A short introduction to stochastic optimizatiom

Jerzy Ombach

Instytut Matematyki, Uniwersytet Jagielloński

Będlewo, February 2015

Main problem

Given set $A \subset \mathbb{R}^n$ and continuous function $f : A \to \mathbb{R}$.

Main problem

Given set $A \subset \mathbb{R}^n$ and continuous function $f : A \to \mathbb{R}$.

$$A^\star = \arg\min f = \{a \in A : f(a) \leqslant f(x) \text{ for all } \in A\}.$$

Main problem

Given set $A \subset \mathbb{R}^n$ and continuous function $f : A \to \mathbb{R}$.

$$A^\star = \arg\min f = \{a \in A : f(a) \leqslant f(x) \text{ for all } \in A\}.$$

If A is compact, then $A^\star$ is nonempty.

Main problem

Given set $A \subset \mathbb{R}^n$ and continuous function $f : A \to \mathbb{R}$.

$$A^\star = \arg\min f = \{a \in A : f(a) \leqslant f(x) \text{ for all } \ \in A\}.$$

If A is compact, then $A^\star$ is nonempty.

The problem

How to find points that approximate $A^\star$.

Examples of global optimization problems

Examples of global optimization problems

1. Maximum Likelihood Method

Find parameter $\theta \in \Theta \subset \mathbb{R}^k$ such that the likelihood function $l(\theta)$ takes its maximum value.

Example: $l(\theta) = g_\theta(x^1) \cdot \ \ldots \ \cdot g_\theta(x^n)$, where $x^1, \ldots, x^n$ is a simple sample drawn from density g_θ.

Examples of global optimization problems

1. Maximum Likelihood Method

Find parameter $\theta \in \Theta \subset \mathbb{R}^k$ such that the likelihood function $l(\theta)$ takes its maximum value.

Example: $l(\theta) = g_\theta(x^1) \cdot \ \ldots \ \cdot g_\theta(x^n)$, where $x^1, \ldots, x^n$ is a simple sample drawn from density g_θ.

2. System of equations solving

Find a solution $x_0 \in A \subset \mathbb{R}^n$ of the system:

$$\begin{cases} g^1(x) & = & 0 \\ \ldots & \ldots & \ldots \\ g^k(x) & = & 0 \end{cases}$$

Examples of global optimization problems

1. Maximum Likelihood Method

Find parameter $\theta \in \Theta \subset \mathbb{R}^k$ such that the likelihood function $l(\theta)$ takes its maximum value.

Example: $l(\theta) = g_\theta(x^1) \cdot \ \ldots \ \cdot g_\theta(x^n)$, where $x^1, \ldots, x^n$ is a simple sample drawn from density g_θ.

2. System of equations solving

Find a solution $x_0 \in A \subset \mathbb{R}^n$ of the system:

$$\begin{cases} g^1(x) & = & 0 \\ \ldots & \ldots & \ldots \\ g^k(x) & = & 0 \end{cases}$$

Define function: $f(x) = g^1(x)^2 + \ \ldots \ + g^k(x)^2$ and look for a global minimum of f, $x_0 \in A$.

Examples of global optimization problems

3. Good enough solution

Given function $g : A \longrightarrow \mathbb{R}$ and a number $\alpha > \min g$.
Find effectively $x_0 \in A$ such that

$$g(x_0) \leqslant \alpha.$$

Examples of global optimization problems

3. Good enough solution

Given function $g : A \longrightarrow \mathbb{R}$ and a number $\alpha > \min g$.
Find effectively $x_0 \in A$ such that

$$g(x_0) \leqslant \alpha.$$

Define function: $f(x) = \max(g(x), \alpha)$ and look for a global minimum of f, $x_0 \in A$.

Deterministic or stochastic

Deterministic or stochastic

Deterministic algorithms.

Deterministic or stochastic

Deterministic algorithms.

Stochastic algorithms.

Deterministic or stochastic

Deterministic algorithms.

Stochastic algorithms.

There exists a lot of numerical optimization procedures. Even fifty years ago most of them were deterministic methods. However, with the spread of computers, stochastic methods have appeared and in recent years we have been witnessing an explosion of heuristic stochastic algorithms.

Deterministic or stochastic

Deterministic algorithms.

Stochastic algorithms.

There exists a lot of numerical optimization procedures. Even fifty years ago most of them were deterministic methods. However, with the spread of computers, stochastic methods have appeared and in recent years we have been witnessing an explosion of heuristic stochastic algorithms.

What are HEURISTICS?

Deterministic or stochastic

Deterministic algorithms.

Stochastic algorithms.

There exists a lot of numerical optimization procedures. Even fifty years ago most of them were deterministic methods. However, with the spread of computers, stochastic methods have appeared and in recent years we have been witnessing an explosion of heuristic stochastic algorithms.

What are HEURISTICS?

A heuristic is understood to be a rule of thumb learned from experience but not always justified by an underlying theory.

Deterministic or stochastic

Deterministic algorithms.

Stochastic algorithms.

There exists a lot of numerical optimization procedures. Even fifty years ago most of them were deterministic methods. However, with the spread of computers, stochastic methods have appeared and in recent years we have been witnessing an explosion of heuristic stochastic algorithms.

What are HEURISTICS?

A heuristic is understood to be a rule of thumb learned from experience but not always justified by an underlying theory.

Metaheuristic – designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution.

Metaheuristics generate at step t random n-dimensional points, say X_t. We are interested in convergence of the sequence X_t to $A^\star$.

