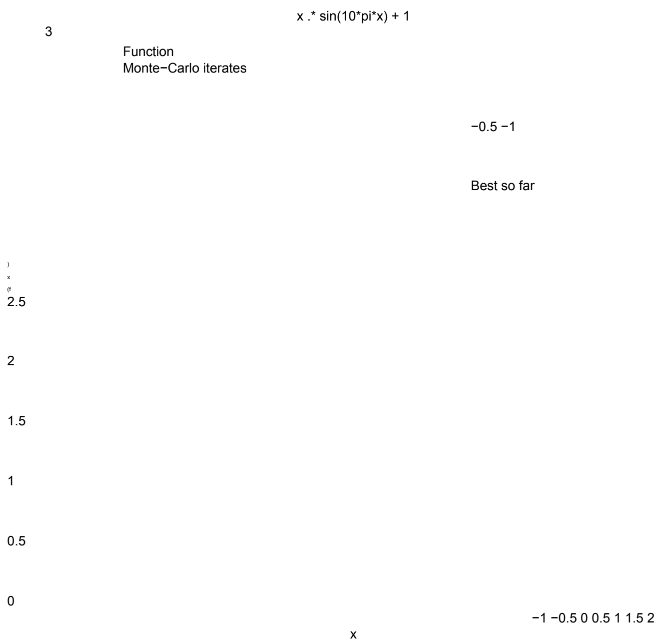
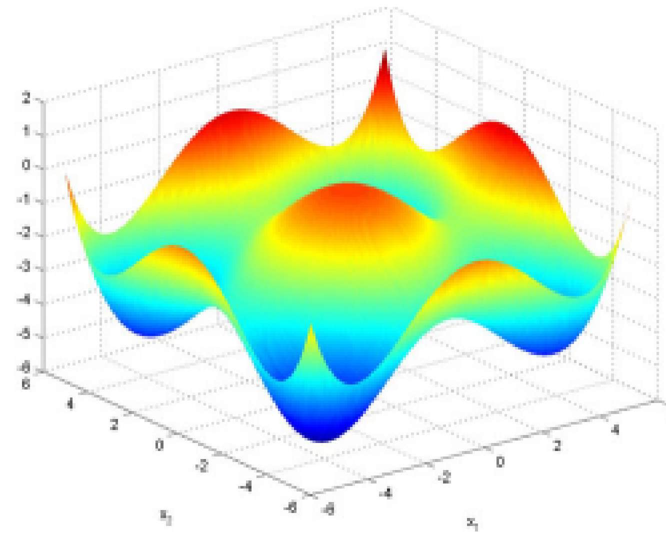
Local and Global Optimization

Formulation, Methods and Applications





Rob Womersley
R.Womersley@unsw.edu.au
http://www.maths.unsw.edu.au/~rsw

School of Mathematics
&
Statistic

Contents

- Optimization Problems
  - Variables
  - Objective functions
  - Constraints
  - Discrete vs Continuous
  - Available information
  - Optimality
  - Problem size
- Local Methods
  - Steepest Descent
  - Newton
  - Quasi-Newton
  - Conjugate Gradient

- Simplex

- Local vs Global minima
  - Continuous examples
  - Travelling Salesman Problem (TSP)
  - Minimum Energy Problems
- Exact methods
  - Enumeration
  - Branch and Bound
  - Interval Methods
- Monte-Carlo Methods
  - Random points
  - Random starting points
  - Quasi-Monte Carlo methods
  - Sparse Grids

- Simulated Annealing
  - Accept larger functions values with certain probability
  - Annealing schedule

- Evolutionary (Genetic) Algorithms
  - Population of individuals (variables)
  - Survival depends on fitness of individual

. New individuals from genetic operators: crossover, mutation Rob

Optimization

- Find values of the variable(s) x to give the
  - . best (minimum or maximum) of an
  - . objective function $f(x)$
  - . subject to any constraints (restrictions) $c(x) = 0$, $c(x) \leq 0$ on

what values the variables are allowed to take. • Calculus, $x \in$

$\mathbb{R} \Longrightarrow f^{0}(x) = 0$ (stationary point) . $f^{00}(x) > 0 \Longrightarrow$ min, $f^{00}(x) < 0$

