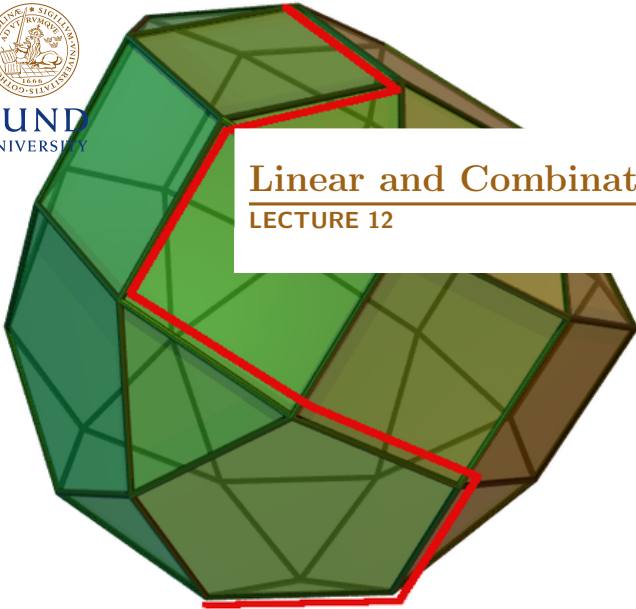




LUND
UNIVERSITY

Linear and Combinatorial Optimization 2020

LECTURE 12



Overview

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

1 Simulated annealing

2 Genetic algorithms

3 Applications of linear programming: L^1 regression

4 Piecewise linear optimization



LUND
UNIVERSITY

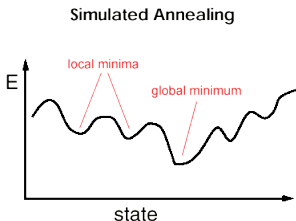
Simulated annealing

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



A function with many local minima

- Simulated annealing is a heuristic method which can be used to find a global optimum for functions with many local maxima or minima.
- When trying to optimize such functions, other algorithms (like e.g. local search) often gets stuck at a local optimum.
- It is not certain that an optimum will be found with simulated annealing either, but it works well in practice for many problems.



LUND
UNIVERSITY

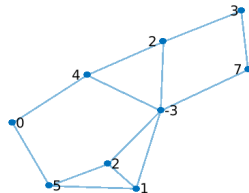
Simulated annealing

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



A function defined on a graph. The nodes are marked with the function values, and the edges tell us which points are neighbours.

- This type of method can be used for continuous or discrete optimization problems, like for example a function defined on a graph or a lattice.
- We can start with a *local search*:
 - 1 Pick a random point.
 - 2 Go to a neighbour with a smaller value.
 - 3 Repeat until this is not possible.
 - 4 This procedure will find a local minimum, but it may not be a global minimum.



Simulated annealing

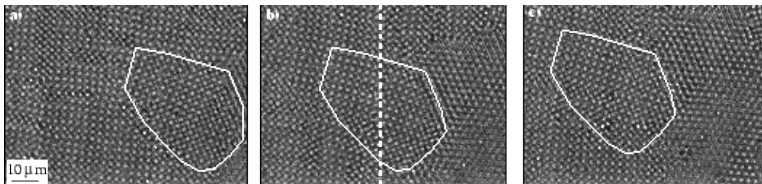
Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization

- Simulated annealing is an approximate method which aims to overcome the problem of getting trapped at a local minimum.
- The idea comes from physics:
 - Fine crystals in a solid of low energy.
 - First melt the solid by increasing the temperature.
 - Then cool the liquid *slowly* so that it crystallizes.
 - The energy is minimized. Too fast cooling leads to imperfections (corresponding to a local minimum).



Optical annealing of a misoriented grain. See

<http://physics.nyu.edu/grierlab/anneal7b/>

Sara Maad Sasane

Linear and Combinatorial Optimization

28 February 2020 5/30



LUND
UNIVERSITY

Simulated annealing

Simulated annealing

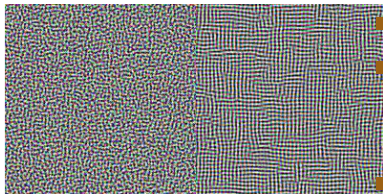
Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY



Simulated annealing with a fast cooling schedule (left) and a slow cooling schedule (right) for an optimization problem involving swapping pixels in an image.