Metaheuristics generate at step t random n-dimensional points, say X_t. We are interested in convergence of the sequence X_t to $A^\star$.

We consider here two types od such convergence.

Metaheuristics generate at step t random n-dimensional points, say X_t. We are interested in convergence of the sequence X_t to $A^\star$.

We consider here two types od such convergence.

Stochastic convergence

.

$$\forall \varepsilon > 0 \quad \text{Prob}(dist(X_t, A^*) < \varepsilon) \ \rightarrow 1, \text{ as } t \longrightarrow \infty,$$

Metaheuristics generate at step t random n-dimensional points, say X_t. We are interested in convergence of the sequence X_t to $A^\star$.

We consider here two types od such convergence.

Stochastic convergence

.

$$\forall \varepsilon > 0 \quad \text{Prob}(dist(X_t, A^*) < \varepsilon) \ \rightarrow 1, \text{ as } t \longrightarrow \infty,$$

Almost sure convergence

$$\text{Prob}\left(X_t \rightarrow A^\star, \text{ as } t \rightarrow \infty\right) = 1,$$

$$\text{i.e. Prob}\left(\{\omega \in \Omega : \ \text{dist}(X_t(\omega), A^\star) \longrightarrow 0, \text{ as } t \longrightarrow \infty\}\right) = 1,$$

where $\text{dist}(x, K)$ denotes the distance x from K.

Metaheuristics generate at step t random n-dimensional points, say X_t. We are interested in convergence of the sequence X_t to $A^\star$.

We consider here two types od such convergence.

Stochastic convergence

.

$$\forall \varepsilon > 0 \quad \text{Prob}(dist(X_t, A^*) < \varepsilon) \ \rightarrow 1, \text{ as } t \longrightarrow \infty,$$

Almost sure convergence

$$\text{Prob}\left(X_t \rightarrow A^\star, \text{ as } t \rightarrow \infty\right) = 1,$$

$$\text{i.e. Prob}\left(\{\omega \in \Omega : \ \text{dist}(X_t(\omega), A^\star) \longrightarrow 0, \text{ as } t \longrightarrow \infty\}\right) = 1,$$

where $\text{dist}(x, K)$ denotes the distance x from K.

Almost sure cinvergence $\Longrightarrow$ stochastic convergence.

Pure Random Search, PRS

$f : A \to \mathbb{R}$ is continuous where $A = [0,1]^n \subset \mathbb{R}^n$.

Algorithm

0. Set $t = 0$. Generate a point x_0 from the uniform distribution on A.
1. Given x_t, generate y_t from the uniform distribution on A.
2. If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$.
3. Increase $t := t + 1$ and go to Step 1.

Pure Random Search, PRS

$f : A \to \mathbb{R}$ is continuous where $A = [0,1]^n \subset \mathbb{R}^n$.

Algorithm

0. Set $t = 0$. Generate a point x_0 from the uniform distribution on A.
1. Given x_t, generate y_t from the uniform distribution on A.
2. If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$.
3. Increase $t := t + 1$ and go to Step 1.

Let X_t, $t = 0, 1, 2, \ldots$ be random vectors which realizations are generated by PRS.

Theorem (folklore)

$$Prob\,(X_t \to A^\star, \ as\ t \to \infty) = 1.$$

Accelerated Random Search, ARS

$f : A \to \mathbb{R}$ is continuous and $A = [0,1]^n \subset \mathbb{R}^n$.

Fix $c > 1$ (a shrinking factor), and $\rho > 0$ (a precision threshold).

Algorithm

0. Set $t = 1$ and $r_1 = 1$. Generate x_1 from the uniform distribution on A.
1. Given $x_t \in A$ and $r_t \in (0,1]$, generate y_t from the uniform distribution on $B(x_t, r_t) \cap A$, where $B(x,r)$ is the ball of radius r centered at x.
2. If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$ and $r_{t+1} = 1$.
3. If $f(y_t) \geqslant f(x_t)$, then:
   1. If $r_t \geqslant \rho$, put $x_{t+1} = x_t$ and $r_{t+1} = r_t/c$.
   2. If $r_{t+1} < \rho$, put $r_{t+1} = 1$.
4. Increase $t := t + 1$ and go to Step 1.

Let X_t, $t = 0, 1, 2, \ldots$ be random vectors which realizations are generated by ARS.

Theorem (Tarłowski, 2013)

Assume, that for any $c \in \mathbb{R}$ the level curve $l_c = \{x \in A : f(x) = c\}$ has its Lebesgue measure 0. Then:

$$Prob\,(X_t \to A^\star,\ as\ t \to \infty) = 1.$$

Theorem (M. J. Appel, R. Labarre, D. Radulovic, 2003)

Assume
f has finitely many global minima,
M_t – the record sequence produced by ARS i.e.

$$M_t = \min\{f(X_s) : s = 1 \dots t\}.$$

$\tilde{M}_t$ – the record sequence produced by PRS.
Given a contraction factor $c > 1$ and a precision threshold $\rho \in (0, 1)$.
Let $m = \frac{|\ln \rho|}{\ln c}$.
Then, for each positive integer $C < \frac{c^m}{3m}$ there exists a positive integer t_C, depending only on C, such that for each $t > t_C$:

$$E(M_t) \leqslant E(\tilde{M}_{t_C}).$$

Theorem (M. J. Appel, R. Labarre, D. Radulovic, 2003)

Assume
f has finitely many global minima,
M_t – the record sequence produced by ARS i.e.

$$M_t = \min\{f(X_s) : s = 1 \dots t\}.$$

$\tilde{M}_t$ – the record sequence produced by PRS.
Given a contraction factor $c > 1$ and a precision threshold $\rho \in (0, 1)$.
Let $m = \frac{|\ln \rho|}{\ln c}$.
Then, for each positive integer $C < \frac{c^m}{3m}$ there exists a positive integer t_C, depending only on C, such that for each $t > t_C$:

$$E(M_t) \leqslant E(\tilde{M}_{t_C}).$$

The above theorem says, that one can choose the shrinking factor and the precision constance such that eventually ARS will require less steps than PRS to attain an approximation of the solution which is at leat of the sam quality.