=⇒ max. Local vs Global

1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
6 1
5 3 2.5
4 2
3 1.5
1
0.5
0
2 0

Examples of optimization problems

- Choosing a course

. Variables: which courses are available (continuous vs discrete) . Objective: compulsory, interesting (many, hard to quantify) . Constraints: one place at a time, pre-requisites

- How much should you invest in the bank, shares, property, ... Variables: fraction of money in each asset (many variables) . Objective: maximize return, minimize risk (several competing objectives) . Constraints: money available, non-negative amounts, fractions in [0, 1]

- Optimality principles: Some form of optimality underlies many problems in .
  Science: physics, chemistry, biology, ...
  . Commerce, economics, management, ...
  . Engineering, Architecture, ...

Finite dimensional optimization – variables

- Variables $x \in \mathbb{R}^n$: $x = (x_1, x_2, \ldots, x_n)^T$, $x_i \in \mathbb{R}$, $i = 1, \ldots, n$ •

n = number of variables.

- $n = 1$, univariate; $n \geq 2$, multivariate

- Ex 1 – What fraction of a portfolio should be invested in each asset class?
  - . n = number of assets;
  - . x_i = fraction invested in asset class i for $i = 1, . . . , n$.
- Ex 2 – In which order should a number of destinations be visited?
  - . n = number of destinations to be visited
  - . x = permutation of $\{1, . . . , n\}$
  - . X_{ij} = 1 if you go from destination i to j; 0 otherwise
- Ex 3 – What are the positions of atoms/molecules in a stable compound?
  - . m = number of atoms; $n = 3m$ for positions (x, y, z) in space . $x = [x_1, y_1, z_1, x_2, y_2, z_2, . . . , x_m, y_m, z_m]$.

Objective functions

- Objective function: Minimize $f(x)$
- Mathematical representation of “best”

- Maximize $\bar{f}(x) \Longleftrightarrow$ Minimize $-\bar{f}(x)$
- Ex 1 – Maximize returns; minimize risk
  - Maximize return $r^T x = \sum_{i=1}^{n} r_i x_i$; r_i = return on asset i . Minimize risk $x^T C x = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i C_{ij} x_j$; Covariance matrix C
- Ex 2 – Minimize cost of visiting all destinations
  - Total cost = $\sum_{i=1}^{n} \sum_{j=1}^{m} X_{ij} C_{ij}$; C_{ij} = cost of going from i to j
- Ex 3 – Minimize the energy of the system
  - Distances between particles $\mathbf{x}_j = [x_j, y_j, z_j]$ for $j = 1, \ldots, m$.

    Energy = $\sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} \varphi(|\mathbf{x}_i - \mathbf{x}_j|)$;
  - Potential $\varphi(r)$, Coulomb $1/r$, Leonard-Jones,

Constraints

- Constraints: $x \in \Omega \subseteq \mathbb{R}^n$, Feasible region Ω

. Simple bounds: $l \le x \le u \Longleftrightarrow l_i \le x_i \le u_i,\ i = 1, \ldots, n$. Linear constraints

$$Ax = b \text{ Equality constraints}$$
$$Ax \le b \text{ Inequality constraints}$$

$$Ax = b \Longleftrightarrow \sum_{j=1}^{n} a_{ij}x_j = b_i,\ i = 1, \ldots, m$$

. Nonlinear constraints

$$c_i(x) = 0,\ i = 1, \ldots, m_e \text{ Equality constraints}$$
$$c_i(x) \le 0,\ i = m_e + 1, \ldots, m \text{ Inequality constraints}$$

. Integrality constraints $x_i \in \mathbb{Z}$

$$x_i \in \{0, 1\} \text{ Zero-one variables}$$
$$x_i \in \{0, 1, 2, \ldots\} \text{ Nonnegative integer variables.}$$

