing and evolutionary algorithms. In addition, it emphasizes the role played by geometrical quantities such as inter-minima distances that are critical to the convergence of several mutation/selection algorithms.

Conversely, this representation helps devising new algorithms. Examples were given in this paper in which three algorithms were proposed that implement the same cost functionals. The algorithms are equivalent at low temperatures, but different behaviors can be observed for fixed parameter settings. On a series of artificial test problems, the best performances were obtained for the most elitist variant, and with the smallest effective population size.

Bibliography

[1] E.H.L Aarts and J.H.M Korst. Simulated annealing and Boltzmann machines, Wiley, New York, 1988.

[2] J. Arabas, Z. Michalewicz and J. Mulawka. GAVaPS-A genetic algorithm with varying population size. In Proceedings 2nd IEEE Conference on Evolutionary Computation, (1994), IEEE Press, 73-78.

[3] R. Cerf. The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré Probab. Statist.*, (1996) **32**, 455-508.

[4] R. Cerf. A new genetic algorithm. *Ann. Appl. Probab.* (1996) **6**, 778-817.

[5] T.E. Davis and J.C. Principe. A simulated annealing like convergence theory for the simple genetic algorithm. In R.K. Belew and L.B. Booker eds. *Proceedings of the fourth Int. Conf. on Genetic Algorithms*, San Mateo CA, Morgan Kauffman, 174-181, 1991.

[6] A.E. Eiben, R. Hinterding and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, (1999) **3**, 124-141.

[7] T. Bäck, A.E. Eiben and N.A.L Van der Vaart. An empirical study on GA's without parameters. *Lecture Notes in Computer Science: Parallel Problems Solving from Nature VI*, M. Schoenauer *et al.* eds, Springer-Verlag, (2000), 315-324.

[8] D.B. Fogel. Evolutionary computation: toward a new philosophy of machine intelligence, IEEE Press, Piscatawan, NJ, 1995.

[9] O. François and C. Lavergne. Design of evolutionary algorithms: A statistical perspective. *IEEE Transactions on Evolutionary Computation*, (2001) To appear.

[10] O. François. An evolutionary strategy for global minimization and its Markov chain analysis, *IEEE Transactions on Evolutionary Computation*, 2, (1998), 77-90.

[11] O. François. Global optimization with exploration/selection algorithms and simulated annealing, *Preprint*, (2000).

[12] A. Frigessi, C.R. Hwang, S.J. Sheu, P. Di Stefano. Convergence rates of the Gibbs sampler, the Metropolis algorithm, and other single-site updating dynamics. *J. Roy. Statist. Soc. Ser. B*, (1993) **55**, 205-220.

[13] H.O. Georgii. *Gibbs measures and phase transitions.* De Gruyter Studies in Mathematics, 1988.

[14] R.J. Glauber. Time dependent statistics of the Ising model, *J. Math. Phys.* (1963) **4** 294-307.

[15] S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi. Optimization by simulated annealing. *Science* (1983) 220, 671-680. 85

[16] T.M. Liggett. *Interacting Particle Systems*, Springer, New York, 1985.

[17] M. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller. Equation of state calculation by fast computing machines. *J. Chem. Phys.*, (1953)**21** 1087-1092.

[18] S.W. Mahfoud and D.E. Golberg. Parallel recombinative simulated annealing: A genetic algorithm. *Parallel Computing*, 21 (1995),1,1995.

[19] E. Van Nimwegen, J.P. Crutchfield and M. Mitchell. Statistical dynamics of the royal road genetic algorithm. Theor. Comput. Sci.,22, 9 (1999), 41-102.

[20] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Stuttgart, Frommann-Holzboog Verlag, 1973.

[21] G. Rudolph. Massively parallel simulated annealing and its relation to evolutionary algorithms, *Evolutionary Computation*, 1, 4, (1993), 361-383.

[22] R. Smith. Adaptively resizing populations: An algorithm and analysis. In S. Forrest ed., *Proceedings of the fifth Int. Conf. on Genetic Algorithms* Urbana-Champaign, Morgan Kauffman, San Mateo, CA, (1993)

[23] H.P. Schwefel. *Evolution and optimum seeking*. New-York: Wiley and Sons, 1995.

[24] A.D. Sokal. Monte Carlo methods in statistical mechanics: Foundations and new algorithms. Lectures at the Cargese summer school on "Functional integration: Basics and Applications". 1996.

[25] P. F. Stadler, R. Happel. Random field models for fitness landscapes, *J. Math. Biol.*, 38 (1999), 435-478.

[26] J. Suzuki. A Markov chain analysis on a genetic algorithm. In S. Forrest ed., *Proceedings of the fifth Int. Conf. on Genetic Algorithms* Urbana-Champaign, Morgan Kauffman, San Mateo, CA, (1993) 146-153.

[27] J. Suzuki. A further result on the Markov chain model of genetic algorithm and its application to a simulated annealing-like theory, *IEEE Trans. Systems, Man, and Cybernetics-B*, 28,1 (1998), 95-102.

[28] D. Whitley, K. Mathias, S. Rana, J. Dzubera. Building better test functions, *Proc. of the sixth International Conference on genetic algorithms*, L.J. Eshelman ed, Morgan-Kaufmann, San Mateo, CA, (1995), 512-519.