Hybrid algorithms

$f : A \to \mathbb{R}$ is continuous and $A \subset \mathbb{R}^n$ is a compact set.

Hybrid algorithms

$f : A \to \mathbb{R}$ is continuous and $A \subset \mathbb{R}^n$ is a compact set.

μ_0, ν – Borel probabilistic measures on the set A.

Hybrid algorithms

$f : A \to \mathbb{R}$ is continuous and $A \subset \mathbb{R}^n$ is a compact set.

μ_0, ν – Borel probabilistic measures on the set A.

$\varphi : A \to A$ – a (deterministic) local method i.e. $f(\varphi(x)) \leqslant f(x)$ for all $x \in A$.
For example:
the gradient method and a lot of its modification.
the identity map is a local method.

Algorithm

0 Set $t = 0$. Generate a point x_0 from the distribution μ_0 on A.

1 Given point x_t, generate $y_t \in A$ according to the distribution ν.

2 Apply φ to y_t.

3 If $f(\varphi(y_t)) < f(x_t)$, then $x_{t+1} = \varphi(y_t)$.

3 Increase $t := t + 1$ and go to Step 1.

Multistart algorithm

$f : A \to \mathbb{R}$ is continuous and $A \subset \mathbb{R}^n$ is a compact set.

M – the set of all Borel probabilistic measures on A (weak topology on M).
$\mu_0 \in M$
m – size of a current population
k – number of point generated in each step
Φ – a set of local methods,
let $N \subset M$ be compact and let N_0 be a closed subset of N, such that for any $\nu \in N_0$, $\nu(G) > 0$ for any open neighborhood G of the set $A^\star$.

Algorithm

0 Let $t = 0$. Choose an initial population, i.e. a simple sample of points from A distributed according to μ_0:

$$x = (x^1, \ldots, x^m) \in A^m.$$

1 Given t-th population $x = (x^1, \ldots, x^m) \in A^m$ generate independently k points $y^i \in A$ according to a distribution $\nu^{t_i} \in N$ each, $i = 1, \ldots, k$. Let $y = (y^1, \ldots, y^k) \in A^k$.

2 Apply $\varphi^{t_i} \in \Phi$ to x^i , $i = 1, \ldots, m$.

3 Sort the sequence $(\varphi^{t_i}(x^1), \ldots, \varphi^{t_m}(x^m), y^1, \ldots, y^k)$ using f as a criterion to get

$$(\bar{x}^1, \ldots, \bar{x}^{m+k}) \text{ with } f(\bar{x}^1) \leqslant \cdots \leqslant f(\bar{x}^{m+k}).$$

4 Form the next population with the first m points

$$\bar{x} = (\bar{x}^1, \ldots, \bar{x}^m)$$

5 Increase $t := t + 1$, let $x = \bar{x}$ and go to Step 1.

Let $\widehat{f} : A^m \longrightarrow \mathbb{R}$ be defined as $\widehat{f}(x) = f(x^1)$.
Let us note that $\widehat{A}^\star = A^\star \times A^{m-1}$ is the set of global minimums of $\widehat{f}$.

Let $\widehat{f} : A^m \to \mathbb{R}$ be defined as $\widehat{f}(x) = f(x^1)$.
Let us note that $\widehat{A}^\star = A^\star \times A^{m-1}$ is the set of global minimums of $\widehat{f}$.

Theorem (Ombach, Tarłowski)

Let $\{X_t : t = 1, 2, 3, \dots\}$ be the sequence generated by the Algorithm, where $X_0 = (X_0^1, \dots, X_0^m)$ is a random vector with distribution $(\mu_0)^m$. Let for each $t = 1, 2, 3, \dots$, $Y_t = (Y_t^1, \dots, Y_t^k)$ be independent random vectors, and independent of X_0, distributed according to $\nu^{t_1} \times \cdots \times \nu^{t_k}$ with $\nu^{t_i} \in N$. Assume that:

(z1) *for any $c \in \mathbb{R}$ and $\nu \in N$, $\nu(l_c) = 0$.*

(z2) *There exists t_0 such that for any $t \geqslant 1$ there is $0 \leqslant s \leqslant t_0$ and some $1 \leqslant j \leqslant k$ with $\nu^{(t+s)_j} \in N_0$.*

Then,

$$\mathrm{Prob}\,(X_t \longrightarrow \widehat{A}^\star, \ \text{as} \ \ t \longrightarrow \infty) = 1. \tag{1}$$

Simulated Annealing, SA

Simulated Annealing, SA

As so far – general scheme

Main step of the algorithm at the time t:
Given approximation x_t draw random point, say z, compute from them a candidate for a new approximation, say $y = Q(x_t, z)$, and put:

$$x_{t+1} = \begin{cases} x_t, & \text{if } f(y) \geqslant f(x_t) \\ y, & \text{if } f(y) < f(x_t), \end{cases}$$