- Annealing means solidifying.
- The analogue of the perfect chrystalized structure is the global minimum that we wish to find in the optimization problem.
- A feasible solution corresponds to a state of the physical system.
- The objective function corresponds to the energy.
- There is a control parameter c , which corresponds to the temperature.

Simulated annealing

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

Algorithm idea:

- 1 Pick c (the temperature) and L (the number of steps for each value of c).
- 2 Let $k = 0$ and pick a random point $x_0 (= x_k)$.
- 3 Generate a new point \bar{x} (e.g. a neighbour of x_k).
- 4 Should we accept \bar{x} as our x_{k+1} ?
 - Yes (always) if $f(\bar{x}) \leq f(x_k)$,
 - Yes with probability $\exp\left(\frac{f(x_k) - f(\bar{x})}{c}\right)$ if $f(x_k) < f(\bar{x})$.
- 5 Repeat steps 3 and 4 L times.
- 6 Update (decrease) c and possibly change L .
- 7 Stop if some stopping criterion is satisfied (the system is said to be frozen). Otherwise, start over from step 3 using the last obtained solution as a starting point.

Simulated annealing

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

- Note that in step 4, an improvement is always accepted while a worse solution is sometimes accepted, with a probability that depends on how much much worse the solution is, and the cooling parameter c .
- The aim is to escape local minimum traps.
- Simulated annealing is an approximate algorithm that works well in practice for some problems.
- There is a theoretical result that says that when $c \rightarrow 0$, a global minimum is reached with probability 1 (*=almost surely* in probability language), but reaching this global minimum may take longer than going through all points in the feasible set and comparing the values.

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

- Genetic algorithms belong to another type of heuristic algorithms designed with the aim of finding a global minimum of a complicated function with many local minima, when the search space is very large.
- Just as with simulated annealing, the inspiration comes from nature, but this time from biology/evolution.
- The ingredients are:
 - Natural selection,
 - Reproduction,
 - Mutation.
- In genetic algorithms, one works with a whole generation of feasible points at once. These are called individuals or chromosomes.

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

Example (The knapsack problem)

Use a binary encoding. In the knapsack problem, there are n items, and each item gets assigned the number (a gene) 0 or 1 (depending if the items should be taken or not). If for example there are 7 items, then a possible chromosome (individual) is 1001011. A chromosome can be randomly generated. If it is not feasible (if the weight/size is too high), then we throw it away and pick another random one.

Example (Travelling salesman)

Here we use permutation encoding. 157382469 represents a tour of 9 cities.

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

- A population consists of a small number N of chromosomes, chosen from the large search space. Usually $N \approx 30$, but what works well depends on the problem.
- Reproduction is done by crossover. Choose two parents from the population.
- Ex. with binary encoding: 1011010 and 0110101. If *single crossover* is used, then a crossover point is selected, and genes are swapped at that location. If the crossover point is 4, the offspring of the above chromosomes will be 1011101 and 0110010, i.e. the first four genes of the offspring come from one of the parents and the rest come from the other parent.

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization



LUND
UNIVERSITY

- If permutation encoding is used, then a possible crossover rule is the following (done in an example with crossover point 5): With parents 123456789 and 453689721, let the 5 first genes of the offspring come from one of the parents, and the rest are taken in order from the other parent if they are not already present in the five first genes of the offspring). With this rule, the offspring of the above parents will be 123456897 and 453681279.
- The offspring is kept in a new generation.

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

- Mutation: Make rare random changes.
 - For example with binary encoding: With a small probability, change 0 to 1 and 1 to 0.
 - For example, with permutation encoding, 2 genes are chosen randomly (with a small probability) and exchanged.
- Elitism: It is usually a good idea to save the best individuals as they are.



LUND
UNIVERSITY

Genetic algorithms

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

We give an outline of the idea of a basic genetic algorithm.

