Stohastička optimizacija i metoda adaptivnog kaljenja

Radojičić, Una

J. J. Strossmayer University of Osijek
Department of Mathematics
Graduate study - Financial Mathematics and Statistics

Una Radojičić

Stochastic optimization and adaptive annealing method

Master Thesis

Osijek, 2017.

J. J. Strossmayer University of Osijek
Department of Mathematics
Graduate study - Financial Mathematics and Statistics

Una Radojičić

Stochastic optimization and adaptive annealing method

Master Thesis

Mentors: Dr. sc. Danijel Grahovac and Dr. sc. Andrew R. Barron

Osijek, 2017.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Diplomski studij financijske matematike i statistike

Una Radojičič

Stohastička optimizacija i metoda adaptivnog kaljenja

Diplomski rad

Mentor: Dr. sc. Danijel Grahovac
Komentor: Dr. sc. Andrew R. Barron

Osijek, 2017.

Contents

1 Introduction **5**

2 Sampling from a distribution **6**
2.1 Inverse sampling 6
2.2 Markov Chain Monte Carlo 7
2.2.1 Definition and basic properties of Markov chains 7
2.2.2 Metropolis-Hastings algorithm 9

3 Applying sampling from distribution to global minimization problem **12**

4 Simulated annealing **14**
4.1 Simulated annealing on a finite set S 14
4.1.1 The convergence analysis 15
4.2 Generalization of the SA for continuous set S 17

5 Adaptive Annealing **20**
5.1 The one-dimensional AA minimisation 21
5.2 Monte Carlo Integration 23
5.2.1 Monte Carlo integration using sample from f_t 23
5.2.2 Monte Carlo integration using $\mathcal{N}(0,1)$ 25
5.2.3 Monte Carlo integration using standard normal with drift 26
5.2.4 Possible issues 31
5.3 Examples 32
5.3.1 AA minimisations of some functions with a lot of local minima 32

6 Conclusion **38**

1 Introduction

Various optimization problems have great applications in everyday life. Therefore, it does not surprise that mathematicians continuously try to find new and efficient ways of function optimization. According to [10], a major issue in optimization is distinguishing between global and local optima. Naturally, with all factors being equal, a globally optimal solution is always more preferable than the local one. On the other hand, in practice it is not always possible to find a globally optimal solution and therefore one have to be satisfied with a locally optimal one. Although the local optimum is certainly better than no solution at all, it can often be far from the globally optimal solution and hence point us to the wrong conclusions. Since one can find maximum of function f by minimizing function $-f$, we will focus on problems of function minimization.

Let $g : S \subset \mathbb{R}^d \to \mathbb{R}$, $d \geq 1$ be cost function we want to minimize. We say that $x^* \in S$ is local minimum of function f if there exists open set $\mathcal{U} \ni x^*$ such that for every $x \in \mathcal{U}$, $g(x^*) \leq g(x)$. Moreover, $x^* \in S$ is global minimum of g if inequality $g(x^*) \leq g(x)$ holds for every $x \in S$. There is no general method for solving optimization problems for all cost functions. Moreover, according to [10], it is usually only possible to ensure that an algorithm will approach a local minimum with a finite amount of resources being put into the optimization process. However, we will discuss several methods of stochastic approximation that are under certain conditions able to find global minimum among multiple local minima.

Popularity of stochastic optimization methods has grown rapidly in the last two decades. Moreover, with a large number of methods it is now becoming the "industry standard" for solving various challenging optimization problems [10].

This work provides a summary of several methods for sampling from distributions and connects them with global optimization problem. We will present some stochastic optimization algorithms such as Simulated Annealing on functions defined on discrete sets, and its generalisation to function defined on more general sets. At the end, we will present the one-dimensional Adaptive Annealing method and we will test it on functions with a lot of local minima.

2 Sampling from a distribution

In this chapter several methods for sampling from a certain distribution will be presented. Even though there are ad-hoc methods that efficiently sample from some particular distributions, the focus will be put on distribution invariant sampling methods. Most softwares for mathematical and statistical computation have built-in methods for generating random numbers from typical distributions, but in the general case, there is no unified approach to the problem. In the one dimensional case, exact and asymptotical methods will be presented, while for generating random vectors, the situation is far more complicated as the components may be dependent.

2.1 Inverse sampling

As it has already been pointed out, many mathematical softwares have built-in methods for generating random numbers from some standard distributions, one of which is uniform distribution on $(0, 1)$ interval, denoted here as $\mathcal{U}(0, 1)$. The following result shows how to transform a random number from $\mathcal{U}(0, 1)$ into number from any continuous distribution.

Theorem 1. *Let U be a random variable from $\mathcal{U}(0, 1)$, and F cumulative distribution function (CDF) we want to sample from. If F is invertible on $(0, 1)$, then the random variable $X = F^{-1}(U)$ has CDF F.*

Proof. The distribution function of $X = F^{-1}(U)$ is

$$F_X(y) = P(F^{-1}(U) \leq y) = P(U \leq F(y)) = F_{\mathcal{U}}(F(y))$$

where $F_{\mathcal{U}}$ is CDF of $\mathcal{U}(0, 1)$. Furthermore,

$$F_{\mathcal{U}}(y) = \begin{cases} 0, & \text{for } y < 0, \\ y, & \text{for } y \in [0, 1), \\ 1, & \text{for } y \geq 1, \end{cases}$$

and since $F(y) \in (0, 1)$, we have

$$F_X(y) = F_{\mathcal{U}}(F(y)) = F(y), \forall\, y \in \mathbb{R}.$$

□

This result gives a "cookbook" about how to directly transform random number from $\mathcal{U}(0, 1)$ into a number from any distribution which has invertible CDF on $(0, 1)$. It should be noted that invertibility of CDF is not very restricting demand for continuous random variables.
Eventhough, at first sight this result gives a rather simple way of generating random numbers, in practice that is not really the case. First of all, target distribution is usually specified by its probability density function (PDF), rather than its CDF. Obtaining CDF explicitly from a given PDF may be impossible, just like in the case of normal distribution. Moreover, this method requires not only CDF which is rarely explicit, but its inverse. For all these reasons, this method could be computationally inefficient for many distributions. Therefore, other methods are preferred, one of which we will present next.

2.2 Markov Chain Monte Carlo

The idea of Markov Chain[1] Monte Carlo methods for sampling from a distribution F is to generate Markov chain whose limiting distribution is target distribution F. These methods are asymptotical and they approximate random number (vector) from target distribution F with state of the particular Markov chain after a certain number of steps. Markov Chain Monte Carlo (MCMC) methods differ in way in which they construct an underlying Markov chain. We present here a MCMC method called the Metropolis-Hastings[2] algorithm [7].

2.2.1 Definition and basic properties of Markov chains

For the simplicity of presentation, we will assume that the target distribution is discrete. The goal is to derive a Markov chain $X = (X_n, n \in \mathbb{N}_0)$ that converges to it. To do so, the chosen Markov chain must have a unique limiting distribution, which is equal to target one. Hence, before further discussion of Metropolis-Hastings algorithm, some basic results about discrete state, time-homogenous Markov chain will be presented.

Definition 1. Let S be a discrete set. A stochastic process $X = (X_n, n \in \mathbb{N}_0)$ on probability space $(\Omega, \mathcal{F}, P)$ with support on the set S is called a discrete time Markov chain if

$$P(X_t = i | X_{t_n} = i_n, ..., X_{t_1} = i_1) = P(X_t = i | X_{t_n} = i_n), \tag{1}$$

for each $t_1, ..., t_n \in \mathbb{N}_0$ such that $t_1 < \cdots < t_n$ and for each $i, i_1, \ldots, i_n \in S$ for which (1) is well defined.

The expression (1) is called the Markov property, and it states that the behaviour of the chain in the future, given present, does not depend on the past. The transition probability function of Markov chain X is defined with the expression

$$p(i, s; j, t) = P(X_t = j | X_s = i), \text{ for } i, j \in S, \ s < t. \tag{2}$$

For $t = s+1$, (2) is called the one-step transition probability function. Every Markov chain is uniquely defined by its one-step transition probability function and distribution λ_0 of initial state X_0.
If the one-step transition probability function does not depend on time t, then the Markov chain X is called time-homogenous Markov chain and the one-step transition probability function takes the form

$$p(i, n; j, n+1) = p(i, m; j, m+1) =: p_{ij}, \forall m, n \in \mathcal{N}_0, \ \ \forall i, j \in S.$$

Additionally, the matrix $\Pi = [p_{ij}]_{ij \in S}$ is called the transition matrix of Markov chain X. Moreover, Π is stochastic matrix, meaning that $\forall\ i, j \in S\ p_{ij} > 0$ and $\forall\, i \in\ S, \sum_{j \in S} p_{ij} = 1$.

Further on, asymptotical behaviour of Markov chain will be explored.

[1] Named after the Russian mathematician Andrey Markov (14 June 1856 N. S. Ryazan, Russian Empire - 20 July 1922 Petrograd, Russian SFSR)

[2] Named after Nicholas Constantine Metropolis (June 11, 1915 - October 17, 1999), a Greek - American physicist who was an author along with Arianna W. Rosenbluth, Marshall Rosenbluth, Augusta H. Teller, and Edward Teller of the 1953 paper Equation of State Calculations by Fast Computing Machines which first proposed the algorithm for the case of symmetrical proposal distributions, and Wilfred Keith Hastings (July 21, 1930 in Toronto, Ontario, Canada - May 13, 2016) a Canadian statistician who extended it to the more general case in 1970.

Definition 2. Let $X = (X_n, n \in \mathbb{N}_0)$ be Markov chain with set of states S and transition probability matrix $\Pi = [p_{ij}]_{i,j \in S}$. A probability distribution $\pi = (\pi_i, i \in S)$ on set S is stationary (invariant) distribution of Markov chain X if

$$\pi = \pi\Pi,$$

that is

$$\pi_j = \sum_{k \in S} \pi_k p_{kj}, \ \ \forall j \in S.$$

Interpretation of the notation $\pi = (\pi_i, i \in S)$ is the following. If X is random variable from distribution π, then for each $j \in S$, $P(X = j) = \pi_j$. It should be noted that the stationary distribution, if it exists, does not depend on the initial state distribution of Markov chain X.

Definition 3. Let $X = (X_n, n \in \mathbb{N}_0)$ be a Markov chain with set of states S and transition probability matrix $\Pi = [p_{ij}]_{i,j \in S}$. A probability distribution $\nu = (\nu_i, i \in S)$ on the set S is the limiting distribution of the Markov chain X, if for every $i, j \in S$

$$\lim_{n \to \infty} p_{ij}^{(n)} = \nu_j, \tag{3}$$

where $p_{ij}^{(n)}$ is the element in (i, j) position of matrix Π^n.

If the limit (3) exists, then it is unique, but it does not have to define a probability distribution.

Remark 1. Let λ_0 stand for distribution of the initial state X_0 of Markov chain X. It is easily shown that the distribution of state X_n of Markov chain X after n steps is $\lambda_n = \Pi^n \lambda_0$ [11]. More precisely, we have

$$P(X_n = j) = [\lambda_n]_j = \sum_{i \in S} [\lambda_0]_i p_{ij}^n.$$

Therefore, if X has limiting distribution $\nu = (\nu_i, i \in S)$, then the distribution of state X_n in the limit when $n \to \infty$ is

$$\lim_{n \to \infty} P(X_n = j) = \lim_{n \to \infty} [\lambda_n]_j = \sum_{i \in S} [\lambda_0]_i \lim_{n \to \infty} p_{ij}^n = \sum_{i \in S} [\lambda_0]_i \nu_j = \nu_j \sum_{i \in S} [\lambda_0]_i = \nu_j, \tag{4}$$

provided that the limit in (4) exists. In other words a Markov chain $(X_n, n \in \mathbb{N})$ converges in distribution to limiting distribution ν. Therefore, a simulation of Markov chain with limiting distribution as our target one, after certain numbers of steps generates number from distribution which is arbitrary close to the target one.

Moreover, there is a strong connection between stationary and limiting distribution.

Proposition 1. *Let $\nu = (\nu_i, i \in S)$ be limiting distribution of Markov chain with the set of states S. Then ν is the stationary distribution.*

The following results will provide an answer under which conditions the stationary and limiting distribution coincide.

Definition 4. For states $i, j \in S$ we say that j is reachable from i and write $i \to j$, if

$$P(T_j < \infty | X_0 = i) > 0,$$

where $T_j = \min\{n \geq 0 : X_n = j\}$. Additionally, we say that i, j communicate and write $i \leftrightarrow j$, if $i \to j$ and $j \to i$.

Furthermore, relation of communication is equivalence relation on S, and therefore induces partition of S to communication classes [11].

Definition 5. Markov chain X is called irreducible if S consists of only one communication class.

Remark 2. If $p_{ij} > 0$ for each $i, j \in S$, then the Markov chain X is irreducible.

Definition 6. For state $i \in S$ we denote $d(i)$ as the greatest common divisor of set $\{n \geq 1 : p_{ii}^{(n)} > 0\}$. We say that state i aperiodic if $d(i) = 1$. Otherwise, it is said to be periodic.