Simulated Annealing, SA

As so far – general scheme

Main step of the algorithm at the time t:
Given approximation x_t draw random point, say z, compute from them a candidate for a new approximation, say $y = Q(x_t, z)$, and put:

$$x_{t+1} = \begin{cases} x_t, & \text{if } f(y) \geqslant f(x_t) \\ y, & \text{if } f(y) < f(x_t), \end{cases}$$

SA breaks the scheme

Simulated Annealing, SA

As so far – general scheme

Main step of the algorithm at the time t:
Given approximation x_t draw random point, say z, compute from them a candidate for a new approximation, say $y = Q(x_t, z)$, and put:

$$x_{t+1} = \begin{cases} x_t, & \text{if } f(y) \geqslant f(x_t) \\ y, & \text{if } f(y) < f(x_t), \end{cases}$$

SA breaks the scheme

From time to time it seems even better to have $x_{t+1} = y$, even if y is worse than x_t.

Simulated Annealing

$f : A \to \mathbb{R}$ is a continuous function and $A \subset \mathbb{R}^n$ is a compact set.

$B \subset \mathbb{R}^d$. $M > 0$ and $[0, M] \ni \beta_t$ satisfies $\lim_{t \to \infty} \beta_t = 0$.

Let μ_0 – Borel probability measure on A

Let ν – Borel probability measure on B

$Q \colon A \times B \to A$ – measurable operator.

Simulated Annealing

$f : A \to \mathbb{R}$ is a continuous function and $A \subset \mathbb{R}^n$ is a compact set.

$B \subset \mathbb{R}^d$. $M > 0$ and $[0, M] \ni \beta_t$ satisfies $\lim_{t\to\infty} \beta_t = 0$.

Let μ_0 – Borel probability measure on A

Let ν – Borel probability measure on B

$Q\colon A \times B \to A$ – measurable operator.

Algorithm

0. Set $t = 0$. generate a point x_0 from the distribution μ_0 on A.
1. Given x_t generate point $z \in B$ according to distribution ν.
2. If $f(Q(x_t, z)) \leqslant f(x_t)$, then $x_{t+1} = f(Q(x_t, z))$.
3. If $f(Q(x_t, z)) > f(x_t)$, then generate point $r \in (0, 1)$ according to the uniform distribution. If

$$r \leqslant \exp\left(-\frac{f(Q(x_t, z)) - f(x_t)}{\beta_t}\right),$$

$x_{t+1} = f(Q(x_t, z))$.

4. Increase $t := t + 1$ and go to step 2.

The essence of the Algorithm is to create an opportunity to substitute the current approximation with the next approximation even if the new one is worse, the chance of such an action decreases with time, but can be zoomed, where the approximation is only slightly better than the new one.

The essence of the Algorithm is to create an opportunity to substitute the current approximation with the next approximation even if the new one is worse, the chance of such an action decreases with time, but can be zoomed, where the approximation is only slightly better than the new one.

Let X_t be random vectors which realizations are generated by SA.

Theorem (Tarłowski, 2014)

Assume that for all $x \in A$, $\nu(D_{f\circ Q}(x)) = 0$, where $D_{f\circ Q}(x)$ consists of $z \in B$ such, that $f \circ Q$ is not continuous at point (x, z). Assume also, that for all $x \in A \setminus A^*$,

$$\nu(\{z \in B\colon f(Q(x,z)) < f(x)\}) > 0. \tag{2}$$

Then,

$$\forall \varepsilon > 0 \ Prob(dist(X_t, A^*) < \varepsilon) \overset{t\to\infty}{\longrightarrow} 1 \quad and \quad E(f(X_t)) \overset{t\to\infty}{\longrightarrow} \min_A f.$$

Markov Chain Monte Carlo, MCMC

Markov Chain Monte Carlo, MCMC

A – finite set – a big one $\quad f : A \to \mathbb{R}$.
We are looking for $\arg\max f$ on A (tradition!)

Markov Chain Monte Carlo, MCMC

A – finite set – a big one $\quad f : A \to \mathbb{R}$.
We are looking for $\arg\max f$ on A (tradition!)

Instead of drawing points from the uniform distribution A it would be better to do it from a distribution π, such that:

$$\pi(x) \geqslant \pi(y) \iff f(x) \geqslant f(y).$$

Example: $\beta > 0$

$$\pi_\beta(x) = \frac{\exp(\beta f(x))}{C}, \quad C = \sum_{y \in A} \exp(\beta f(y))$$

Markov Chain Monte Carlo, MCMC

A – finite set – a big one $\quad f : A \to \mathbb{R}$.
We are looking for $\arg\max f$ on A (tradition!)

Instead of drawing points from the uniform distribution A it would be better to do it from a distribution π, such that:

$$\pi(x) \geqslant \pi(y) \iff f(x) \geqslant f(y).$$

Example: $\beta > 0$

$$\pi_\beta(x) = \frac{\exp(\beta f(x))}{C}, \quad C = \sum_{y \in A} \exp(\beta f(y))$$

Basic rule:
Form a sample from π and find a maximum of f on it, say x_0^π.
Point x_0^π, is 'statistically close' to $\{a \in A : \forall x \in A\ f(x) \leqslant f(a)\}$.

Markov Chain Monte Carlo, MCMC

A – finite set – a big one $\quad f : A \to \mathbb{R}$.
We are looking for $\arg\max f$ on A (tradition!)