. $\bar{c}_i(x) \ge 0 \Longleftrightarrow -\bar{c}_i(x) \le 0$

Constraints – Examples

- Ex 1 – fraction of portfolio
  - fraction $0 \le x_i \le 1$ for $i = 1, \ldots, n$
  - fully invested $\sum_{i=1}^{n} x_i = 1$
  - investment guidelines $x_1 + x_2 + x_3 \le 0.6$
  - minimum return $r^T x \ge 0.1$
  - maximum risk $x^T C x \le 0.4$
- Ex 2 – visit all destinations exactly once
  - Go somewhere $\sum_{j=1}^{n} X_{ij} = 1$ for all i
  - Come from somewhere $\sum_{i=1}^{n} X_{ij} = 1$ for all j
  - $X_{ij} \in \{0, 1\}$
- Atoms/electrons/molecules

. Particles on a surface M: $x_j \in M$
. Bonds between particles
. Geometry

Dynamic constraints: Optimal Control

- Variables: a function $x(t)$

. Time scales: $t \in [0, T]$
. Space of functions (continuous, differentiable, ...)

- Objective is a function of x

. Final state $x(T)$

. $\int_0^T |x''(t)|\, dt$

- Constraints: Differential equations plus algebraic equations .

Differential equations governing evolution of a system over time . Initial conditions $x(0)$ at $t = 0$, current state

- Bounds: $a \le x(t) \le b$ for $t \in [0, T]$ – infinite number of constraints. Rob

Optimization problem classes (Terminology)

Optimization Technology Centre http://www.ece.northwestern.edu/OTC .

Combinatorial problems – finite but typically very large set of solutions

- Unconstrained problems – no constraints, any variables $x \in R^n$ are allowed

- Linearly constrained problems – only linear constraints (simple bounds and/or general linear constraints)

. Nonlinearly constrained problems – at least one constraint is nonlinear . Linear programming – objective and all constraints are linear, continuous variables . Nonlinear programming – nonlinear objective or constraints, continuous variables . Integer programming – variables are restricted to be integers

. Mixed integer programming – some variables are integers, some are continuous . Stochastic optimization – some of the problem data is not deterministic

Available information

- Objective function $f(x)$, $x \in \mathrm{R}^n$
- Objective gradient: n-vector,

$$\nabla f(x) =$$

matrix $\partial f(x)\ \partial x_n$

$\partial f(x)\ \partial x_1$

- Objective ..
.
Hessian: *n* by *n*

$\partial x^2_1 \cdot\ \cdot\ \cdot \partial^2 f(x)\ \partial^2 f(x)$

∂x^2_n

$\nabla^2 f(x) =$ $\partial x_1 x_n$

.. .. ..
. . .

$\partial^2 f(x)$ $\partial x_n x_1 \cdot\ \cdot$

$\cdot \partial^2 f(x)$

- Calculation: hand, symbolic (Maple, Mathematica), numerical (finite differ ence), automatic differentiation [6, 1]

- Example:

$f(x) = 0.01X^3$

$i=1$

$(x_i + 0.5)^4 - 30x^2{}_i - 20x_i$

Find the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ at $x = [0, 0, 0]^T$. Rob Womersley –

Optimality – Unconstrained

- x^*, $f(x^*)$ local minimum $\Longrightarrow$ $\nabla f(x^*) = 0$ (stationary point) .
  Can be minimum, maximum or saddle point
- Hessian information determines nature of stationary point: . Hessian positive definite (eigenvalues: all > 0) $\Longrightarrow$ local minimum . Hessian negative definite (eigenvalues: all < 0) $\Longrightarrow$ local maximum . Hessian indefinite (eigenvalues: some > 0, some < 0) $\Longrightarrow$ saddle point

Local minimum: Eigenvalues 1.3820, 3.6180

30

0

25

0

−5

20

−30
2

−10

0
0
Local maximum: Eigenvalues −3.6180, −1.3820

15

40

30

20

10

0

−15

10

2

−10

−20
2

2
1

2
1

1

−20

5

1

1
0

0
0
Saddle point: Eigenvalues −1.5414, 4.5414

−25

1
0

2 −1 −2 −1 −1 −2 −1 −1 −2 −1
x_2−2 −2
x_1 x_1 x_2 x_2−2 x_1

Problem Size

- Number of variables n, $x \in R^n$
- Limitations
  - Compute time
  - Memory
- Example: If a method takes $n^3+O(n^2)$ flops (floating point operations), what is the largest problem that can be solved in 24 hours on a 3 GHZ quad

core workstation?