Remark 3. If $p_{ii} > 0$ for $i \in S$, then i is aperiodic state. Furthermore, it can be shown that periodicity is a feature of the communication class. Hence, irreducible Markov chain can either be periodic or aperiodic [11].

Theorem 2. *Let $(X_n, n \in \mathbb{N}_0)$ be irreducible, aperiodic Markov chain with the set of states S, transition probability matrix $\Pi = [p_{ij}]_{i,j \in S}$ and stationary distribution π. Then π is also the limiting distribution.*

Proofs of Proposition 1 and Theorem 2 can be found in [11].
In the class of irreducible aperiodic Markov chains, limiting and stationary distributions coincide. Hence, the results that guarantee existence and uniqueness of stationary distribution, do the same for the limiting one. The question remains to be answered, for which Markov chain does limiting distribution exist.

Definition 7. A Markov chain $(X_n, n \in \mathbb{N}_0)$ with the set of states S and transition probability matrix $\Pi = [p_{ij}]_{i,j \in S}$ is said to be reversible for probability distribution $\mu = (\mu_i, i \in S)$, if

$$\mu_i p_{ij} = \mu_j p_{ji}, \ \forall i, j \in S.$$

Remark 4. It should be noticed that since $\mu_i p_{ij} = \mu_j p_{ji}$, then

$$\sum_{i \in S} \mu_i p_{ij} = \sum_{i \in S} \mu_j p_{ji} = \mu_j \sum_{i \in S} p_{ji} = \mu_j.$$

In the other words, probability distribution μ from the Definition 7 is stationary distribution for Markov chain X. Therefore, if the generated Markov chain is irreducible, aperiodic and reversible for distribution μ, then X has limiting distribution equal to μ.

2.2.2 Metropolis-Hastings algorithm

In this section, the main goal is to obtain a form of the particular Markov chain $X = (X_n, n \in \mathbb{N}_0)$, that is its transition matrix $\Pi = [p_{ij}]_{ij}$, which has properties that guarantee existence of the limiting distribution equal to the target distribution $\pi = (\pi_i, i \in S)$. Further on, target distribution F will be denoted as π.

According to Remark 4, if Markov chain X is reversible for π then it has stationary distribution equal to π. In that manner it is desired for transition probability matrix $\Pi = [p_{ij}]_{ij}$ of Markov chain X to satisfy

$$\pi_i p_{ij} = \pi_j p_{ji}. \tag{5}$$

Guided with the rejection sampling algorithm [8], the approach would be to separate transition from state i to state $j \in S$ into two independent sub-steps: the proposal and the acceptance-rejection. Let $Q = [q_{ij}]_{ij}$ be symmetric, stochastic matrix such that for every $i, j \in S$ and $n \in \mathbb{N}_0$, q_{ij} stands for conditional probability of j being proposed as value for X_{n+1} given $X_n = i$. It should be noticed that for every $i \in S$, i-th row of matrix Q defines conditional distribution $q(\cdot|i)$ such that if X^{cand} is random variable from $q(\cdot|i)$, then $q_{ij} = P(X^{cand} = j|X_n = i)$. Furthermore, let for each $i, j \in S$, $A_{ij} = \min\{1, \frac{\pi_j}{\pi_i}\} \in [0, 1]$ be conditional probability of proposed j being accepted as value of X_{n+1} given $X_n = i$. More precisely, conditionally on X_n

$$X_{n+1} = \begin{cases} X^{cand} & \text{with probability } A_{ij} \\ X_n & \text{with probability } 1 - A_{ij}, \end{cases}$$

where X^{cand} is the proposed value from $q(\cdot|X_n)$ for X_{n+1}. Hence, for each $i, j \in S$ the following equality holds

$$p_{ij} = q_{ij} A_{ij}. \tag{6}$$

For $i, j \in S$, A_{ij} and q_{ij} are further on called acceptance and proposal probabilities of $X_{n+1} = j$ given $X_n = i$ respectively. Moreover,

$$\frac{A_{ij}}{A_{ji}} = \begin{cases} \frac{\pi_j}{\pi_i}, & \text{if } A_{ij} < 1 \\ \frac{1}{\frac{\pi_i}{\pi_j}}, & \text{if } A_{ij} = \end{cases} = \frac{\pi_j}{\pi_i}.$$

Moreover, since Q is symmetric matrix, then

$$\frac{A_{ij}}{A_{ji}} = \frac{\pi_j}{\pi_i} = \frac{\pi_j q_{ji}}{\pi_i q_{ij}},$$

which is due to (6) equivalent to $\pi_i p_{ij} = \pi_j p_{ji}$. We have just justified the use of this special form of the acceptance probability in a way that Markov chain obtained in this way has a stationary distribution equal to target distribution π. On the other hand, the irreducibility of X should be checked in each specific application [7]. Unlike the acceptance probability, no conditions on the form of the proposal probability had been made. It is a free parameter of the Metropolis-Hastings method which should be adjusted to each particular problem.

Algorithm

$k = 0$: Generate the initial state x_0 from an arbitrary distribution with support on S, choose the proposal probability matrix $Q = [q_{ij}]_{i,j \in S}$ and number of iterations $T \in \mathbb{N}$ of the algorithm

$k = 1$:

1. Generate x^{cand} from distribution $q(\cdot|x_0)$
2. Generate random number u from $\mathcal{U}(0, 1)$
3. Set

$$x_1 = \begin{cases} x^{cand} & \text{for } u \leq A_{x^{cand}x_0} \\ x_0 & \text{otherwise} \end{cases}$$

Obtain x_1 which is equal to proposal x^{cand} or remains same as x_0

$\vdots$

$k = T : x_{T-1}$ had already been generated

1. Generate x^{cand} from distribution $q(\cdot|x_{T-1})$
2. Generate random number u from $\mathcal{U}(0,1)$
3. Set

$$x_T = \begin{cases} x^{cand} & \text{for } u \leq A_{x^{cand}x_{T-1}} \\ x_{T-1} & \text{otherwise} \end{cases}$$

It remains to show that this algorithm really generates Markov chain with transition probabilities $p_{ij} = g_{ij}A_{ij}$. This follows from the fact that $P(U \leq \alpha) = \alpha$, where U is random variable from $\mathcal{U}(0,1)$ and $\alpha \in (0,1)$. More precisely,

$$p_{ij} = P(X_{t+1} = j|X_t = i) = P(X^{cand} = j|X_t = i)P(U \leq A_{ji}) = q_{ij}A_{ji}.$$

Remark 5. We have presented Metropolis Hastings algorithm for the discrete set S. When target distribution π is continuous, W. K. Hastings in [7] proposes the use of proper discrete approximation to π and then applying the same procedure as in the discrete case. Generally, the algorithm can be generalised to continuous distributions as well.

3 Applying sampling from distribution to global minimization problem

So far several methods for generating samples from certain distribution have been presented and it remains to connect them with the global optimization problem.

Let $g : S \subset \mathbb{R}^d \to \mathbb{R}$ be a real valued function we want to minimise, with the set of global minima S^*. Generating a random vector from a distribution with support on S^*, would generate a global minima of g with probability 1. The obvious problem in this concept is the lack of insight in set S^*. Therefore, the idea is to consider the distributions with the property: the random number generated from these distributions is close to the global minimum of g with a high probability. That could be obtained by deriving a sequence of distributions $(\pi_t)_t$, that converges to a distribution with support on S^*. Index t in notation π_t denotes time index, unlike the previous notation $\pi = (\pi_i, i \in S)$, where index i stood for state $i \in S$. Further on, it will be clear from context weather index denotes time index or state.

Since every PDF π_t should take large values on 'small' intervals that contain a global minima of g and small values otherwise, it is reasonable to assume that each π_t is proportional to $-\gamma(t)g$, where $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is a nondecreasing function of t such that $\gamma(t) \to \infty$ for $t \to \infty$. The purpose of the function γ is to provide easier separation of global from the local minimum. Furthermore, as PDF should be nonnegative, we will consider $\pi_t \propto e^{-\gamma(t)g}$, where $\propto$ means proportional to. Hence, for function γ, we define

$$\pi_t(a; \gamma(t)) = \frac{e^{-\gamma(t)g(a)}}{c_t}, \tag{7}$$

where c_t is a normalizing constant such that the Riemann-Stieltjes integral $\int_{a\in\mathbb{R}} \mathrm{d}F_t(a) = 1$, where F_t is CDF of π_t. The distribution with PDF given by (7) is known as Gibbs distribution[3] with parameter $\gamma(t)$. It can be shown that the sequence (π_t) converges in distribution to probability distribution concentrated on S^* [4].

We provide an informal proof of that statement in the case where g is defined on a discrete set S, and has a finite number of global minima. In this case c_t is of the form $c_t = \sum_{j\in S} e^{-\gamma(t)g(j)}$, hence

$$\pi_t(i) = \frac{1}{\sum_{j\in S} e^{\gamma(t)(g(i)-g(j))}}.$$

For $i \in S$ we can write

$$\pi_t(i) = \frac{1}{\sum_{j\in S_1} e^{\gamma(t)(g(i)-g(j))} + \sum_{j\in S_2} e^{\gamma(t)(g(i)-g(j))} + \sum_{j\in S_3} e^{\gamma(t)(g(i)-g(j))}},$$

where $S_i^1 = \{j \in S : g(i) < g(j)\}$, $S_i^2 = \{j \in S : g(i) > g(j)\}$, $S_i^3 = \{j \in S : g(i) = g(j)\}$. We need to prove that the distribution concentrated on S^* is a point-wise limit of (π_t) as $t \to \infty$

1. case: $i \notin S^*$

$S_i^1 \neq \emptyset$, and $\gamma(t)(g(i) - g(j)) > 0 \ \forall j \in S_i^1$. Hence, in the limit for $t \to \infty$, that is $\gamma(t) \to \infty$, $\forall j \in S_i^1$, $\gamma(t)(g(i) - g(j)) \to \infty$ and consequently $\sum_{j\in S_1} e^{\gamma(t)(g(i)-g(j))} \to \infty$. Since both of the

[3]Named after Josiah Willard Gibbs (February 11, 1839 - April 28, 1903), an American scientist who made important theoretical contributions to physics, chemistry, and mathematics

other two sums in the denominator of π_t are positive,

$$\lim_{t\to\infty} \pi_t(i) = 0, \text{ for } i \notin S^*.$$

2. case: $i \in S^*$
$S_i^1 = \emptyset$ and $\forall j \in S_i^2 \gamma(t)(g(i) - g(j)) < 0$. Hence, in the limit for $t \to \infty$, that is $\gamma(t) \to \infty$, $\gamma(t)(g(i)-g(j)) \to -\infty$ for $j \in S_i^1$ and consequently $e^{\gamma(t)(g(i)-g(j))} \to 0$ so $\sum_{j\in S_1} e^{\gamma(t)(g(i)-g(j))} \to 0$. Moreover, $e^{\gamma(t)(g(i)-g(j))} = 1$ for $j \in S_i^3 = S^*$, hence $\sum_{j\in S_3} e^{\gamma(t)(g(i)-g(j))} = |S^*|$.
We conclude that

$$\lim_{t\to\infty} \pi_t(i) = \frac{1}{n}, \text{ for } i \in S^*, \text{ where } n := |S^*|.$$

If we now define $\pi : S \to \mathbb{R}_+$ as a point-wise limit of π_t when $t \to \infty$, then

$$\pi(i) = \begin{cases} \frac{1}{n}, & \text{if } i \in S^* \\ 0, & \text{otherwise,} \end{cases}$$

such that $\sum_{i\in S} \pi(i) = 1$, then π is density of uniform distribution on set S^*.
A similar result can be obtained for functions defined on continuous sets, but proofs are technically more demanding [4].

Remark 6. Let X_t be a random variable (vector) from π_t and X a random variable (vector) from π. Then sequence (X_t) converges to X in distribution. More precisely, let for every $t \in \mathbb{N}_0$, F_t be CDF of X_t and F be CDF of X. Then for every $i \in S$, $F_t(i) = \sum_{j\le i} \pi_t(j)$ and $F(i) = \sum_{j\le i} \pi(j)$. Moreover, for every $i \in S$

$$\lim_{t\to\infty} F_t(i) = \lim_{t\to\infty} \sum_{j\le i} \pi_t(j) = \sum_{j\le i} \lim_{t\to\infty} \pi_t(j) = \sum_{j\le i} \pi(j) = F(i).$$

Therefore, sequence (X_t) converges to π in distribution. Moreover, if g has a unique global minimum, then X is almost surely constant, so convergence in distribution implies convergence in probability.

A stochastic global optimization algorithm that combines this idea with the Metropolis-Hastings method will be presented in the next chapter.

4 Simulated annealing

Simulated annealing (SA) is a stochastic optimization method proposed by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983 and by Vlado Černy in 1985. It is mostly and quite successfully used for solving image processing and combinatorial problems. Later on, the algorithm was modified to search global minimum of functions with domains in $\mathbb{R}^d$. The method itself was inspired by a 7000 year old process of annealing of the metal [2].

The annealing process is done in three stages: heating up metal to a temperature high enough, keeping metal heated long enough and slowly cooling it down, usually to room temperature. Slow cooling induces changes in the crystal cell of metal in a way that it reaches its minimum energy configuration. The result is a firmer metal with better qualities. The tricky part is that cooling has to be done slow enough so that changes in the crystal structure of the metal can be properly done.