Instead of drawing points from the uniform distribution A it would be better to do it from a distribution π, such that:

$$\pi(x) \geqslant \pi(y) \iff f(x) \geqslant f(y).$$

Example: $\beta > 0$

$$\pi_\beta(x) = \frac{\exp(\beta f(x))}{C}, \quad C = \sum_{y \in A} \exp(\beta f(y))$$

Basic rule:
Form a sample from π and find a maximum of f on it, say x_0^π.
Point x_0^π, is 'statistically close' to $\{a \in A : \forall x \in A\ f(x) \leqslant f(a)\}$.

How to draw points from π?

MCMC

MCMC main idea

Define on A a specific ergodic Markov Chain $\mathbf{P}$ having π as the (uique) stationary state.
Now, choosing an arbitrarily initial point $x_0 \in A$ generate points $x_1, x_2, \ldots$ according to $\mathbf{P}$.

MCMC

MCMC main idea

Define on A a specific ergodic Markov Chain $\mathbf{P}$ having π as the (uique) stationary state.
Now, choosing an arbitrarily initial point $x_0 \in A$ generate points $x_1, x_2, \ldots$ according to $\mathbf{P}$.

It is granted that for t large enough x_t are actually drawn from π.

MCMC

MCMC main idea

Define on A a specific ergodic Markov Chain $\mathbf{P}$ having π as the (uique) stationary state.
Now, choosing an arbitrarily initial point $x_0 \in A$ generate points $x_1, x_2, \ldots$ according to $\mathbf{P}$.

It is granted that for t large enough x_t are actually drawn from π.

How to built the Markov Chain $\mathbf{P}$?

MCMC

MCMC main idea

Define on A a specific ergodic Markov Chain $\mathbf{P}$ having π as the (uique) stationary state.
Now, choosing an arbitrarily initial point $x_0 \in A$ generate points $x_1, x_2, \ldots$ according to $\mathbf{P}$.

It is granted that for t large enough x_t are actually drawn from π.

How to built the Markov Chain $\mathbf{P}$?

Metropolis Algorithm

Metropolis Algorithm for Rucksack problem

Maximize value

$$f(x) = c^T x$$

under constraints

$$w^T x \leqslant W, \quad x^i \in \{0, 1\},$$

$c = (c^1, \ldots, c^k)$ vector of prices,
$w = (w^1, \ldots, w^k)$ vector of weights,
W – maximum weight allowed.

Metropolis Algorithm for Rucksack problem

Maximize value

$$f(x) = c^T x$$

under constraints

$$w^T x \leqslant W, \quad x^i \in \{0, 1\},$$

$c = (c^1, \ldots, c^k)$ vector of prices,
$w = (w^1, \ldots, w^k)$ vector of weights,
W – maximum weight allowed.

Here

$$A = \{x \in \{0, 1\}^k : w^T x \leqslant W\}$$

Metropolis Algorithm for Rucksack problem

Maximize value

$$f(x) = c^T x$$

under constraints

$$w^T x \leqslant W, \quad x^i \in \{0, 1\},$$

$c = (c^1, \ldots, c^k)$ vector of prices,
$w = (w^1, \ldots, w^k)$ vector of weights,
W – maximum weight allowed.

Here

$$A = \{x \in \{0, 1\}^k : w^T x \leqslant W\}$$

Example: $W = 20$,
[proce,weight] = [50,10], [20,5], [20,4], [10,1], [5,3], [5,5], [4,4], [3,3], [3,3], [3,3], [2,2], [3,1], [3,1], [2,1], [2,1], [2,1], [2,1], [1,1], [1,1], [1,1]
The solution:

$$x^* = (1, 1, 1, 1, 0, 0, \ldots, 0), \quad f(x^*) = 100.$$

Metropolis Algorithm for Rucksack problem

Fix $\beta > 0$ and $T > 0$.
Choose initial $x_0 \in A$. Substitute $x \leftarrow x_1$
For t $= 0$ to $T - 1$ do:

1. Generate J uniformly from $1, \ldots, k$.
2. Let $y = (x^1, \ldots, x^{J-1}, 1 - x^J, x^{J+1}, \ldots, x^k)$.
3. If $y \notin A$, then $x_{t+1} \leftarrow x_t$.
   Otherwise:
4. Let $\alpha = \begin{cases} \exp(-\beta\, c^J), & \text{when} \quad x^J = 1 \\ 1 & \text{when} \quad x^J = 0 \end{cases}$.
5. Generate $u \in \{0, 1\}$ according to distribution:

$$\text{Prob}(0) = 1 - \alpha, \quad \text{Prob}(1) = \alpha.$$

If $u = 0$, then $x_{t+1} \leftarrow x_t$.
If $u = 1$, then $x_{t+1} \leftarrow y$.

For large T, X_T have distribution π_β.

Metropolis Algorithm for Rucksack problem

Modifications

Metropolis Algorithm for Rucksack problem

Modifications

Admit slow increase of β.
For example:
$\beta(t) = \log t$,$\beta(t) = 1.0001^t$,
$\beta(t) = b_0 \left(1 + \frac{1}{T}\right)^{at}$, where T is the number of steps, b_0 and a are positive.

Metropolis Algorithm for Rucksack problem

Modifications

Admit slow increase of β.
For example:
$\beta(t) = \log t$,$\beta(t) = 1.0001^t$,
$\beta(t) = b_0 \left(1 + \frac{1}{T}\right)^{at}$, where T is the number of steps, b_0 and a are positive.

Item J is generated according to the distribution: $\text{Prob}(J = i) = \frac{c^i}{\sum_{s=1}^{k} c^s}$.