Ans: $n \approx 10^5$

- Example: What is the largest Hessian (n by n symmetric matrix) that can be stored in IEEE double precision in 32 bit Windows (maximum 2 Gb addressable block)?

Ans: $n \approx 16,000$

Local search methods

- Line search methods:

. Given $x^{(1)}$ initial guess

. At $x^{(k)}$ generate search direction $d^{(k)}$

. Exact or approximate line search: $\alpha^{(k)} = \mathrm{argmin}_{\alpha\geq 0}\, f x^{(k)} + \alpha d^{(k)}$. New points $x^{(k+1)} = x^{(k)} + \alpha d^{(k)}$

. Descent method: $fx^{(k+1)} < f\, x^{(k)}$

- Steepest descent

. $d^{(k)} = -\nabla f(x^{(k)})$

. Global convergence: $x^{(k)} \longrightarrow x^*$, x^*stationary point ($\nabla f(x^*) = 0$) from any starting point

. Arbitrarily slow linear rate of convergence $|x^{(k+1)} - x^*| \approx \beta |x^{(k)} - x^*|$, $0 < \beta < 1$.

. Requires gradient $\nabla f(x)$, and $O(n)$ storage and work per iteration Rob

- Newton's method

. Solve linear system $\nabla^2 f(x^{(k)}) d^{(k)} = -\nabla f(x^{(k)})$ for search direction $d^{(k)}$. Only locally convergent

- Quadratic rate $|x^{(k+1)} - x^*| \approx |x^{(k)} - x^*|^2$ if $x^{(1)}$ sufficiently close to “nice” solution.
- Requires $\nabla^2 f(x)$, $O(n^2)$ storage, $O(n^3)$ work per iteration

• Quasi-Newton methods

- Solve $B^{(k)} d^{(k)} = -\nabla f(x^{(k)})$ for search direction $d^{(k)}$
- Update $B^{(k+1)} \approx \nabla^2 f(x^{(k)})$
- Superlinear rate $|x^{(k+1)} - x^*| \approx |x^{(k)} - x^*|^\tau$, $1 < \tau < 2$ under conditions
- Requires $\nabla f(x)$, $O(n^2)$ storage, $O(n^2)$ work per iteration

• Conjugate Gradients methods

- $d^{(k)} = -\nabla f(x^{(k)}) + \beta^{(k)} d^{(k-1)}$
- Update $\beta^{(k)}$
- Quadratic termination (conjugate directions)
- Requires $\nabla f(x)$, $O(n)$ storage and work per iteration

Local vs global optimization

- Local minimum $f^* = f(x^*)$, local minimizer x^*
  - smallest function value in some feasible neighbourhood
  - $x^* \in \Omega$
    - there exists a $\delta > 0$ such that $f^* \le f(x)$ for all x in $\{ x \in \Omega : |x - x^*| \le \delta \}$

- Global minimum $f^* = f(x^*)$, global minimizer x^*
  - smallest function value over all feasible points
  - $f^* \le f(x)$ for all x in Ω