The SA algorithm randomly explores an area around the current approximation of minimum and accepts the proposed neighbour as the next approximation if it has a lower function value compared to the previous one. The difference with deterministic local optimisation methods is that the SA algorithm also accepts a worse solution with some positive probability $p > 0$. The latter feature allows the algorithm to escape a local minimum in order to find the global one. At the initial stages of the algorithm, when the temperature is higher, it is more likely that the algorithm would accept a worse solution. At latter stages, as the temperature lowers, the algorithm would not as likely accept a worse solution. In other words, it is easier for SA to escape local minimum in the early stages of the algorithm.

4.1 Simulated annealing on a finite set S

We are minimizing function $g : S \to \mathbb{R}$, where $S \subset \mathbb{R}^d$ is a finite set. Even though S is finite, its cardinality is usually very large, especially in combinatorial problems and that is what makes them hard to optimize. For example, a traveling salesman is a NP-hard problem. Theoretically, the problem of finding a global minimum could be solved with the use of the Metropolis-Hastings algorithm by taking target distribution to be a Gibbs distribution with a parameter $\gamma(t)$ large enough. This should generate a number very close to a global minimum of g. The main problem of the approach is that for $\gamma(t)$ very large, the time needed for Markov chain to reach equilibrium can be exponential to $\gamma(t)$. The reason why this happens is that generating homogenous Markov chain from Gibbs distribution with large $\gamma(t)$ will very likely result in "sticking" to local minimum due to the probability of accepting a worse solution, which allows avoiding these situations, being very small. The SA algorithm tries to overcome this by generating discrete time-inhomogeneous Markov chain that converges in distribution to the distribution concentrated on the set of global minima of g.

Remark 7. In this section, for consistency with the literature, the nonincreasing function $T = T(t)$ called the cooling schedule will be used instead of the increasing function γ. For simplicity, $T(t) = 1/\gamma(t)$.

Before further discussion let us formalize basic elements used in the SA algorithm.

1. Finite set $S \subset \mathbb{R}^d$
2. A function $g : S \to \mathbb{R}$, called the cost function, with set of global minima S^*
3. For each $i \in S$ set of neighbours of i, $S(i) := S \setminus \{i\}$

4. Nonincreassing function $T : \mathbb{N} \to (0, \infty)$ called cooling schedule, whose function value $T(t)$ is called the temperature at time t

5. Symmetric stochastic matrix $[q_{ij}]_{ij \in S}$

The SA algorithm builds discrete time, time-inhomogenous Markov chain $X = (X(t))_t$ that, under certain assumptions converges in distribution to the distribution concentrated on the set S^*. One-step transition probability function of X is given with

$$p(i, j; t) = P(X(t+1) = j | X(t) = i) = q_{ij} \min\{1, e^{\frac{g(i)-g(j)}{T(t)}}\} > 0, \tag{8}$$

where q_{ij} is probability of j being proposed for value of $X(t+1)$ conditionally on $X(t) = i$.

4.1.1 The convergence analysis

Before the introduction of the main convergence result of SA let us give a few definitions.

Definition 8. We say that SA algorithm converges, if

$$\lim_{t \to \infty} P(X(t) \in S^*) = 1.$$

Definition 9. We say that state i communicates with S^* at height h if there exists a path in S that starts at i and ends up at some element of S^* such that the largest value of g along that path is $g(i) + h$.

Remark 8. Definition 9 can be interpreted in a way that there exists trajectory of Markov chain generated with SA that starts at $i \in S$, and ends up in S^*, which at some point requires 'climbing uphill' to the height for h higher than the one trajectory has started from.

Theorem 3. *Let d^* be the smallest number such that every $i \in S$ communicates with S^* at height d^*. Then, the SA algorithm converges iff*

$$\begin{aligned} &i)\ \lim_{t \to \infty} T(t) = 0, \ \textit{and} \\ &ii)\ \sum_{t=1}^{\infty} e^{\frac{-d^*}{T(t)}} = \infty. \end{aligned}$$

Proof of Theorem 3 can be found in [2].
The constant d^* is a sort of measure of difficulty for SA to escape the local minima of g. Specifically, d^* is the height at which the lowest local minimum, which is not a global one, communicates with S^*. Let the current state $X(t)$ be the local minimum $i \in S$. Then there exists a path in S such that $g(j) - g(i) \leq d^*$ for each j along the path. The probability of successfully escaping this local minimum is

$$\begin{aligned} P(X(t+1) \neq i | X(t) = i) &= P(X(t+1) \in S(i) | X(t) = i) \\ &= \sum_{j \in S(i)} P(X(t+1) = j | X(t) = i) \\ &\leq |S(i)| e^{\frac{-d^*}{T(t)}} \propto e^{\frac{-d^*}{T(t)}}, \end{aligned}$$

where we have used (8) and the definition of constant d^* from Definition 9.
Assume now that an infinite number of trials had been made in order to escape i. Theorem 3

states that the SA converges iff infinite number of those trials had been successful. According to the condition (ii) of Theorem 3, $T(t)$ should not decrease faster than of order $1/\log(t)$. According to [2], one of the most popular (at least theoretically) cooling schedules are of form

$$T(t) = d/\log(t). \tag{9}$$

If cooling schedule $T = T(t)$ decreases as slowly as (9), then it can be fairly approximated by piecewise constant cooling schedule $T' = T'(t)$ defined with

$$T'(t) = 1/k \tag{10}$$

for $t \in [t_k, t_{k+1})$, where $t_1 = 1, t_{k+1} = t_k + e^{kd}$, for $d \geq d^*$. Moreover, according to [2], the statistics of the Markov chain X under slowly decreasing cooling scedule stays pretty much unchanged if a piece-wise cooling schedule (10) is used instead. Therefore, in order to better understand rationale behind the SA algorithm, let us observe Markov chain X corresponding to the cooling schedule (10). It is obvious that (10) satisfies conditions from the Theorem 3. Moreover, T' is constant on intervals $t_k \leq t < t_{k+1}$, so Markov chain X is homogenous along the same intervals. In that manner let $X^k = (X_t^k)$ be Markov chain $X(t)$ for $t \in [t_k, t_k + e^{kd})$, that is with the transition matrix $\Pi_k = [q_{ij} \min\{1, e^{k(g(i)-g(j))}\}]_{ij}$.

Remark 9. Since for each $k \in \mathbb{N}$, $[\Pi_k]_{ij} = q_{ij} \min\{1, e^{k(g(i)-g(j))}\} > 0$, then according to Remarks 2 and 3 Markov chain X^k is irreducibile and apperiodic.

Consistently with conclusions in Section 2 and Remark 2 for every $k \in \mathbb{N}$ limiting distribution of X^k is Gibbs distribution with parameter k. Length of $[t_k, t_{k+1})$ is e^{kd}. Therefore, for smaller k, the interval corresponding to it is shorter, but at the same time Metropolis-Hastings would need less iterations to reach equillibrium. On the other hand, for a larger k, length of the corresponding interval is exponential to k, but so is the time for Markov chain to reach equillibrium. According to that, the SA works in a way that simulates Markov chain X^k whose limiting distribution is Gibbs distribution with parameter k. Then it uses a random number generated from (close to) this limiting distribution as the initial state of Markov chain X^{k+1}. In this way, at the early stages, the algorithm escapes local minimum quickly. In later stages, the algorithm acts more like a gradient method, focusing on drifting previously obtained solutions towards global minimum.
In this manner one can seek an interpretation of constant $d > d^*$. Since the length of the interval $[t_k, t_{k+1})$ is e^{kd}, the more 'difficult' function g is to optimise, that is the harder it is to escape its local minima, more time will be needed in order to do so and consequentially larger d is required.

Algortihm
Generate the approximation x^* of global minimum of g from an arbitrary distribution with support on S. Define cooling schedule $T = T(t)$, initial and terminal temperature T_0 and T^*

While $T_0 \leq T \leq T^*$

1. Generate x^{cand} from distribution $q(\cdot|x^*)$
2. Generate random number u from $\mathcal{U}(0, 1)$
3. Update current approximation x^* of global minimum of g in way

$$x^* = \begin{cases} x^{cand} & \text{for } u \leq e^{\frac{x_0 - x^{cand}}{T}} \\ x^* & \text{otherwise} \end{cases}$$

4. Update temperature T

Generalization of the SA algorithm to infinite discrete S can be done in a similar way. Firstly, because the underlying theory of Markov chain behind the SA is done generally for discrete sets S. Nevertheless, according to [6], rigorous proofs of convergence of the SA in its original formulation have been given only in the case of the finite state space S. Even though generalisation of the SA for continuous sets S is also possible, we will move to slightly different approach using diffusions. More precisely, we will move from discrete time optimization on discrete set S, to continuous time optimization on continuous set S.

4.2 Generalization of the SA for continuous set S

Motivated by image processing[4] problems with continuous grey-levels, Ulf Grenander and Steve Geman proposed use of a special stochastic differential equation known as Langevin equation[5] as another global minimization algorithm [5].
One can find a local minimum of $g : \mathbb{R}^d \to \mathbb{R}$ by starting at arbitrary $x_0 \in \mathbb{R}^d$ and then applying Gradient descent algorithm

$$x_{t+1} = x_t - h\nabla g(x_t), \tag{11}$$

for x_{t+1}, where $h > 0$ small enough [9]. If we generalise (11) to continuous time, that is, if we consider it in limit as $h \to 0$, we obtain

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = -\nabla g(x_t).$$

The idea was to "upgrade" gradient descent method in continuous time by adding some randomness into it. The result is a method that has a capability of escaping the local minima by accepting worse solutions with positive probability. Furthermore, guided with results from the SA, these random fluctuations in the early stages of the algorithm should have more influence on the behaviour of the algorithm, while in latter ones its influence should drop.

Langevin equation The stochastic differential equation

$$\mathrm{d}X(t) = \sqrt{2T}\mathrm{d}B(t) - \nabla g(X(t))\mathrm{d}t \tag{12}$$

is called Lengevin equation at temperature T, where g is function with domain in $\mathbb{R}^d$, $B(t)$ is the standard d-dimensional Brownian motion and T "temperature" that controls amplitude of random fluctuations.
Equation (12) defines d-dimensional diffusion X^T whose limiting distribution (with some conditions on g) is Gibbs distribution with the parameter $1/T$ [4]. Markov processes in continuous time with the continuous set of states and almost surely continuous trajectories are called diffusions. As already stated, when $T \to 0$, Gibbs distribution with parameter $1/T$ concentrates on the set of the global minima of g. Hence, in low temperature equilibrium it can be expected to find X^T near the global minimum. Unfortunately, the time

[4]Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Details can be found at [13]

[5]Named after Paul Langevin (23 January 1872 - 19 December 1946), a prominent French physicist who developed Langevin dynamics and the Langevin equation

required to approach equilibrium increases exponentially with $1/T$, just as in the discrete case. This suggests that (12) should be considered with a gradually decreasing temperature, $T = T(t) \to 0$ [4]. Therefore, given nonincreasing function $T : \mathbb{R}_+ \to \mathbb{R}_+$ we define diffusion X with equation

$$\mathrm{d}X(t) = \sqrt{2T(t)}\mathrm{d}B(t) - \nabla g(X(t))\mathrm{d}t. \tag{13}$$

The hope is that in early stages of the algorithm, when $T(t)$ is large, random fluctuations caused by $\sqrt{2T(t)}\mathrm{d}B(t)$ in (12) will be influential enough to move X from local minimum. At later stages, small values of $T(t)$ will cause the behaviour of algorithm to be essentially a gradient descent.

Convergence analysis

Definition 10. Let (S, d) is a metric space and $\mathcal{S} = \mathcal{B}(d)$ the Borel σ-algebra there. Let (μ_n) be a sequence of probability measures on $(S, \mathcal{S})$. We say that (μ_n) converges weakly to a probability measure μ on $(S, \mathcal{S})$ if for every continuous, bounded function $f : S \to \mathbb{R}$, $\int f\mathrm{d}\mu_n \to \int f\mathrm{d}\mu$, when $n \to \infty$.

Remark 10. In case where $S = \mathbb{R}$, (μ_n) converges weakly to a probability measure μ iff $F_n(x) \to F(x)$ when $n \to \infty$, for all x such that F is continuous at x. F_n and F are CDFs of μ_n and μ respectively [12].

Let

$$p(t, x) = p(t_0, x_0; t, x),$$

be the transition density function for diffusion X defined by (12) at temperature T, where t_0, x_0 are initial time and state respectively. Furthermore, let $\pi(t, x)$ be its stationary distribution, Gibbs distribution with the parameter $1/T(t)$.

Theorem 4. *Let $g : \mathbb{R}^d \to \mathbb{R}$, $d \geq 1$ be C^∞ function such that*

$$\lim_{|x|\to\infty} f(x) = \infty,$$

$$\lim_{|x|\to\infty} |\nabla f(x)| = \infty.$$

Furthermore, assume that $\Delta f/(1+|\nabla f|^2)$ is bounded below. Then there exists constant $\delta > 0$ such that if

$$\lim_{t\to\infty} T(t) = 0, \ \textit{and}$$

$$\int_{t_0}^{\infty} e^{-\frac{\delta}{T(t)}}\mathrm{d}t = \infty,$$

then

$$|p(t_0, x_0 : t, \cdot) - \pi(t, \cdot)| \to 0, \quad \textit{as } t \to \infty.$$

Smoothness condition on function g can be weakened. Proof of Theorem 4 is based on the parabolic partial differential equation and can be found in [5].