Example: $T = 100$, $\beta = \beta(t) = 0.5(1 + \frac{1}{T})^t$, $x_1 = (0, \ldots, 0)$.

$$\begin{bmatrix}
[0,0,1,1,0,0,0,1,1,1,0,0,0,0,1,1,0,0,1,0] & [44,17] \\
[0,0,1,1,0,0,0,1,1,1,0,0,0,0,1,1,0,0,1,0] & [44,17] \\
[0,0,1,1,0,0,0,1,1,1,0,0,1,0,1,1,0,0,1,0] & [47,18] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0] & [50,19] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0] & [50,19] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0] & [50,19] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,1] & [51,20] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,1] & [51,20] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,1] & [51,20] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,0,0,1,0] & [50,19] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,1,0,1,0] & [52,20] \\
[0,0,1,1,0,0,0,1,1,1,0,1,1,0,1,1,1,0,1,0] & [52,20]
\end{bmatrix}$$

Some other stochastic algorithms

Evolutionary Algorithms (EA) including
Genetic Algorithms (GA),
Particle Swarm Optimization (PSO),
Ant Colony Optimization (ACO),
Artificial Bee Colony (ABC),
Grenade Explosion Method (GEM),
Covariance Matrix Adaptation (CMA),
Markov Chain Monte Carlo (MCMC),
Differential Evolution (DE),
and more.

Assessing the quality

The concept of good or pure performance is not always clear but depends on the specific situation in which the algorithm is used.

Assessing the quality

The concept of good or pure performance is not always clear but depends on the specific situation in which the algorithm is used.

In online optimization we would prefer short time criterion than accuracy.

Assessing the quality

The concept of good or pure performance is not always clear but depends on the specific situation in which the algorithm is used.

In online optimization we would prefer short time criterion than accuracy.

For example, in

> *Bialy, J., Ciecko, A., Cwiklak, J., Grzegorzewski, M., Koscielniak, P., Ombach, J., Oszczak, S., Ąircraft Landing System Utilizing a GPS Receiver with Position Prediction Functionality,"Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 457-467.*

a problem of fast short time interval prediction during aircraft landing is discussed, when optimization process has to be as quick as possible.

Assessing the quality

The concept of good or pure performance is not always clear but depends on the specific situation in which the algorithm is used.

In online optimization we would prefer short time criterion than accuracy.

For example, in

> *Bialy, J., Ciecko, A., Cwiklak, J., Grzegorzewski, M., Koscielniak, P., Ombach, J., Oszczak, S., Ąircraft Landing System Utilizing a GPS Receiver with Position Prediction Functionality,"Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 457-467.*

a problem of fast short time interval prediction during aircraft landing is discussed, when optimization process has to be as quick as possible.

In design optimization, creating long-term schedules or data mining, when optimization processes would usually be carried out only once in a long time, the accuracy and certainty of the result is then crucial.

Assessing the quality

According to:

> *K.P. Bennett, E. Parrado-Hern, The Interplay of Optimization and Machine Learning Research, Journal of Machine Learning Research 7(2006) pp. 1265 – 1281.*

desirable properties of an optimization algorithm from the Machine Learning perspective are:

Assessing the quality

According to:

> *K.P. Bennett, E. Parrado-Hern, The Interplay of Optimization and Machine Learning Research, Journal of Machine Learning Research 7(2006) pp. 1265 – 1281.*

desirable properties of an optimization algorithm from the Machine Learning perspective are:

good generalization,
scalability to large problems,
good performance in practice in terms of execution times and memory requirements,
simple and easy implementation of algorithm,
exploitation of problem structure fast convergence to an approximate solution of model,
robustness and numerical stability for class of machine learning models attempted,
theoretically known convergence and complexity.

Assessing the quality

Experimental approach

A common practise is to run the algorithms on some already known test (benchmark) functions or on collections of such functions known as suites or testbeds and compare the results.

Assessing the quality

Experimental approach

A common practise is to run the algorithms on some already known test (benchmark) functions or on collections of such functions known as suites or testbeds and compare the results.

Empirical and experimental approaches to comparing algorithms have many disadvantages, especially when the algorithms are designed to be robust, general purpose optimization tools.

Assessing the quality

Experimental approach

A common practise is to run the algorithms on some already known test (benchmark) functions or on collections of such functions known as suites or testbeds and compare the results.

Empirical and experimental approaches to comparing algorithms have many disadvantages, especially when the algorithms are designed to be robust, general purpose optimization tools.

One obvious danger with empirically evaluating algorithms is that the resulting conclusions depend as much on what problems are used for testing the algorithms that are being compared. This can have the side effect that algorithms are designed and tuned to perform well on a particular test suite; the resulting specialization may or may not translate into improved performance on other problems or applications.

Assessing the quality

COCO (COmparing Continuous Optimisers) is a platform for systematic and sound comparisons of real-parameter global optimisers. COCO provides benchmark function testbeds and tools for processing and visualizing data generated by one or several optimizers. The COCO platform has been used for the Black-Box-Optimization-Benchmarking (BBOB) workshops that took place during the GECCO conference in 2009, 2010, 2012, and 2013. The next edition is going to take place as a special session in May 2015 during the next IEEE Congress on Evolutionary Computation (CEC'2015) in Sendai, Japan. The COCO source code is available at the downloads page at http://coco.gforge.inria.fr.