- There can be many local minima which are not global minima • In the

context combinatorial problems, global optimization is NP-hard

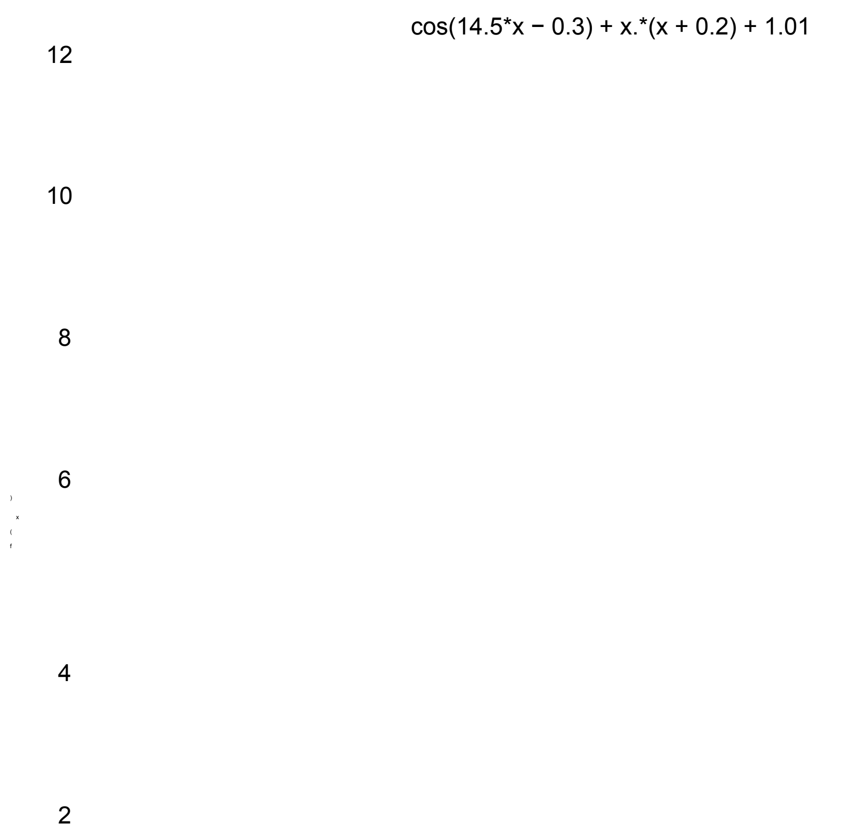
- Special properties (eg. convexity) of feasible region Ω and objective function f imply that any local solution is a global solution.

- References: Pinter [20]

One-dimensional example

- $f(x) = \cos(14.5x - 0.3) + x(x + 0.2) + 1.01$

$\Omega = [-3, 3]$



0

−3 −2 −1 0 1 2 3 x

Two-dimensional example

- $f(x) = \sum_{i=1}^{2} \left((x_i + 0.5)^4 - 30x_i^2 - 20x_i\right)/100$

- $\Omega = \{x \in \mathbb{R}^2 : -6 \le x_i \le 5,\ i = 1, 2\}$

5

4

3

2

1

0

x_2

−1

−2

−3

−4

−5

−6

−6 −5 −4 −3 −2 −1 0 1 2 3 4 5 x_1

Travelling Salesman Problem (TSP)

A salesman must visit every one of n cities exactly once and return to their starting city. If the cost of going from city i to city j is C_{ij}, find the route that minimizes the total cost.

- Variables
  - $X_{ij} = 1$ if go from city i to city j; 0 otherwise
  - $x_i = i$th city visited; Permutation of $\{1, 2, \ldots, n\}$
- Objective
  - $f(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$

- $f(x) = \sum_{i=1}^{n} c_{i,x_i}$

- Combinatorial optimization problem
  - $(n-1)!$ possible tours
  - Enumerating all tours, comparing costs $\Longrightarrow$ $n!$ operations
  - Impossible except for small numbers of cities
  - NP-hard
  - References [14]

Minimum Energy Problems

The optimal geometry is one which minimizes the total energy of the system.