The constant δ from Theorem 4 is sort of equivalent to the constant d^* from Theorem 3. Unlike the discrete case, there is no insight in the form of the constant δ. Therefore, optimal value of δ will be given in the case where the set of minima of g consists of isolated points. Let $a_l, a_2, ..., a_N$ be minima of g. For simplicity, we assume that g has the unique global

minimum a_N. Furthermore, let $a_l, ..., a_N$ be nondegenerate, that is the Hessian of g at $a_l, ..., a_N$ is positive definite. Consider now the class of curves $\beta_i(s)$ for $s \in [0,1]$ such that for each $i = 1, ..., N$ $\beta_i(0) = a_i$, $\beta_i(1) = a_N$, and define

$$\delta_i = \min_{\beta_i} \max_{s \in [0,1]} (g(\beta(s)) - g(a_i)).$$

δ_i is maximal height that the algorithm needs to climb uphill in order to reach a_N from a_i in the "easiest" way. It is now clear that δ_i corresponds to the height from Definition 9, at which the local minimum a_i communicates with the global minimum a_N.
Finally, we define

$$\delta = \max_{i=1,...,N-1} \delta_i,$$

as the optimal (smallest) value of constant δ that fulfils conditions of Theorem 4 in the case where the set of minima of g consists of isolated points.
Furthermore, condition $\int_{t_0}^{\infty} e^{-\frac{\delta}{T(t)}} \mathrm{d}t = \infty$ of Theorem 4 implies that cooling schedule $T = T(t)$ should not decrease faster then of order $1/\log(t)$. In that manner, analogous conclusions about approximation of such slow decreasing schedule with piece-wise constant one as in discrete case hold here as well.

5 Adaptive Annealing

So far we were dealing with the problem of global minimization of real valued function g with domain in $\mathbb{R}^d$. Initially, idea was to sample from Gibbs distribution with parameter large enough to ensure that the generated number is close to global minimum of g with high probability. The Metropolis-Hastings method turned out to be computationally inefficient due to long time needed for appropriate Markov chain to reach its limiting distribution. Simulated annealing tries to compensate this by generating time-inhomogeneous Markov chain which under certain conditions converges to distribution with support on the set of global minima of g. Theorems that guarantee convergence of such Markov chain require, among other, that cooling schedule is slow enough so that $\sum_{t=1}^{\infty} e^{\frac{-d^*}{T(t)}} = \infty$, or in the continuous case $\int_{t_0}^{\infty} e^{-\frac{\delta}{T(t)}} \mathrm{d}t = \infty$, for initial time t_0 and constants $d^*, \delta > 0$. In that manner, $T(t)$ should not decrease faster then of order $1/\log(t)$. Therefore, if cooling schedule $T = T(t)$ decreases slowly as in (9), then it can be fairly approximated by piecewise constant cooling schedule (10). Moreover, according to [2], statistics of the Markov chain X under slowly decreasing cooling schedule stays pretty much unchanged if piece-wise cooling schedule (10) is used instead. According to that, we have showed that the SA works in a way that simulates time-homogenous Markov chain $X^k = (X_t^k)_t$ whose limiting distribution is Gibbs distribution with the parameter k. Then the algorithm uses a random number generated from (close to) that limiting distribution as the initial state of Markov chain X^{k+1}. Since length of each interval $[t_k, t_{k+1})$ corresponding to X_k is e^{kd}, that for each k, time exponential to k is required in order to move from Gibbs distribution with parameter k to Gibbs distribution with parameter $k+1$.
The idea of the Adaptive Annealing (AA) method is to construct stohastic process (a_t), which is no longer necessary Markov process, such that for every t, a_t is from distribution "similar" to Gibbs distribution with the parameter $\gamma(t)$. Therefore, time needed for transition from Gibbs distribution with parameter k to Gibbs distribution with parameter $k+1$ is proportional to the inverse of the parameter function γ, i.e. for γ linear required time would also be linear.

Let $g \in C^2(\mathbb{R}^d)$ be the cost function we want to minimize. As already stated, the approach is to generate a stochastic process (a_t) such that for every $t > 0$, a_t is random vector from distribution with PDF

$$f(a;t) = f_t(a) = \frac{e^{-\gamma(t)g(a)}}{c_t} f(a;0), \tag{14}$$

where $f_0(a) = f(a;0)$ is PDF of multidimensional standard normal distribution ($\mathcal{N}(0, I)$), $\gamma \in \mathcal{C}^1(\mathbb{R}_+)$ nondecreasing function of time t, such that $\gamma(t) \to \infty$ when $t \to \infty$ and $c_t > 0$ normalizing constant defined as $c_t = \int_{\mathbb{R}^d} e^{-\gamma(t)g(a)} f_0(a) \mathrm{d}a$. Let a_t be a random vector describing the current approximation of a global minimum of g. The random vector describing the next approximation a_{t+h} is determined in the following way. The deterministic function G_t is determined so that

$$a_{t+h} = a_t - hG_t(a_t) := H_t(a_t),$$

is random vector from f_{t+h}, where a_t is random vector from f_t.

5.1 The one-dimensional AA minimisation

Let $g \in C^2(\mathbb{R})$ be the function we want to minimize. Moreover, let for every t

$$\begin{aligned}
&a_t \text{ random variable from } f_t\\
&G_t \in C^1(\mathbb{R}) \text{ function whose form will be specified below}\\
&H_t : \mathbb{R} \to \mathbb{R}, \text{ function defined as } H_t(a) = a + hG_t(a)\\
&h > 0.
\end{aligned}$$

Since G_t is smooth function, for every $x, y \in \mathbb{R}$ $x \approx y$ we have $G_t(x) \approx G_t(y)$. In that manner, for h small enough, $G_t(a_t) \approx G_t(a_{t+h})$ so, $a_{t+h} = a_t - hG_t(a_{t+h})$, that is

$$a_t = a_{t+h} + hG_t(a_{t+h}) = H_t(a_{t+h}).$$

It is now evident that G_t, and hence H_t have to satisfy the following. If a_t is a random variable from PDF f_t describing the current approximation of global minimum of g, then the random variable $a_{t+h} = H_t^{-1}(a_t)$ describing the next approximation has to be from f_{t+h}.

Theorem 5. *Let $X = (X_1, ..., X_n)$ be continuous random vector with PDF f_X and let $g : \mathbb{R}^n \to \mathbb{R}^n$ be a Borel function such that*

$$\begin{aligned}
&g : L \to T, \text{ is bijection where } T \subseteq \mathbb{R}^n,\ L = \{(x_1, ..., x_n) \in \mathbb{R}^n : f_X(x_1, ..., x_n) > 0\}\\
&g^{-1} \text{ is smooth on } T \text{ and } Dg^{-1}(y_1, ..., y_n) \neq 0 \text{ for all } (y_1, ..., y_n) \in T, \text{ where}
\end{aligned}$$

$$Dg^{-1}(y_1, ..., y_n) = det\big[\frac{\partial g_i^{-1}(y_1, ..., y_n)}{\partial y_i}\big]_{ij=1...n}.$$

Then $Y = g(X) = (Y_1, ..., Y_n)$ is continuous random vector with PDF

$$f_Y(y_1, ..., y_n) = f_X(g^{-1}(y_1, ..., y_n))|Dg^{-1}(y_1, ..., y_n)|I_T(y_1, ..., y_n).$$

Proof of Theorem 5 can be found in [11]. According to Theorem 5 f_{t+h} as PDF of a random variable $a_{t+h} = H_t^{-1}(a_t)$, is given by

$$f_{t+h}(a) = f(a; t+h) = f(H_t(a))|H_t'(a)|.$$

Since h is arbitrary small, we can assume $1 + hG'(a) \geq 0$, so that the absolute value can be omitted. Therefore,

$$f_{t+h}(a) = f(a + hG_t(a)(1 + hG_t'(a)). \tag{15}$$

By applying Taylor expansion to expression $f_t(a + hG_t(a))$ in (15) around a we obtain

$$\begin{aligned}
f(a; t+h) &= \big(f(a; t) + hG_t(a)\frac{\partial}{\partial a}f(a; t) + O(h^2)\big)\big(1 + hG_t'(a)\big)\\
&= f(a; t) + h\big(G_t(a)\frac{\partial}{\partial a}f(a; t) + G_t'(a)f(a; t)\big) + O(h^2)\\
&= f(a; t) + h\frac{\partial}{\partial a}\big(G_t(a)f(a; t)\big) + O(h^2),
\end{aligned}$$

that is,

$$\frac{f(a; t+h) - f(a; t)}{h} = \frac{\partial}{\partial a}\big(G_t(a)f(a; t)\big) + \frac{O(h^2)}{h}. \tag{16}$$

If we consider (16) in limit when $h \to 0$ we obtain differential equation

$$\frac{\partial f}{\partial t}(a;t) = \frac{\partial}{\partial a}\big(G_t(a) f(a;t)\big). \tag{17}$$

To ensure uniqueness of solution G_t of ordinary differential equation (ODE) (17) we add an initial condition $\lim\limits_{a\to\infty} G_t(a) = 0$ to it. That is a reasonable initial condition since we deal with unconstrained optimization. Finally, function G_t that solves boundary value problem

$$\begin{cases} \frac{\partial f}{\partial t}(a;t) = \frac{\partial}{\partial a}\big(G_t(a) f(a;t)\big) \\ \lim_{a\to\infty} G_t(a) = 0 \end{cases}$$

is given by

$$G_t(a) = \frac{1}{f(a;t)} \int_{-\infty}^{a} \frac{\partial f}{\partial t}(s;t)\mathrm{d}s. \tag{18}$$

Therefore, if a_t is random variable from f_t, transformation

$$a_{t+h} = a_t - h\frac{1}{f(a_t;t)} \int_{-\infty}^{a_t} \frac{\partial f}{\partial t}(s;t)\mathrm{d}s \tag{19}$$

is random variable from f_{t+h}.
It should be noted that for most function g, $\partial_t f_t$ can not be exactly determined. Therefore, in order to use it in practice, it should be approximated. One way of expressing $\partial_t f_t$ is

$$\begin{aligned} \partial_t f(a;t) &= -\frac{c'(t)}{c(t)}\frac{1}{c(t)} e^{-\gamma(t)g(a)} f(a;0) - g(a)\gamma'(t)\frac{1}{c(t)} e^{-\gamma(t)g(a)} f(a;0) \\ &= f(a;t)\big(-\frac{c'(t)}{c(t)} - g(a)\gamma'(t)\big). \end{aligned}$$

Since $c(t) = \int_{\mathbb{R}} e^{-\gamma(t)g(a)} f(a;0)\mathrm{d}a$, then

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = \int_{\mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}t} e^{-\gamma(t)g(a)} f(a;0)\mathrm{d}a = -\int_{\mathbb{R}} g(a)\gamma'(t) e^{-\gamma(t)g(a)} f(a;0)\mathrm{d}a,$$

and therefore

$$-\frac{c'(t)}{c(t)} = \int_{\mathbb{R}} \frac{g(a)\gamma'(t)}{c(t)} e^{-\gamma(t)g(a)} f(0;a)\mathrm{d}a = \gamma'(t)\int_{\mathbb{R}} g(a) f(t;a)\mathrm{d}a = \gamma'(t)E_t[g(a)] = \gamma'(t)\mu_t,$$

where $\mu_t := E_t\big[g(a)\big]$ is expectation of transformation $g(a)$ of random variable a from f_t. Hence, $\partial_t f_t$ can be expressed with

$$\partial_t f(t;a) = \gamma'(t) f(t;a)\big(\mu_t - g(a)\big). \tag{20}$$

By applying (20) in (18) it follows that

$$G_t(a_t) = \frac{\gamma'(t)}{f(a_t;t)} \int_{-\infty}^{a_t} \big(\mu_t - g(a)\big) f(a;t)\mathrm{d}a := \frac{\gamma'(t)}{f(a_t;t)} I_1(a_t). \tag{21}$$

The other way to express $\partial_t f(a;t)$ is to use approximation

$$\frac{\partial f}{\partial t}(a;t) \approx \frac{f(a;t) - f(a;t-h)}{h}, \tag{22}$$

and therefore obtain approximation for G_t

$$G_t(a_t) \approx \frac{1}{hf(a_t;t)} \int_{-\infty}^{a_t} \big(f(a;t) - f(a;t-h)\big)\mathrm{d}a =: \frac{1}{hf(a_t;t)} I_2(a_t). \tag{23}$$

In continuation, we will discuss both (21) and (23) separately.

Remark 11. It is important to note that the expressions (21) and (23) for most function g can not be explicitly defined. Therefore, numerically the problem of generating the next approximation a_{t+h} of a global minima of a function g is reduced to finding an efficient way to calculate the value of G_t in current approximation a_t. We will deal with this problem below.