Convergence of Markov type algorithms

1. J. Ombach, A Proof of Convergence of General Stochastic Search for Global Minimum, Journal of Difference Equations and Applications 13 (2007), pp. 795 - 802.
2. M. Radwański, Convergence of nonautonomous evolutionary algorithm, Universitatis Iagellonicae Acta Mathematica, 45, (2007), pp. 197 - 206.
3. J. Ombach, Stability of evolutionary algorithms, Journal Math Anal Appl. 342(2008), pp. 326 - 333.
4. D. Tarłowski, Sufficient conditions for the convergence of non-autonomous stochastic search for a global minimum, UIAM (2011), 73-83
5. Bialy, J., Ciecko, A., Cwiklak, J., Grzegorzewski, M., Koscielniak, P., Ombach, J., Oszczak, S., Ąircraft Landing System Utilizing a GPS Receiver with Position Prediction Functionality,"Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 457-467.
6. J. Ombach, D. Tarłowski, Nonautonomous Stochastic Search in Global Optimization, Journal in Nonlinear Sci., vol. 22 (2012), 169 - 185.
7. D. Tarłowski, Nonautonomous stochastic search for global minimum in continuous optimization, Journal Math Anal Appl. 412(2014), 631-645
8. D. Tarłowski, Nonautonomous Dynamical Systems in Stochastic Global Optimization, Ph.D. thesis, Department of Mathematics, Iagiellonian University 2014.
9. J. Ombach, D. Tarłowski, Stochastyczne algorytmy optymalizacji z perspektywy ukĹ‚adów dynamicznych, in Polish, preprint.

Convergence of Markov type algorithms

General stochastic algorithm

$$X_t = T_t(X_{t-1}, Y_t), \text{ for } t = 1, 2, 3 \ldots. \tag{3}$$

Here X_t, for $t \geqslant 0$ denote random variables corresponding to successive outcomes of the algorithm and Y_t are vectors responsible for randomness in steps $1, 2, 3, \ldots$.
T_t define the mechanism of the algorithm itself.

Convergence of Markov type algorithms

General stochastic algorithm

$$X_t = T_t(X_{t-1}, Y_t), \text{ for } t = 1, 2, 3 \ldots. \tag{3}$$

Here X_t, for $t \geqslant 0$ denote random variables corresponding to successive outcomes of the algorithm and Y_t are vectors responsible for randomness in steps $1, 2, 3, \ldots$.
T_t define the mechanism of the algorithm itself.

Assumptions:
$A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^d$,
$T_t : A \times B \to A$, for $t = 1, 2, 3 \ldots$ measurable operators,
$(\Omega, \Sigma, \mathrm{Prob})$ – probability space ,
$X_0 : \Omega \to A$ – distributed according to some measure μ_0,
$Y_t : \Omega \to B$ – random vectors distributed according to some measures ν_t.
We assume that $X_0, Y_1, Y_2, Y_3, \ldots$ are independent.

PRS

$A = B = [0,1]^n$, μ_0, ν_t for all t – the Lebesgue measure and:

$$T_t(x,y) = \begin{cases} x, & \text{if } f(y) \geqslant f(x) \\ y, & \text{if } f(y) < f(x), \end{cases}$$

Algorithm

0. Set $t = 0$. Generate a point x_0 from the uniform distribution on A.
1. Given x_t, generate y_t from the uniform distribution on A.
2. If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$.
3. Increase $t := t + 1$ and go to Step 1.

Hybrid Algorithm

$\nu_t = \nu$

$$T_t(x, y) = \begin{cases} x, & \text{if } f(\varphi(y)) \geqslant f(x) \\ \varphi(y), & \text{if } f(\varphi(y)) < f(x), \end{cases}$$

Algorithm

0 Set $t = 0$. Generate a point x_0 from the distribution μ_0 on A.

1 Given point x_t, generate $y_t \in A$ according to the distribution ν.

2 Apply φ to y_t.

3 If $f(\varphi(y_t)) < f(x_t)$, then $x_{t+1} = \varphi(y_t)$.

3 Increase $t := t + 1$ and go to Step 1.

Multistart Algorithm

$T_t : A^m \times A^k \to A^m$ as: $T_t(x, y) = \bar{x}$,
Instead of measures μ_0 and ν_t we use the product measures μ_0^m and ν^k respectively.

Algorithm

0. Let $t = 0$. Choose an initial population, from A distributed according to μ_0: $x = (x^1, \ldots, x^m) \in A^m$.
1. Given t-th population $x = (x^1, \ldots, x^m) \in A^m$ generate independently k points $y^i \in A$ according to a distribution $\nu^{t_i} \in N$ each, $i = 1, \ldots, k$. Let $y = (y^1, \ldots, y^k) \in A^k$.
2. Apply $\varphi^{t_i} \in \Phi$ to x^i , $i = 1, \ldots, m$.
3. Sort the sequence $(\varphi^{t_i}(x^1), \ldots, \varphi^{t_m}(x^m), y^1, \ldots, y^k)$ using f as a criterion to get $(\bar{x}^1, \ldots, \bar{x}^{m+k})$ with $f(\bar{x}^1) \leqslant \cdots \leqslant f(\bar{x}^{m+k})$.
4. Form the next population with the first m points

$$\bar{x} = (\bar{x}^1, \ldots, \bar{x}^m).$$

5. Increase $t := t + 1$, let $x = \bar{x}$ and go to Step 1.

SA

$$T_t(x,z,r) = \begin{cases} Q(x,z), & \text{if } f(Q(x,z)) \leqslant f(x), \\ Q(x,z), & \text{if } f(Q(x,z)) > f(x) \wedge r \leqslant \exp\left(-\frac{f(Q(x_t,z))-f(x_t)}{\beta_t}\right), \\ x, & \text{otherwise.} \end{cases}$$

$T_t : A \times (B \times [0,1]) \to A$.
$\nu_t = \nu \times \lambda$ – the product measure, where λ is the Lebesgue measure on the the unit interval.