- Protein folding: find the 3-dimensional protein structure given the sequence of amino acids of the protein
  - Variables: Positions of each amino acid, or relative positions (distances, angles)
  - Objective: Stretching, bending, torsion, electrostatic energy
  - Constraints: given order of amino acids,
- Example potentials
  - Reisz s-energy: $V(r) = 1/r^s$ ($s = 1$ Coulomb potential)
  - Leonard-Jones $V(r) = c_{12}/r^{12} - c_6/r^6$
- Characteristic: Many local minima
  - Number of local minima grows exponentially with problem size
  - Many local minima close to global minima
- General mathematical survey by Neumaier [17], others [19] Rob Womersley –

Potential between two particles

(Coulomb potential) 00.2 0.4 0.6 0.8 1

40

20

0

Reisz 1−energy

1.7 1.6 1.5 1.4 1.3

1.2 1.1 1

Reisz 0.1−energy

0 0.2 0.4 0.6 0.8 1

100 80

60

Reisz 4−energy

Leonard−Jones potential

3000

−1000

−2000

10x 10^7 8

6

2000

4

1000

2

0

0 0.2 0.4 0.6 0.8 1 Distance

−3000

0 0.01 0.02 0.03 0.04 Distance

0

Minimum Energy on the Sphere

- Particles (electrons) on the surface of the unit sphere
- Using Coulomb potential
- Voronoi cells around each particle gives positions of

atoms • 32 electrons gives C_{60}, other Carbon fullerenes

  - Stable configurations have few local minima, unstable configurations many

Minimum energy – Local minima

Exact methods

- Enumeration
  - Only possible for combinatorial problems
  - Exponential explosion makes only very small problems possible
- Branch and Bound
  - Bound: Relax some constraints $\Longrightarrow$ bound on objective value
  - Branch: Add constraints to remove infeasible points
- Interval methods
- References
  - Hansen [7], [9, 10]

Grids – Curse of Dimensionality

- *k* points in each variable; *n* variables
- Tensor product grid has $N = k^n$ points (Curse of Dimensionality) •

Example: $\Omega = [0, 1]^n$, $n = 1000$, $k = 2 \Longrightarrow N = 2^{1000} \approx 10^{300}$

20 by 20 grid, N = 400

1
0.9
0.8
0.7
0.6
x_2
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 x_1

Monte-Carlo Methods

- Minimize $f(x)$ over $x \in \Omega$
  - $f_{\min} = \infty$
  - For $k = 1, \ldots, N$
  - Generate $x^{(k)} \in \Omega$ uniformly distributed in Ω
  - Evaluate $f^{(k)} = f(x^{(k)})$
  - If $f^{(k)} < f_{\min}$ then $f_{\min} = f^{(k)}$; $x_{\min} = x^{(k)}$
- Generate $x^{(k)}$ by making random changes to $x^{(k-1)}$
- Rate of convergence: Probabilistic
  - Expected value of error $O N^{-1/2}$ (slow)
  - Independent of dimension n (very nice)
- Issues
  - Convergence/number of iterations