5.2 Monte Carlo Integration

As discussed above, the initial problem is reduced to problem of efficient calculation (approximation) of one of following integrals

$$I_1(a_t) = \int_{-\infty}^{a_t} \big(\mu_t - g(s)\big) f(s;t)\mathrm{d}s \tag{24}$$

$$I_2(a_t) = \int_{-\infty}^{a_t} \big(f(a;t) - f(a;t-h)\big)\mathrm{d}s. \tag{25}$$

The approach is to express integrals in (24) and (25) as the expectation of a proper transformation of a random variable from a distribution that is hopefully easy to sample from. After integrals (24) and (25) are expressed as such expectations, then they can be approximated by appropriate sample means. It should be noted that (24) and (25) are transformations of random variable a_t and therefore are random variables themselves. We will now consider three different approaches to problem of approximating (24) and (25).

5.2.1 Monte Carlo integration using sample from f_t

First, let us discuss approximation of integral in (24). As commented in the introductory section of Monte Carlo integration, first

$$I_1(a_t) = \int_{-\infty}^{\infty} (\mu_t - g(s))\, I_{(-\infty,a_t]}(s) f(s;t)\mathrm{d}s = E_{f_t}\left[(\mu_t - g(a)) I_{(-\infty,a_t]}(a)\right],$$

where $E_{f_t}\left[(\mu_t - g(a)) I_{(-\infty,a_t]}(a)\right]$ is expectation with respect to random variable a from f_t. Therefore, if $a_t^1, ..., a_t^n$ is random sample from f_t, then integral in (24) may be approximated with

$$E_{f_t}[(\mu_t - g(a)) I_{(-\infty,a_t]}(a)] \approx \frac{1}{n}\sum_{i=1}^{n} (\mu_t - g(a_t^i)) I_{(-\infty,a_t]}(a_t^i) = \frac{1}{n} \sum_{\{i:a_t^i \le a_t\}} (\mu_t - g(a_t^i)).$$

Furthermore, when (26) is applied to (21) it follows that

$$G_t(a_t) \approx \frac{\gamma'(t)}{n f_t(a_t)} \sum_{\{i:a_t^i \le a_t\}} (\mu_t - g(a_t^i)),$$

where $a_t^1, ..., a_t^n$ is random sample from f_t.

In order to approximate (25) we will deal with at first sight different approach, which is essentially equal to previous one. First, it should be noticed that

$$I_2(a_t) = \int_{-\infty}^{a_t} \big(f(a;t) - f(a;t-h)\big)\mathrm{d}a = F_t(a_t) - F_{t-h}(a_t),$$

where for every $t > 0$ F_t is CDF of PDF f_t. According to Theorem 1 $F_t(a_t) = U$, where U is random variable from $\mathcal{U}(0,1)$. Therefore,

$$I_2(a_t) = \int_{-\infty}^{a_t} \left(f(a;t) - f(a;t-h)\right) \mathrm{d}a = U - F_{t-h}(a_t), \tag{26}$$

where U and a_t are random variables from $\mathcal{U}(0,1)$ and f_t respectively. Furthermore, according to Glivenko-Cantelli Theorem of uniform convergence of the empirical distribution functions,

$$F_{t-h}(a_t) \approx \frac{1}{n}\sum_{i=1}^{n} I_{\{a_i^{t-h} \leq a_t\}}, \tag{27}$$

where $a_{t-h}^1, ..., a_{t-h}^n$ is a random sample from f_{t-h}. When (27) is applied to (26) it follows

$$I_2(a_t) \approx U - \frac{1}{n}\sum_{i=1}^{n} I_{\{a_{t-h}^i \leq a_t\}},$$

where a_t and U are random variables from f_t and $\mathcal{U}(0,1)$ respectively and $a_{t-h}^1, ..., a_{t-h}^n$ a random sample from f_{t-h}. Hence, (23) can be approximated with

$$G_t(a_t) \approx \frac{1}{hf(a_t;t)}\left(U - \frac{1}{n}\sum_{i=1}^{n} I_{\{a_{t-h}^i \leq a_t\}}\right),$$

where a_t and U are random variables from f_t and $\mathcal{U}(0,1)$ respectively and $a_{t-h}^1, ..., a_{t-h}^n$ a random sample from f_{t-h}.

Algorithm 1.
Let $T > 0$ be such that the random number a_T from f_T is the target approximation of a global minimum of function g. Furthermore, let $h > 0$ be step size and n length of the random sample $a_t^1, .., a_t^n$ from f_t, $t > 0$, such that the sample mean $\overline{a_t}$ is target approximation of expectation of random variable a_t from f_t. One should notice that with target density index T, and step size h, algorithm will have $T/h =: m$ iterations. Number of iteration will be denoted with $k \in \mathbb{N}_0$.

$k = 0$: Generate random sample of length $(m-1)n$ from f_0, that is, from standard normal distribution and sort it in the increasing order, i.e.

$$a_0^1, a_0^2, ..., a_0^n, a_0^{n+1}, ..., a_0^{(m-1)n}.$$

$k = 1$: Transform generated sample from $\mathcal{N}(0,1)$ into sample from f_1 the in following

way:

$$
\begin{aligned}
a_1^1 &= a_0^n - \frac{h}{nf_t(a_0^n)} \sum_{\{i:a_0^i \le a_0^n\}} (\mu_0 - g(a_0^i)) \\
&= a_0^n - \frac{h}{nf_0(= a_0^n)} \sum_{i=1}^{n} (\mu_0 - g(= a_0^i)) \\
a_1^2 &= a_0^{n+1} - \frac{h}{nf_0(a_0^{n+1})} \sum_{i=1}^{n+1} (\mu_0 - g(a_0^i)) \\
&\vdots \\
a_1^{(m-1)n-(n-1)} &= a_1^{(m-2)n+1} = a_0^{(m-1)n} - \frac{h}{nf_t(a_0^{(m-1)n})} \sum_{i=1}^{(m-1)n} (\mu_0 - g(a_0^i))
\end{aligned}
$$

We obtain sample $a_1^1, a_1^2, ..., a_1^n, a_1^{n+1}, ..., a_1^{(m-2)n+1}$, from f_1.

What is important to notice is that $a_1^1, a_1^2, ..., a_1^n, a_1^{n+1}, ..., a_1^{(m-2)n+1}$ are no longer independent. Hence, law of large numbers does not guarantee that sample mean converges to expectation. Moreover, in each iteration of algorithm dependence between components of obtained sample increases. After certain number of steps, that might result with unusable sample. For that reason we will turn to slightly different approach.

5.2.2 Monte Carlo integration using $\mathcal{N}(0,1)$

As it became clear in previous section, approximation of integrals in (24) and (25) with sample mean of random sample from f_t turned out to be impractical. Therefore, we will try with slightly different approach. In that manner

$$
\begin{aligned}
I_1(a_t) &= \int_{-\infty}^{a_t} \big(\mu_t - g(a)\big) f(a;t) \mathrm{d}a \\
&= \int_{-\infty}^{a_t} (\mu_t - g(a)) \frac{e^{-\gamma(t)g(a)}}{c(t)} I_{(-\infty,a_t]}(a) f(a;0) \mathrm{d}a \\
&= E_{\mathcal{N}(0,1)} \left[(\mu_t - g(a)) \frac{e^{-\gamma(t)g(a)}}{c(t)} I_{(-\infty,a_t]}(a) \right], \qquad (28)
\end{aligned}
$$

where $E_{\mathcal{N}(0,1)} \left[(\mu_t - g(a)) \frac{e^{-\gamma(t)g(a)}}{c(t)} I_{(-\infty,a_t]}(a) \right]$ is expectation with respect to random variable a from $\mathcal{N}(0,1)$. Hence, integral in (24) may be approximated with

$$
I_1(a_t) \approx \sum_{\{i:a^i \le a_t\}} \frac{\mu_t - g(a^i)}{c(t)} e^{-\gamma(t)g(a^i)}. \qquad (29)
$$

By applying (29) to (21) we obtain the following approximation

$$
G_t(a_t) \approx \frac{\gamma'(t)}{nf_t(a_t)} \sum_{\{i:a^i \le a_t\}} \frac{\mu_t - g(a^i)}{c(t)} e^{-\gamma(t)g(a^i)} = \frac{\gamma'(t)e^{\gamma(t)g(a_t)}}{nf_0(a_t)} \sum_{\{i:a^i \le a_t\}} \big(\mu_t - g(a^i)\big) e^{-\gamma(t)g(a^i)},
$$

where $a^1, ..., a^n$ is random sample from $\mathcal{N}(0,1)$.
Even though this idea in theory works just fine, in practice it shows some major issues. Problem occurs with functions whose global minima is not likely to be a number from standard normal distribution. No matter how large sample size n we choose, we won't be able to reach enough of those $a^i \in \mathbb{R}$ such that for large t, $e^{-\gamma(t)g(a^i)}$ is significantly larger than zero. Since $a \mapsto e^{-\gamma(t)g(a)}$ reaches its maximum at minima of function g and since a_t is the best approximation of global minima of g in step t, the idea is to express for every $t > 0$ the integral in (24) as expectation with respect to random variable from $\mathcal{N}(a_t, 1)$. Notation $\mathcal{N}(a_t, 1)$ stands for normal distribution $\mathcal{N}(a, 1)$ conditionally to $a_t = a$, where a_t is random variable from f_t.

5.2.3 Monte Carlo integration using standard normal with drift

The third approach we consider is to express (24) and (25) as expectations of proper transformation of variable from $\mathcal{N}(a_t, 1)$. Therefore, let $\tilde{f}_t$ be PDF of $\mathcal{N}(a_t, 1)$. Since $\tilde{f}_t > 0$, the integral in (24) can be expressed as

$$\begin{aligned}
I_1(a_t) &= \int_{-\infty}^{a_t} (\mu_t - g(a))\, f(a;t)\mathrm{d}a \\
&= \int_{-\infty}^{a_t} (\mu_t - g(a))\, \frac{e^{-\gamma(t)g(a)}}{c(t)} f_0(a)\mathrm{d}a \\
&= \int_{-\infty}^{a_t} (\mu_t - g(a))\, \frac{e^{-\gamma(t)g(a)}}{c(t)} \frac{f_0(a)}{\tilde{f}_t(a)} \tilde{f}_t(a)\mathrm{d}a \\
&= \int_{-\infty}^{a_t} \frac{\mu_t - g(a)}{c(t)} w_t(a)\tilde{f}_t(a)\mathrm{d}a, \text{ where } w_t(a) = e^{-\gamma(t)g(a)} \frac{f_0(a)}{\tilde{f}_t(a)} = e^{\frac{a^2}{2} - aa_t - \gamma(t)g(a)} \\
&= E_{\mathcal{N}(a_t,1)} \left[\frac{\mu_t - g(a)}{c(t)} w_t(a) I_{(-\infty, a_t]}(a) \right],
\end{aligned}$$

where $E_{\mathcal{N}(a_t,1)}\big[\frac{\mu_t - g(a)}{c(t)} w_t(a) I_{(-\infty,a_t]}\big]$ is expectation with respect to random variable a from $\mathcal{N}(a_t, 1)$, for a_t random variable from f_t. Therefore, if $a_t^1, ..., a_t^n$ is random sample from $\mathcal{N}(a_t, 1)$, then integral in (24) may be approximated with

$$E_{\mathcal{N}(a_t,1)} \left[\frac{\mu_t - g(a)}{c(t)} w_t(a) I_{(-\infty,a_t]}(a) \right] \approx \frac{1}{n} \sum_{\{i:a_t^i \le a_t\}} \frac{\mu_t - g(a_t^i)}{c(t)} w_t(a_t^i). \tag{30}$$

Furthermore, when (30) is applied to (21) it follows

$$G_t(a_t) \approx \frac{\gamma'(t)}{n f_t(a_t)} \sum_{\{i:a_t^i \le a_t\}} \frac{\mu_t - g(a_t^i)}{c(t)} w_t(a_t^i),$$

where $a_t^1, ..., a_t^n$ is random sample from $\mathcal{N}(a_t, 1)$.

Integral in (25) will be approximated in similar way. First, according to (26)

$$\begin{aligned} I_2(a_t) &= U - \int_{-\infty}^{a_t} f(a; t-h)\mathrm{d}a \\ &= U - \int_{-\infty}^{a_t} \frac{f(a;t-h)}{\tilde{f}(a)}\tilde{f}(a)\mathrm{d}a \\ &= U - \int_{-\infty}^{a_t} \frac{e^{\frac{a_t^2}{2}-aa_t-\gamma(t-h)g(a)}}{c(t-h)}\tilde{f}(a)\mathrm{d}a \\ &= U - E_{\mathcal{N}(a_t,1)}\left[\frac{e^{\frac{a_t^2}{2}-aa_t-\gamma(t-h)g(a)}}{c(t-h)} I_{(-\infty,a_t]}(a)\right], \end{aligned} \tag{31}$$

where $E_{\mathcal{N}(a_t,1)}\left[\frac{e^{\frac{a_t^2}{2}-aa_t-\gamma(t-h)g(a)}}{c(t-h)} I_{(-\infty,a_t]}(a)\right]$ is expectation with respect to random variable a from $\mathcal{N}(a_t, 1)$ and U random variable from $\mathcal{U}(0, 1)$. Therefore, the integral in (31) can be approximated with

$$I_2(a_t) = U - \sum_{\{i:a_t^i \le a_t\}} \frac{e^{\frac{a_t^2}{2}-a_t^i a_t-\gamma(t-h)g(a_t^i)}}{c(t-h)}, \tag{32}$$

where $a_t^1, ..., a_t^n$ is random sample from $\mathcal{N}(a_t, 1)$. a_t and U are random variables from f_t and $\mathcal{U}(0, 1)$ respectively. Furthermore, when (32) is applied to (23) we obtain

$$G_t(a_t) \approx \frac{1}{hf_t(a_t)}\left(U - \frac{1}{nc(t-h)} \sum_{\{i:a_t^i \le a_t\}} e^{\frac{a_t^2}{2}-a_t^i a_t-\gamma(t-h)g(a_t^i)}\right), \tag{33}$$

where $a_t^1, ..., a_t^n$ is random sample from $\mathcal{N}(a_t, 1)$, a_t and U are random variables from f_t and $\mathcal{U}(0, 1)$ respectively.