Algorithm

0. Set $t = 0$. enerate a point x_0 from the distribution μ_0 on A.
1. Given x_t generate point $z \in B$ according to distribution ν.
2. If $f(Q(x_t,z)) \leqslant f(x_t)$, then $x_{t+1} = f(Q(x_t,z))$.
3. If $f(Q(x_t,z)) > f(x_t)$, then generate point $r \in (0,1)$ according to the uniform distribution. If

$$r \leqslant \exp\left(-\frac{f(Q(x_t,z)) - f(x_t)}{\beta_t}\right),$$

$x_{t+1} = f(Q(x_t,z))$.

Convergence of Markov type algorithms

Some other examples of General Algorithm (Tarłowski, 2014)

$$X_t = T_t(X_{t-1}, Y_t), \text{ for } t = 1, 2, 3 \ldots. \tag{4}$$

ARS
GEM
PSO
$ES(\mu/\varrho + \lambda)$

Convergence of Markov type algorithms

Assumptions:
$A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^d$,
$T_t : A \times B \to A$, for $t = 1, 2, 3 \ldots$ measurable operators,
$(\Omega, \Sigma, \mathrm{Prob})$ – probability space ,
$X_0 : \Omega \to A$ – distributed according to some measure μ_0,
$Y_t : \Omega \to B$ – random vectors distributed according to some measures ν_t.
We assume that $X_0, Y_1, Y_2, Y_3, \ldots$ are independent.

Convergence of Markov type algorithms

Assumptions:
$A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^d$,
$T_t : A \times B \to A$, for $t = 1, 2, 3 \ldots$ measurable operators,
$(\Omega, \Sigma, \mathrm{Prob})$ – probability space ,
$X_0 : \Omega \to A$ – distributed according to some measure μ_0,
$Y_t : \Omega \to B$ – random vectors distributed according to some measures ν_t.
We assume that $X_0, Y_1, Y_2, Y_3, \ldots$ are independent.

$M(A)$ – the spaces of all probability Borel measures on A.
$M(B)$ – the spaces of all probability Borel measures on B.
$\mathcal{T}$ – the space of the all measurable operators $T : A \times B \to A$ equipped with the topology of uniform convergence.

Convergence of Markov type algorithms

Theorem (Ombach, Tarłowski, 2012)

Assume that A is a compact set and $f : A \to \mathbb{R}$ is continuous. Let $U \subset \mathcal{T} \times M(B)$ be a compact set. Assume that for any $u = (T, \nu) \in U$:

(A) *For any $x_0 \in A$, there is a Borel set $D_T(x_0) \subset B$ with $\nu(D_T(x_0)) = 0$, such that T is continuous in (x_0, y), for any $y \notin D_T(x_0)$.*

(B) *For any $x \in A^\star$ and $y \in B$, $T(x, y) \in A^\star$.*

(C1) *For any $x \in A \setminus A^\star$:*

$$\int_B f(T(x,y))\,\nu(dy) \leqslant f(x). \tag{5}$$

(C2) *There is a closed set $U_0 \subset U$ such that for any $(T, \nu) \in U_0$ and $x \in A \setminus A^\star$:*

$$\int_B f(T(x,y))\,\nu(dy) < f(x). \tag{6}$$

Convergence of Markov type algorithms

Let $\{u_t = (T_t, \nu_t) : t \geqslant 1\} \subset U$ satisfy the following:

(U0) There is $t_0 \geqslant 1$ such that for any $t \geqslant 1$ there is $s \leqslant t_0$ with $u_{t+s} \in U_0$.

Then, for every $\varepsilon > 0$:

$$\lim_{t \to \infty} \mathrm{Prob}\left(\mathrm{dist}(X_t, A^\star) < \varepsilon\right) = 1. \tag{7}$$

Convergence of Markov type algorithms

Let $\{u_t = (T_t, \nu_t) : t \geqslant 1\} \subset U$ satisfy the following:

(U0) There is $t_0 \geqslant 1$ such that for any $t \geqslant 1$ there is $s \leqslant t_0$ with $u_{t+s} \in U_0$.

Then, for every $\varepsilon > 0$:

$$\lim_{t \to \infty} \mathrm{Prob}\left(\mathrm{dist}(X_t, A^\star) < \varepsilon\right) = 1. \tag{7}$$

Assume additionally

(D) For any $t \geqslant 1$, $x \in A$ and $y \in B$: $f(T_t(x, y)) \leqslant f(x)$.

Then,

$$\mathrm{Prob}\left(X_t \longrightarrow A^\star, \text{ as } t \longrightarrow \infty\right) = 1. \tag{8}$$

Stochastic algorithms in R

- GenSA - Generalized Simulated Annealing
- DEoptim - Differential Evolutionary Optimization
- soma - Self-Organising Migrating Algorithm
- rgenoud - GENetic Optimization Using Derivatives
- cmaes - Covariance Matrix Adapting Evolutionary Strategy
- mco - Multi Criteria Optimisation
- mcga - Machine Coded Genetic Algorithm
- emoa - Evolutionary Multiobjective Optimisation Algorithms
- soobench - Single Objective Optimization Benchmark Functions

T H A N K Y O U

THANK YOU

DZIĘKUJĘ