Difference with minimum function value

2

x_2

Two dimensional example

5

4

3

3.5

1

3

0

2

−2

2.5

−1

1.5
−3

−4

1

−5

−6
−6 −5 −4 −3 −2 −1 0 1 2 3 4 5

0

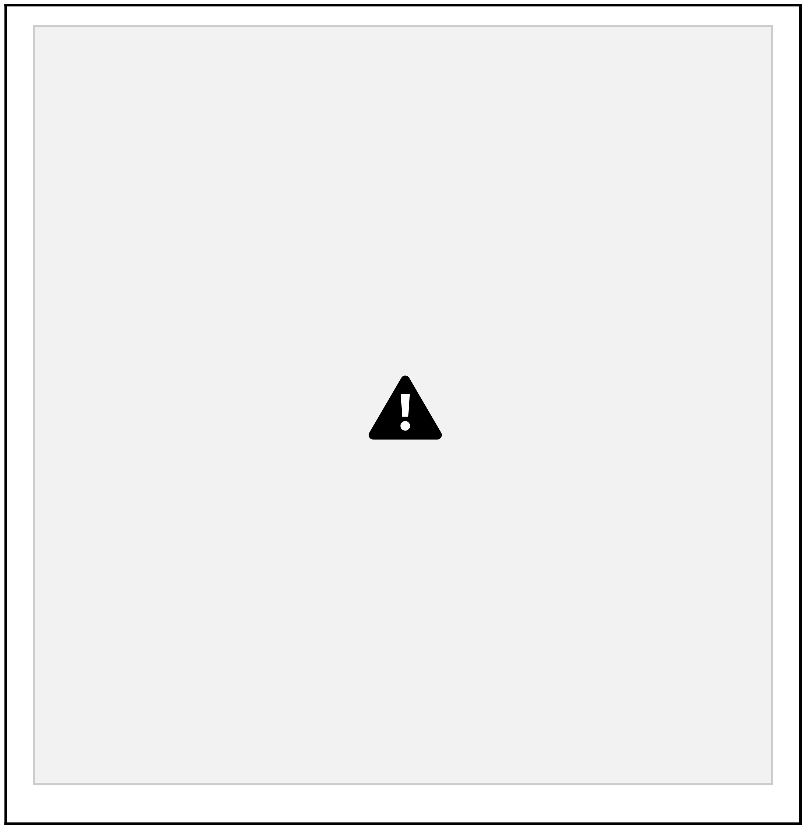
x_1

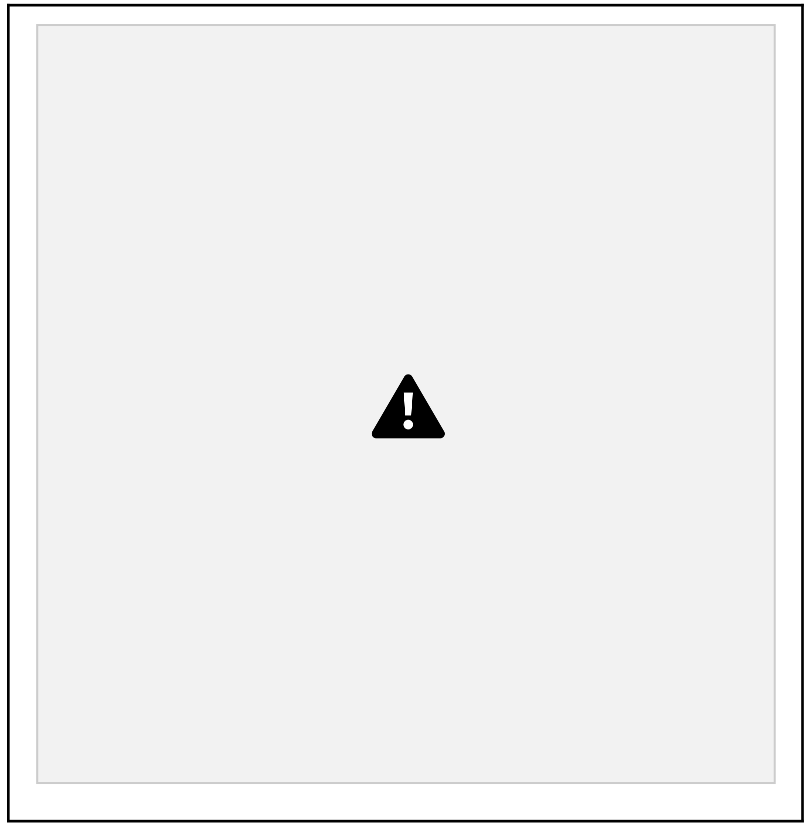
0 20 40 60 80 100 120 140 160 180 200 Iterations

0.5

Quasi Monte-Carlo (QMC) Methods

- QMC points are chosen deterministically to be ”well distributed” in $[0, 1]^n$.
- Examples: Sobol, Halton ,Faure, Niedereitter [18] (s, t)-nets, Lattices
- Pseudo-random vs Sobol points





Sparse Grids

- Selective points are chosen to explore very high dimensional space
- Jochen Garcke, Sparse Grid Tutorial [4].
- Michael Griebel http://wissrech.ins.uni-bonn.de/main/

From: http://wissrech.iam.uni-bonn.de/research/projects/zumbusch/fd.html

Simulated Annealing

- Annealing: a molten substance, initially at a high temperature and

disor dered, is slowly cooled so the the system is approximately in equilibrium. The frozen (minimum energy) ground state at $T = 0$ is ordered