Algorithm 2.
Let $T > 0$ be such that the random number a_T from f_T is the target approximation of a global minimum of function g. Furthermore, let $h > 0$ be step size and n length of the random sample $a_t^1, .., a_t^n$ from $\mathcal{N}(a_t, 1)$, $t > 0$, such that sample mean $\overline{a_t}$ is target approximation of expectation of random variable a_t from PDF f_t. Number of iteration will be denoted with $k \in \mathbb{N}_0$.
To simplify the notation, the indices in the following will be denoted as if $h = 1$. More precisely, we write $f_l, a^l, c(l)...$, to denote $f_t, a^t, c(t)...$ for $t = (l-1)h$, i.e. $l = 1$ corresponds to $t = h$, $l = 2$ corresponds to $t = 2h$ etc.

$k = 0$: Generate $a_1^0, ..., a_n^0$, random sample from $\mathcal{N}(0, 1)$, $\mu_0 := 0$

$k = 1$: For each $i \in \{1, ..., n\}$

1. generate random sample $a_1'^i, ..., a_n'^i$ from $\mathcal{N}(0, 1)$
2. $G_0(a_i^0) = \frac{e^{\gamma(0)g(a_i^0)}}{nf_0(a_i^0)} \sum_{\{j:a_j'^i \le a_i^0\}} (\mu_0 - g(a_j'^i))w_0(a_j'^i)$
3. $a_i^1 = a_i^0 - hG_0(a_i^0)$

We obtain random sample $a_1^1, ..., a_1^n$ from f_1.

$\vdots$

$k = l + 1 : \mu_l = \dfrac{1}{n} \sum_{i=1}^{n} a_i^l$

For each $i \in \{1, ..., n\}$

1. generate random sample $a_1'^i, ..., a_n'^i$ from $\mathcal{N}(0, a_l)$
2. $G_l(a_i^l) = \dfrac{e^{\gamma(l) g(a_i^l)}}{n f_0(l)(a_i^l)} \sum\limits_{\{j : a_j'^i \leq a_i^l\}} (\mu_l - g(a_j'^i)) w_l(a_j'^i)$
3. $a_i^{l+1} = a_i^l - h G_l(a_i^l)$

We obtain random sample $a_{l+1}^1, ..., a_{l+1}^n$ from f_{l+1}

Since $a_l^i = a_l^i(a_i^1, ..., a_i^n, \mu_{l-1})$, then for every i, a_l^i is function of $\mu_{l-1} = \mu_{l-1}(a_{l-1}^1, ..., a_{l-1}^n)$. We notice slight dependence between $a_l^1, ..., a_l^n$ which could possibly create issues with convergence of sample mean to expectation. Moreover, it is clear that for approximation of μ_l to be close enough to theoretical value of μ_l , n must be reasonably large. Besides that, we need to transform n numbers form $\mathcal{N}(0, 1)$ into numbers from f_t, in order to approximate mean μ_t. If we could avoid that by approximating μ_t in a different way, we would speed up algorithm up to n times. In that manner, since

$$\mu_t := \int_{\mathbb{R}} \frac{g(a)}{c(t)} e^{-\gamma t g(a)} f(0; a) \mathrm{d}a = \int_{\mathbb{R}} \left(\frac{g(a)}{c(t)} w_t(a) \right) \tilde{f}_t(a) \mathrm{d}a = E_{\tilde{f}_t} \left[\frac{g(a)}{c(t)} w_t(a) \right],$$

where $E_{\tilde{f}_t} \left[\dfrac{g(a)}{c(t)} w_t(a) \right]$ is expectation with respect to random variable a from $\mathcal{N}(a_t, 1)$, then μ_t can be approximated with

$$\mu_t \approx \frac{1}{n} \sum_{i=1}^{n} \frac{g(a_t^i)}{c(t)} w_t(a_t^i),$$

where $a_t^1, ..., a_t^n$ is random sample from $\mathcal{N}(a_t, 1)$. By avoiding approximation of μ_t via sample mean of random sample from f_t, we have solved both dependence issue and we have reduced transformation of entire n-dimensional sample into transformation of only one number from $\mathcal{N}(0, 1)$ into number from f_T. Using this result we obtain the following algorithm:

Algorithm 3.

$k = 0$: Generate a_0 from $\mathcal{N}(0, 1)$, $\mu_0 := 0$

$k = 1 : a_0$ is random number from f_0

1. Generate random sample $a_0^1, ..., a_0^n$ from $\mathcal{N}(0, 1)$
2. $G_0(a_0) = \dfrac{\gamma'(0) e^{\gamma(0) g(a_0)}}{n f_0(a_0)} \sum\limits_{\{i : a_0^i \leq a_0\}} (\mu_0 - g(a_0^i)) w_t(a_0^i)$
3. $a_1 = a_0 - h G_0(a_0)$

We obtain a_1, from f_1.

$\vdots$

$k = l + 1 : a_l$ is random number from f_l

1. Generate random sample $a_l^1, ..., a_l^n$ from $\mathcal{N}(a_l, 1)$
2. $\mu_l = \frac{1}{n} \sum_{i=1}^{n} \frac{g(a_l^i)}{c(l)} w_l(a_l^i)$
3. $G_l(a_l) = \frac{\gamma'(l) e^{\gamma(l) g(a_l)}}{n f_0(a_l)} \sum_{\{i: a_l^i \leq a_l\}} (\mu_l - g(a_l^i)) w_l(a_l^i)$
4. $a_{l+1} = a_l - h G_l(a_l)$

We obtain a_{l+1} from f_{l+1}.

Remark 12. Even though it may not seem that way, necessity of calculation of c_t in every step of algorithm is a huge problem. Let us for example take T to be only 100, $h = 0.01$ and $\gamma(t) = t$. That brings us to around 10000 iterations of algorithm, in which c_t has to be calculated. This motivates us to approximate c_t using MCMC method as well.

Since

$$c(t) = \int_{\mathbb{R}} e^{-\gamma(t) g(a)} f_0(a) \mathrm{d}a = \int_{\mathbb{R}} w_t(a) \tilde{f}_t(a) \mathrm{d}a = E_{\tilde{f}_t}[w_t(a)],$$

where $E_{\tilde{f}_t}[w_t(a)]$ denotes expectation with respect to random variable a from $\mathcal{N}(a_t, 1)$, then it may be approximated with sample mean of the appropriate transformation of a random sample from $\mathcal{N}(a_t, 1)$ as well. This brings us to another AA algorithm:

Algorithm 4.

$k = 0$: Generate a_0 from $\mathcal{N}(0, 1)$, $\mu_0 := 0$

$k = 1 : a^0$ is random number from f_0

1. Generate random sample $a_0^1, ..., a_0^n$ from $N(0, 1)$
2. $G_0(a_0) = \frac{\gamma'(0) e^{\gamma(0) g(a_0)}}{n f_0(a_0)} \sum_{\{i: a_0^i \leq a_0\}} (\mu_0 - g(a_0^i)) w_0(a_0^i)$
3. $a_1 = a_0 - h G_0(a_0)$

We obtain a_1, from f_1

$\vdots$

$k = l + 1 : a_l$ is random number from f_l

1. Generate random sample $a_l^1, ..., a_l^n$ from $\mathcal{N}(a_l, 1)$

2. $c_l = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} w_l(a_l^i)$

2. $\mu_l = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \dfrac{g(a_l^i)}{c_l} w_l(a_l^i)$

3. $G_l(a_l) = \dfrac{\gamma'(l) e^{\gamma(l) g(a_l)}}{n f_0(a_l)} \displaystyle\sum_{\{i: a_l^i \leq a_l\}} (\mu_l - g(a_l^i)) w_l(a_l^i)$

3. $a_{l+1} = a_l - h G_l(a_l)$

We obtain a_{l+1}, from f_{l+1}

In case where we use (33) as approximation of function G_t we obtain the following algorithm

Algorithm 5. Let $T > 0$ be such that a random number a_T from distribution f_T is target approximation of global minimum of function g. Furthermore, let $h > 0$ be step size and n length of random sample $a_t^1, .., a_t^n$ from $\mathcal{N}(a_t, 1)$, $t > 0$, such that sample mean $\overline{w_t(a_t)}$ is target approximation of c_t. Number of iteration will be denoted with $k \in \mathbb{N}_0$.

$k = 0$: Generate a_0 from $\mathcal{N}(0, 1)$, $c(0) = 1$

$k = 1$: Using Algorithm 4 in iteration $k = 1$ transform a_0 into a_1 from f_1

$k = 2 : a_0$ and a_1 are random numbers from f_0 and f_1 respectively

1. Generate random sample $a_1^1, ..., a_1^n$, from $\mathcal{N}(a_1, 1)$
2. Generate random number u from $\mathcal{U}(0, 1)$
3. $c(1) = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} w_1(a_1^i)$
4. $G_1(a_1) = \dfrac{1}{h f_1(a_1)} \left(u + \dfrac{1}{nc(0)} \displaystyle\sum_{\{i: a_1^i \leq a_1\}} e^{\frac{a_1^2}{2} - a_1^i a_1 - \gamma(0) g(a_1^i)} \right)$
5. $a_2 = a_1 - G_1(a_1)$

We obtain a_2 from f_2

$\vdots$

$k = l + 1$: a_l and a_{l-1} are random numbers from f_l and f_{l-1} respectively, $c(l-1)$ is standardizing constant of f_{l-1}

1. Generate random sample $a_l^1, ..., a_l^n$ from $\mathcal{N}(a_l, 1)$
2. Generate random number u from $\mathcal{U}(0, 1)$
3. $c(1) = \frac{1}{n} \sum_{i=1}^{n} w_l(a_l^i)$
4. $G_l(a_l) = \frac{1}{h f_l(a_l)} \left(u + \frac{1}{nc(l-h)} \sum_{\{i: a_l^i \leq a_l\}} e^{\frac{a_l^2}{2} - a_l^i a_l - \gamma(l-1) g(a_l^i)} \right)$
5. $a_{l+1} = a_l - G_l(a_l)$.

5.2.4 Possible issues

For every $t > 0$, G_t is determined such that for a_t from f_t, a_{t+h} defined implicitly with $a_t := a_{t+h} + hG_t(a_{t+h})$ is random variable from f_{t+h}. On the other hand, once $G_t(a_t)$ is determined that way, we calculate a_{t+h} as $a_{t+h} = a_t - hG_t(a_t)$. For h very small, PDF of a_{t+h} is rather close to f_{t+h}, but is not exactly it. Besides that, since a_t is given by analogous transformation of a_{t-h}, the PDF of a_t is not exactly f_t. This could make PDF of a_{t+h} even further from f_{t+h}. Additionally, it is clear that we are, on several places, instead of theoretical, using numerical approximations of certain values. This, combined with the previous discussion implies that it is questionable if $f_t(a_t)$, for large t, numerically differs from zero. This may cause numerical problems since we calculate $G_t(a_t)$ as $\frac{1}{f_t(a_t)} I$, where I is certain approximation of $\int_{-\infty}^{a_t} \partial_t f_t(s) \mathrm{d}s$. Whether this will really be an issue remains to be checked at each particular problem.

5.3 Examples

In this chapter we will illustrate how the presented method works on several examples by implementing Algorithm 4. We will study form of PDFs f_t and trajectory of sequence $(a_t)_t$. Furthermore, we will study the absolute error $(\varepsilon_t)_t = (|a_t - a^*|_t)$, where a^* is global minimum of function g. From the obtained results we will empirically investigate the rate of convergence.