- Generate new state $x^{(k+1)}$ of system:

. If energy $f(x^{(k+1)}) < f(x^{(k)})$, accept new state $x^{(k+1)}$;

. If the change in energy $\Delta f^{(k)} = f(x^{(k+1)}) - f(x^{(k)}) > 0$, accept $x^{(k+1)}$ with probability ~ $e^{-K\Delta f_{(k)}/T}$

- Issues:

. Generating new state;

. Initial temperature T_0; Cooling schedule

- References

. Metropolis (1953) [15]; Kirkpatrick (1983) [13], Cermy (1985) [2] . Numerical Recipes [21] - Second edition.
. Ingber Adaptive Simulated Annealing code (ASA) [11, 12]

Evolutionary Algorithms

• Inspired by Genetic Algorithms: natural selection and survival of fittest • Algorithm outline

. Population: many individuals $x^{(k)}$, with fitness $-f(x^{k)})$
. New population using genetic operators: recombination (crossover), mu tation, ... Use fitness of individuals to select those who survive

• GA usually applied to combinatorial optimization problems, with binary rep resentation of population. • Convergence to global optimum in weak probabilistic sense • Continuous variables versions (Michalewicz)

• Nonlinear constraints difficult: many individuals not feasible •

References

. Holland (1975) [8], Goldberg (1989) [5], Michalewicz [16] and [3] Rob

References

References

[1] AD, *Automatic differentiation*, tech. report, Univeristy of Aachen and Argonne National Laboratory, 2005. http://www.autodiff.org/.

[2] V. Cerny, *Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm*, Journal of Optimization Theory and Ap plications, 45 (1985), pp. 51–51.

[3] L. Davis, ed., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[4] J. Garcke, *Sparse grid tutorial*, tech. report, Tech nical University of Berlin, 2007. http://www.math.tu berlin.de/˜garcke/paper/sparseGridTutorial.pdf.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.

[6] A. Griewank, *Evaluating derivatives: principles and techniques of algorith mic differentiation*, SIAM, Philadelphia, 2000.

[7] E. R. Hansen, *Global Optimization using Interval Analysis*, Marcel

Dekker, New York, 1992.

[8] J. H. Holland, *Adaption in Natural and Artificial Systems*, University of Michigan Press, 1975. Reprinted by MIT Press, Cambridge MA, 1992.

[9] R. Horst and P. M. Pardalos, eds., *Handbook of Global Optimization*, Kluwer Academic, 1995.

[10] R. Horst, P. M. Pardalos, and N. V. Thoai, eds., *Introduction to Global Optimization*, Kluwer Academic, 2nd ed., 2000.

[11] L. Ingber, *Very fast simulated re-annealing*, Mathematical and Computer Modelling, 12 (1989), pp. 967–973.

[12] L. Ingber and B. Rosen, *Genetic algorithms and very fast simulated reannealing: A comparison*, Mathematical and Computer Modelling, 16 (1992), pp. 87–100.

[13] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.

[14] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem : A Guided Tour of Combinatorial Opti mization*, John Wiley, 1985.

[15] N. Metropolis, A. Rosenbluth, A. Teller, and E. Teller, *Equa tion of state calculations by fast computing machines*, Journal of Chemical Physics, 21 (1953), pp. 1–87–1092.

[16] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Pro grams*, Springer-Verlag, Berlin, second ed., 1994.

[17] A. Neumaier, *Molecular modelling of proteins and mathematical prediction of protein structure*, SIAM Review, 39 (1997), pp. 407–460.

[18] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.

[19] P. M. Pardalos, D. Shalloway, and G. Xue, *Optimization methods for computing global minima of nonconvex potentail energy functions*,

Journal of Global Optimization, 4 (1994), pp. 117–133.

[20] J. D. Pinter, *Global Optimization in Action*, Kluwer, Dordrecht, 1995.

[21] W. M. Press, B. Flannery, S. Teuklosky, and W. Vettering, *Nu merical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, 2nd ed., 1992. http://www.nr.com//.