5.3.1 AA minimisations of some functions with a lot of local minima

AA minimization of $g(a) = sin(a)/a$
Using Algorithm 4. we will minimize function

$$g(a) = \begin{cases} \frac{\sin(a)}{a}, & \text{for } a \neq 0, \\ 1, & \text{for } a = 0. \end{cases}$$

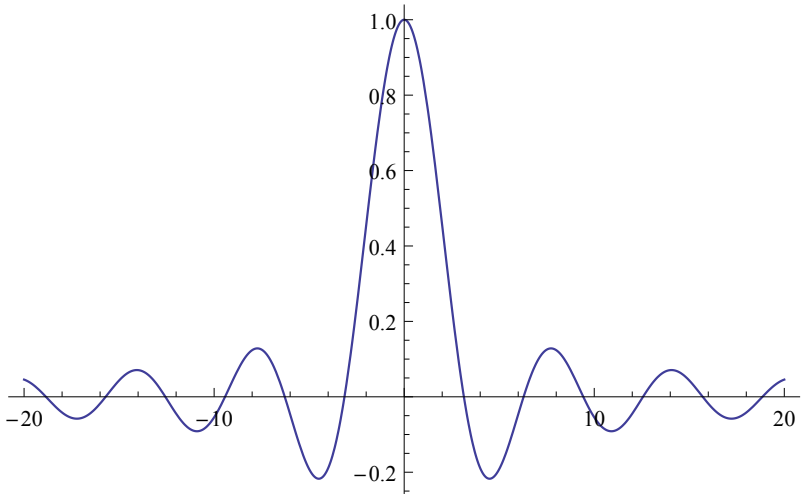
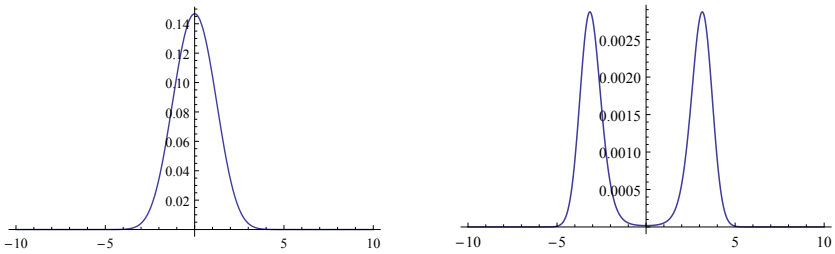


Figure 1: Graph of function g

Function g has infinitely many local minima, but only two global minima. In order to determine these global minima we have first applied Wolfram Mathematica's built-in optimisation method called *FindMinimum*. textitFindMinimum is a procedure based on deterministic optimization methods which as input requires the cost function g and an initial approximation. If we have no information of position of global minima, which is often the case, we would have to guess initial approximation. Given various initial approximations, *FindMinimum* returns different outputs. In tabular form

Initial approximation	100	50	20	10	1
Output a^*	98.9501	48.6741	17.2208	10.9041	4.49341
Function value at a^*	−0.0101056	−0.0205405	−0.0579718	−0.0913252	−0.217234

One can see that for different input parameters, method returns very different approximations of minimum, and this is potentially large issue. Before discussing result obtained by AA method, let us first discuss behaviour of PDFs f_t.



(a) f_0 (b) f_{10}

(c) f_{50} (d) f_{100}

Figure 2: Graphs of nonstandardized PDFs f_1, f_{10}, f_{50} and f_{100}

In Figure 2 convergence of sequence (f_t) towards distribution concentrated in set of global minima of g is nicely seen. Moreover, according to Picture (d) in (2), we would expect that generating number from f_{100} would result with fair approximation of global minimum of g. We will run algorithm with input parameters $h = 0.01$, $n = 500$ and $T = 100$ for three different initial approximations a_0 from f_0. In tabular form

Initial approximation	0.485679	0.623366	1.21226
Output a_{100}	4.32236	4.30985	4.47135
Function value at a_{100}	-0.21398	-0.213481	-0.217181
Error $a^* - a_{100}$	0.171049	0.183559	0.0220595

One should notice that for all three initial approximations, the target approximation is reasonably close to global minimum. Moreover, one should notice that all three approximations have underestimated global minimum. We will notice this behaviour of the algorithm at other examples as well. Let us consider trajectories of approximations.
As one can see in Figure 3, trajectories of approximations a_t have some kind of logarithmic shape.

Definition 11. We say that the sequence $(a_t)_t$ converges linearly to a^*, if there exists a constant $\mu \in (0, 1)$ such that

$$\lim_{t\to\infty} \frac{|a_{t+1} - a^*|}{|a_t - a^*|} = \mu. \tag{34}$$

Moreover, if limit (34) exists and $\mu = 0$ the convergence is superlinear, and if limit (34) exists and $\mu = 1$ the convergence is sublinear. If the sequence converges sublinearly and additionally

$$\lim_{t\to\infty} \frac{|a_{t+2} - a_{t+1}|}{|a_{t+1} - a_t|} = 1,$$

we say that the sequence converges logarithmically to a^*.

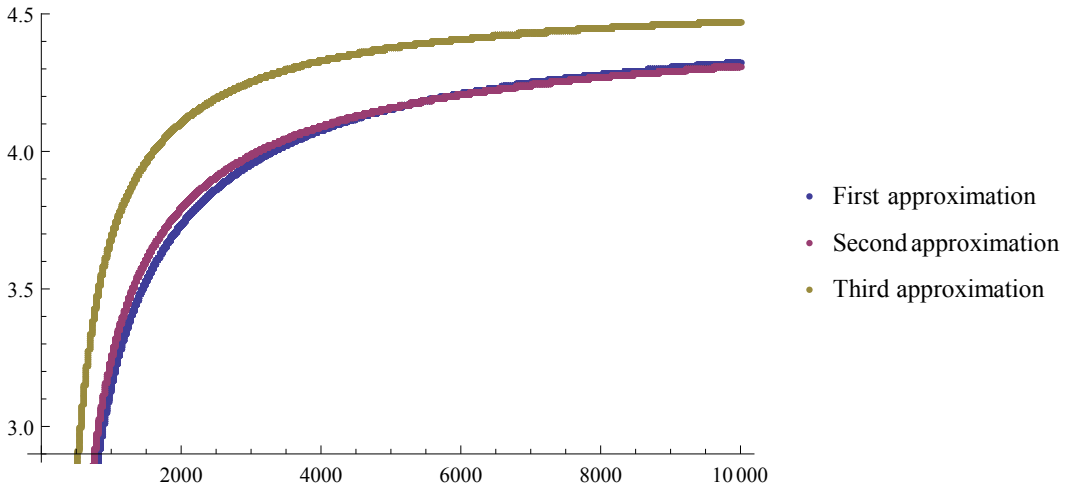


Figure 3: Trajectories of approximations $(a_t)_t$ given three different initial approximations

According to Definition 11, we will consider following sequences. Let $(\varepsilon_t)_t = (|a_t - a^*|)_t$ be sequence of absolute errors. In order to empirically determine rate of convergence we will consider trajectories of $(\varepsilon_{t+1}/\varepsilon_t)_t$.

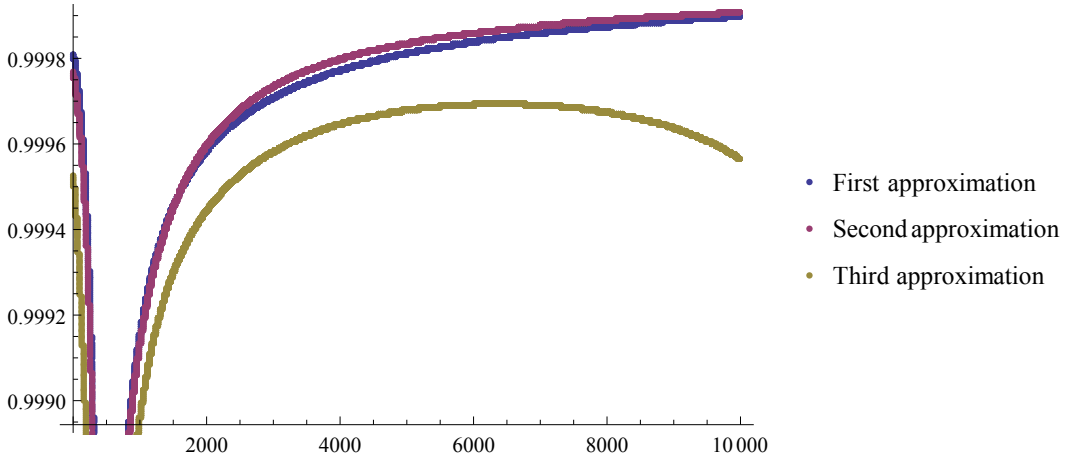


Figure 4: Trajectories of $(\varepsilon_{t+1}/\varepsilon_t)_t$ given three different initial approximations

Due to Figure 4, we could conclude that the rate of convergence in case of first two initial approximations is sublinear. In third case we could guess linear rate, but it would be better to observe what happens with sequence in future. Regardless of the third case, we will stick to the idea of logarithmic rate of convergence. That is, we will observe worst case. Moreover, let us see if we could guess logarithmic convergence rate. In that manner we will consider trajectories of $\left(\dfrac{|a_{t+2} - a_{t+1}|}{|a_{t+1} - a_t|}\right)_t$.

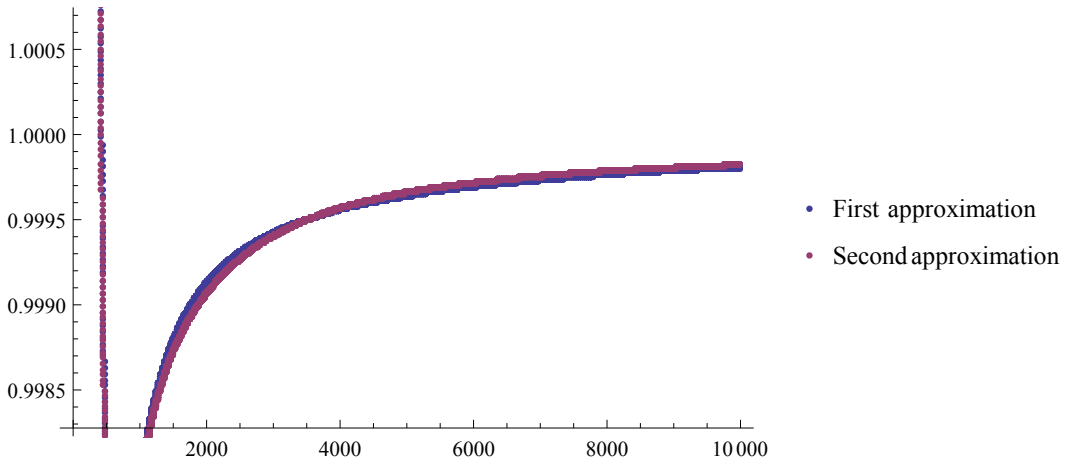


Figure 5: Trajectories of $\left(\frac{|a_{t+2}-a_{t+1}|}{|a_{t+1}-a_t|}\right)_t$ given first two initial approximations

It is evident from Figure 5 that both trajectories converge to $a = 1$. Therefore, we could conclude logarithmic rate.

AA minimization of $g(a) = sin(a - 10)/(a - 10)$
Using Algorithm 4. we will minimize function

$$g(a) = \begin{cases} \frac{\sin(a-10)}{a-10}, & \text{for } a \neq 10, \\ 1, & \text{for } a = 10, \end{cases}$$

which is essentially function from previous example translated for 10. Purpose of this example is to show possible issues connected with algorithm.

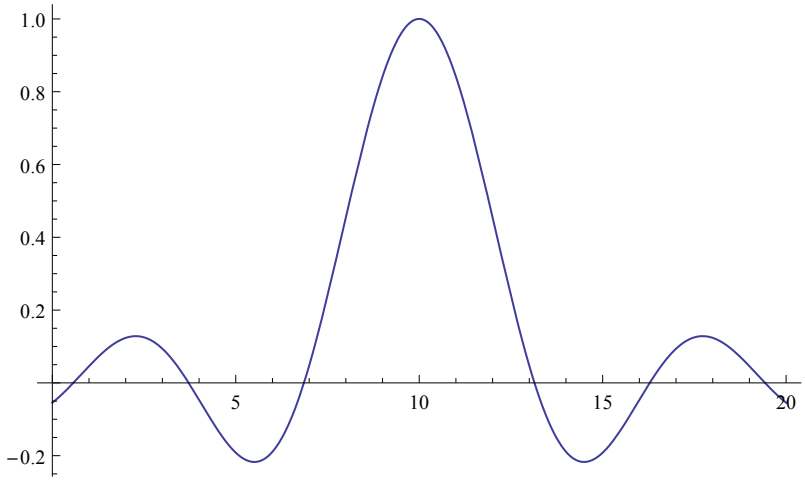
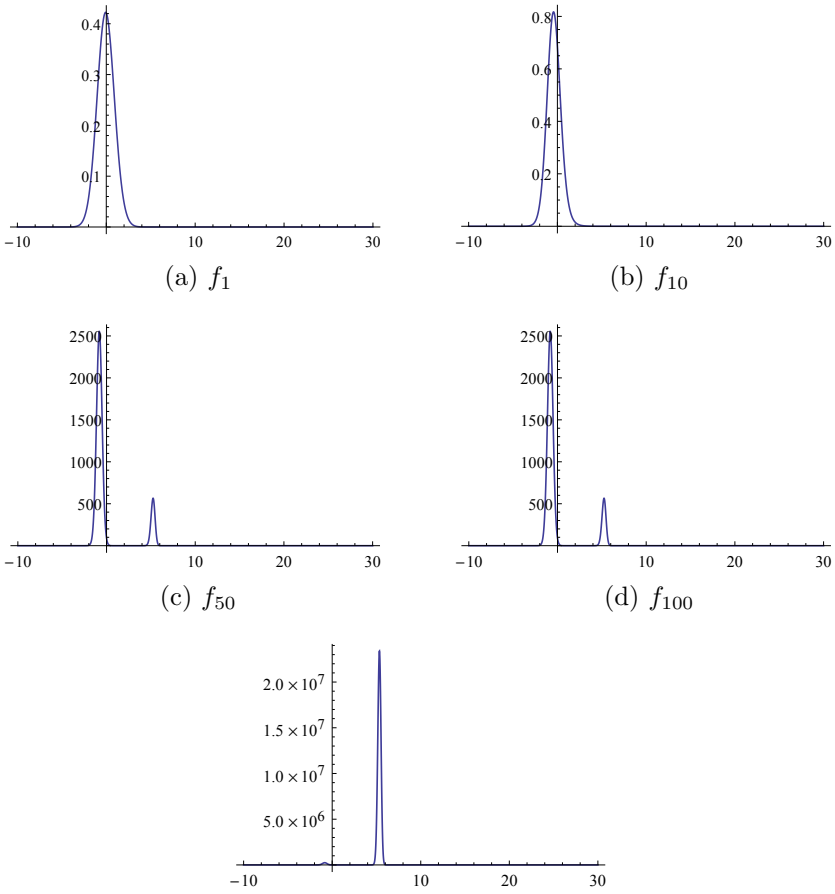


Figure 6: Graph of function g

As in previous example, function g has also infinity many local minima, but only two global minima. As in previous example Wolfram Mathematica's built-in optimisation method called *FindMinimum* is very sensitive to initial approximation and therefore can not determine global minimum. Before discussing result obtained by AA method, let us discuss behaviour of PDFs f_t.



(a) f_1 (b) f_{10}

(c) f_{50} (d) f_{100}

(e) f_{150}

Figure 7: Graphs of nonstandardized PDFs f_1, f_{10}, f_{50}, f_{100}, f_{150}

Unlike in the previous example, convergence of sequence (f_t) towards distribution concentrated in set of global minima of g is not so obvious. Only at f_{150} we see highest peak at global minima of g. Moreover, one should notice that g has two local minima, and only one of them is visible at f_{150}. Reason to that is factor f_0 in expression (14) of f_t. If global minima a^* of function g is not likely to be number from $\mathcal{N}(0,1)$, then $f_t(a^*) = \frac{e^{\gamma(t)a^*}}{c_t} f_0(a^*)$ will be close to zero if factor $\dfrac{e^{\gamma(t)a^*}}{c_t}$ is not influential enough over $f_0(a^*)$.

We will run algorithm with input parameters $h = 0.01$, $n = 500$ and $T = 150$ for three different initial approximations a_0 from f_0. In tabular form

Initial approximation	0.485679	0.623366	1.21226
Output a_{150}	-0.722768	-0.662596	-0.493262
Function value at a_{150}	-0.0898106	-0.0886344	-0.0835271
Error $a^* - a_{150}$	6.22936	6.16919	5.99985

In this case we notice some serious deviations from global minimum. To get better intuition why this happens, let us consider trajectories of approximations.

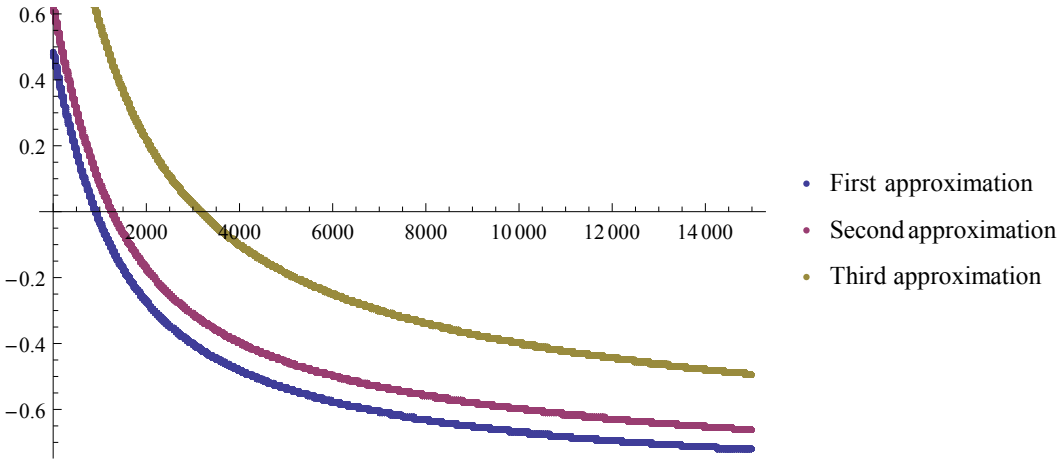


Figure 8: Trajectories of approximations $(a_t)_t$ given three different initial approximations

As visible in Figure 8, trajectories of approximations a_t simply deviate from global minimum. It is understandable, according to Figure 7.d why would we obtain such results for $T = 100$, but according to Figure 7.e we would expect solutions very close to global minimum. Since term $G_t(a_t)$ is what moves a_{t+h} from a_t, we will consider trajectories of sequence $(G_t(a_t))_t$ in order to obtain better intuition into this behaviour.

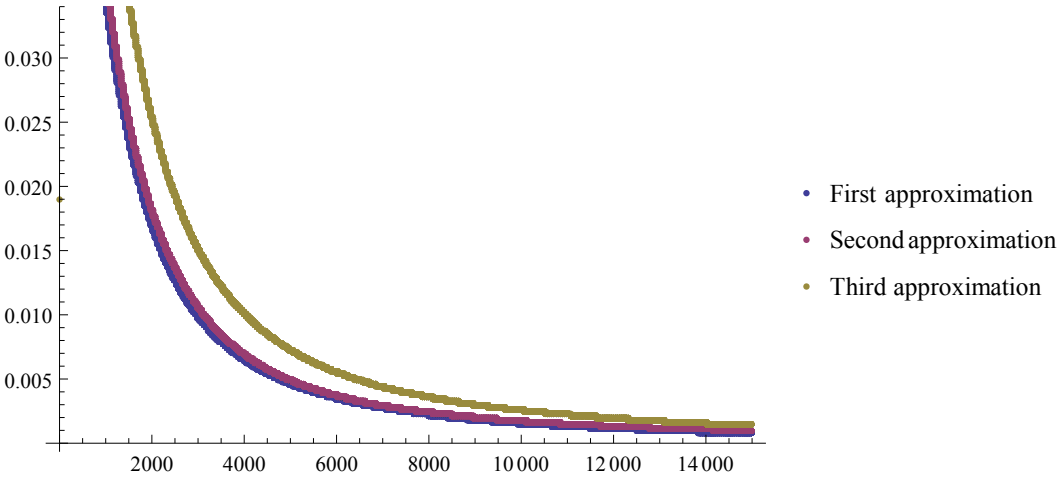


Figure 9: Trajectories of sequence $(G_t(a_t))_t$ given three different initial approximations

It is evident that all three trajectories are essentially zero, after 10000 steps. At $T = 100$ algorithm generated approximation a_{100} from f_{100} which is due to f_{100} very close to zero. Because of behaviour of $(G_t(a_t))_t$ after $T = 100$, that is after 10000 steps, algorithm does not have "enough power" to shift a_t towards global minimum.

6 Conclusion

We have provided a summary of several methods for sampling from distributions and connected them with global optimization problem and presented some stochastic optimization algorithms such as Simulated Annealing for functions defined on discrete sets, and its generalisation to functions defined on more general sets. Furthermore we have concluded that global minimum of function can be well approximated by generating number from Gibbs distribution with parameter large enough.

In that manner we have presented Adaptive Annealing method for stochastic optimization. We have paid special attention to approximation of certain integrals in order to implement algorithm and test it on concrete examples. At the end, we have presented algorithm on two examples. In first one, we have shown the ability of algorithm to escape local minima in order to find global one. We used second example to show issues that algorithm faces while optimizing functions whose minimum is not likely to be from $\mathcal{N}(0,1)$ and when function value at global minimum does not differ much from values around zero.

References

[1] A. R. BARRON, X. LUO, *Adaptive Annealing*, Fourty-Fifth Annual Allerton Conference, Allerton House, UIUC, Illinois, USA, September 26-28, 2007, 665-673

[2] D. RERTSIMAS, J. TSITSIKLIS, *Simulated Annealing*, Statistical Science 1993, Vol. 8, No. 1 (1993), 10-15

[3] L. C. EVANS, *Partial Differential Equations*, Graduate Studies in Mathematics, Vol. 19, American Mathematical Society

[4] S. GEMAN, C.R. HWANG, *Diffusions for global optimization*, Siam J. control and optimization Vol. 24, No. 5 (September 1986), 1031-1043.

[5] B. GIDAS, *The Langevin Equation as a Global Minimization Algorithm*, Disordered Systems and Biological Organization 1986, 321-326.

[6] H. HAARIO, E. SAKMAN, *Weak convergence of the Simulated Annealing in general state space*, Annales Academie Scientiarum Fennica Series A. I. Mathematica Vol. 17 (1992), 39-50

[7] W. K. HASTINGS, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Biometrika Vol. 57, No. 1 (Apr., 1970), 97-109.

[8] L. MARTINO, J. MIGUEZ, *Generalized rejection sampling schemes and applications in signal processing*, Signal Processing Volume 90, Issue 11 (November 2010), 2981–2995.

[9] R. SCITOVSKI, N. TRUHAR, Z. TOMLJANOVIĆ, *Metode optimizacije*, Odjel za matematiku Sveučilišta Josipa Jurja Strossmayera, Osijek, 2012.

[10] J. C. SPALL, *Stochastic Optimization*, Handbook of Computational Statistics, Springer Heidelberg 2004, 170-176

[11] Z. VONDRAČEK, *Markovljevi lanci*, predavanja, 30. rujna 2008.

[12] *Weak Convergence, Course: Theory of Probability I, Term: Fall 2013, Instructor: Gordan Zitkovic*, avaliable on `https://www.ma.utexas.edu/users/gordanz/notes/weak.pdf`, last time visited on August 20, 2017

[13] *Introduction to Image Processing, University of Tartu*, avaliable on `https://sisu.ut.ee/imageprocessing/book/1`, last time visited on August 20, 2017

Abstract
This work provides a summary of several methods for sampling from distributions and connects them with global optimization problem. We will present some stochastic optimization algorithms such as Simulated Annealing for functions defined on discrete sets, and its generalisation to functions defined on more general sets. At the end, we will present the one-dimensional Adaptive Annealing method and we will test its preformance on functions with a lot of local minima.

Key words *Markov Chain, MCMC, Metropolis-Hastings, Simulated Annealing, Gibbs distribution, Adaptive Annealing, global optimization*

Sažetak
U ovom radu predstavit ćemo nekoliko metoda za uzorkovanje iz distribucije te ćemo ih povezati s problemom globalne optimizacije. Predstavit ćemo nekoliko stohastičkih optimizacijskih algoritama kao što su simulirano kaljenje za funkcije definirane na diskretnom setu, te generalizacije istoga na općenitije skupove. Na kraju, predstavit ćemo metodu adaptivnog kaljenja za jednodimenzionalnu optimizaciju, te ćemo testirati metodu na konkretnim primjerima.

Ključne riječi *Markovljev lanac, MCMC, Metropolis-Hastings, simulirano kaljenje, Gibbsova distribucija, adaptivno kaljenje, globalna optimizacija*

Biography

I was born on August 2nd in 1993 in Slavonski Brod, Croatia. In 2008 I completed elementary school education in Slavonski Brod and enrolled in Matija Mesić Gymnasium, Slavonski Brod. In 2012, I enrolled in undergraduate study in mathematics at the Department of Mathematics of Josip Juraj Strossmayer University in Osijek. During my undergraduate study I held lectures in Linear Algebra II, Multivariate Calculus, Complex Analysis Ordinary and Differential Equations. In December 2013 I received "Excellence is IN" award of Rotary Club Slavonski Brod for academic excellence achieved during the academic year 2012/2013. In May 2015 I held presentation in "Primatijada 2015" on topic Motivational Problems and elementariness of Calculus of Variations. At the end of July 2015, I participated in an International Mathematical Competition (IMC). In September 2015 I successfully completed my studies with the final work "Calculus of Variations" under the mentor Dr. sc. Krešimir Burazin. In October 2015, I enrolled in graduate study in mathematics, Financial Mathematics and Statistics at the Department of Mathematics of Josip Juraj Strossmayer University in Osijek. During my graduate studies, I held lectures in Algebra and Graph Theory and I participated as a lecturer in Mathematical Preparations for high-school students organized by the Osijek Association of Mathematicians. In May 2016 I received the Rector's award for the seminar work from the course Ordinary Differential Equations under the title "Calculus of Variations". In December 2016, I received the Lions Club Prize that is annually given to the best students of the final year of study. In May 2017 I received the Rector's award for the seminar work from the course Mathematical Models under the title "A model of AIDS spread". In June 2016 I enrolled DAAD Intensive Course in Approximation Theory and Applications, while in August 2016 I enrolled Scuola Matematica Interuniversitaria in Perugia, Italy. In June 2017 I enrolled Bocconi Summer School in Advanced Statistics and Probability, Statistical Causal Learning. Furthermore, I have published paper in Osječki matematički list 16(2) with Dr. sc. Krešimir Burazin on the topic Introduction to the Calculus of Variations and its history.