- 1 Generate a random generation of N chromosomes.
- 2 Evaluate the fitness (i.e. objective function value) of each chromosome in the population.
- 3
 - *Select* chromosomes according to some rule where “better” chromosomes are more likely to be selected.
 - *Crossover*: Form new offspring of selected parents using crossover with some crossover probability.
 - *Mutation*: With a mutation probability for each gene, mutate the new offspring.
 - *Place* the offspring in a new population.
- 4 *Replace*: Use the new generation for a further run of the algorithm.
- 5 *Continue* (i.e. repeat from 2) until some stopping criterion is satisfied (e.g. no improvement in the mean for several generations).

Genetic algorithms

Simulated annealing

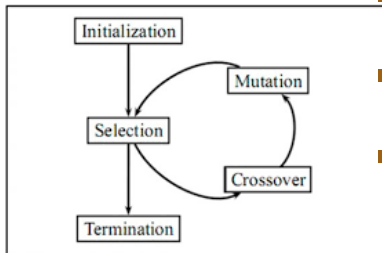
Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY



The outline of a genetic algorithm

- The method is easy to implement but hard to analyze.
- Convergence of genetic algorithms tend to be slow.
- Here is a list of applications of genetic algorithms in many different fields in engineering, economics, linguistics, etc.
- To try out simulated annealing and/or a genetic algorithm yourself, you can do the optional computer exercise.

Regression – line fitting

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

- We consider the problem of fitting a line to a set of given data points (x_j, y_j) , $j = 1, \dots, n$.
- The most common way of doing this is to use least squares. This is simple to do, and it is also preferred if the errors are normally distributed. With this assumption, the least squares line is the maximum likelihood estimator.
- If the errors cannot be assumed to be normally distributed, other norms can be considered instead of the L^2 norm that is used in the least squares method.



LUND
UNIVERSITY

Least squares – L^2 regression

Simulated annealing

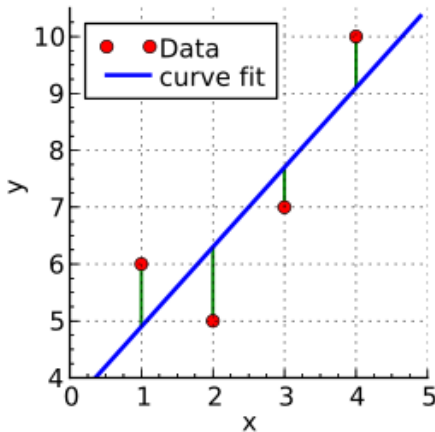
Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY



- The problem is to find the parameters k and m such that the sum of the squared distances (in the y direction) from the data points to the line $y = kx + m$ is minimized:

$$\min_{k, m \in \mathbb{R}} \sum_{j=1}^m (y_j - kx_j - m)^2.$$

- Note that finding the optimal k and m is equivalent to finding the “best” line.

The least squares line is the line for which the sum of the squared distances is

minimized

Least squares – L^2 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

- There is a closed formula for finding k and m . To find it, just differentiate the expression on the previous slide with respect to k and m , and set the derivatives to 0. You will obtain a linear system that can be solved.
- Convexity implies that the solution is a minimizer.
- The least squares fitted line is sensitive to outliers.
- If the assumption on normally distributed errors is not satisfied, it may be better to use L^1 regression, which is less sensitive to outliers.



LUND
UNIVERSITY

Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



LUND
UNIVERSITY

- In L^1 regression, we solve the minimization problem

$$\min_{k, m \in \mathbb{R}} \sum_{j=1}^n |y_j - kx_j - m|. \quad (1)$$

- The difference from least squares is that we sum the distances from the points to the line (still in the y direction), without squaring them first.
- Unlike the situation for least squares, there is no closed formula for the minimizer, *but* we can rewrite the problem as an LP problem, and solve it efficiently using the simplex method.

Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

- In order to write the minimization problem as an LP problem, we introduce new variables t_j , $j = 1, \dots, n$, i.e. one for each data point.
- Clearly, (1) is equivalent to

$$\begin{aligned} & \text{minimize} \quad \sum_{j=1}^n t_j \\ & \text{subject to} \quad \begin{cases} -t_j = y_j - kx_j - m \text{ or } t_j = y_j - kx_j - m, \\ k, m \in \mathbb{R} \text{ (unrestricted),} \\ t_j \geq 0, \quad j = 1, \dots, n. \end{cases} \end{aligned} \quad (2)$$

- Problem (2) is also not an LP problem, and so we will rewrite it one more time.



LUND
UNIVERSITY

Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization



LUND
UNIVERSITY

Theorem

The L^1 regression minimization problem (1) is equivalent to the LP problem

$$\begin{aligned} & \text{minimize} \quad \sum_{j=1}^n t_j \\ & \text{subject to} \quad \begin{cases} -t_j \leq y_j - kx_j - m \leq t_j, & j = 1, \dots, n, \\ k, m \in \mathbb{R} \text{ (unrestricted)}, \\ t_j \geq 0, & j = 1, \dots, n, \end{cases} \end{aligned} \quad (3)$$

in the sense that, (k_0, m_0) is an optimal solution of (1) if and only if there exist $t_j \geq 0, j = 1, \dots, n$, such that $(k_0, m_0, t_1, \dots, t_n)$ is optimal for (3).

Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization



LUND
UNIVERSITY

Proof.

We will prove that (2) \Leftrightarrow (3), since we have already seen that (1) and (2) are equivalent.

\Rightarrow) Suppose that $(k^*, m^*, t_1^*, \dots, t_n^*)$ minimizes (2). Clearly, $(k^*, m^*, t_1^*, \dots, t_n^*)$ is a feasible point for (3). We will now show that it is optimal. We argue by contradiction, and suppose that this is not the case. Then there exists a point (k, m, t_1, \dots, t_n) which is feasible for (3), and has a smaller objective value. It must satisfy $-t_j < y_j - kx_j - m < t_j$ for some j , since otherwise it would be feasible for (2) with a strictly smaller objective value than the value of the minimizer $(k^*, m^*, t_1^*, \dots, t_n^*)$, a contradiction. But then t_j could be reduced (and hence reducing the objective value) while (k, m, t_1, \dots, t_n) still remains in the feasible set of (3). Hence (k, m, t_1, \dots, t_n) cannot be a minimizer for (3), and we have reached a contradiction.

Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization



LUND
UNIVERSITY

Proof (Cont.)

⇒ Suppose that $(k_0, m_0, t_1, \dots, t_n)$ is a minimizer for (3). If we can show that it belongs to the feasible set of (2), then it will be a minimizer for (2) as well, since the feasible set of (2) is a subset of the feasible set of (3). Hence, we would need to prove that $t_j = |y_j - k_0 - m_0|$ for $j = 1, \dots, n$. Again, we argue by contradiction, and assume that $|y_j - k_0 - m_0| < t_j$ for some j . But then t_j could be reduced to obtain a feasible point of (3) with a lower objective value. This is a contradiction, and hence $(k_0, m_0, t_1, \dots, t_n)$ is feasible (and optimal) for (2) as well.



Least sum of distances – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization



LUND
UNIVERSITY

- We have showed that the L^1 regression problem is equivalent to

$$\begin{aligned} & \text{minimize} && \sum_{j=1}^n t_j \\ & \text{subject to} && \begin{cases} -x_j k - m - t_j \leq -y_j, & j = 1, \dots, n, \\ x_j k + m - t_j \leq y_j, & j = 1, \dots, n, \\ k, m \in \mathbb{R} & \text{(unrestricted),} \\ t_j \geq 0, & j = 1, \dots, n, \end{cases} \end{aligned} \quad (4)$$

- It can be put in standard form by replacing k and m by the four new variables k_+ , k_- , m_+ and m_- , where the relationship between the old and new variables are $k = k_+ - k_-$ and $m = m_+ - m_-$.

Least sum of squares – L^1 regression

Simulated annealing

Genetic algorithms

Applications of linear
programming: L^1
regression

Piecewise linear
optimization

The resulting LP problem on standard form is

$$\begin{cases} \text{maximize } \mathbf{c}^T \mathbf{U} \\ \text{subject to } \begin{bmatrix} -\mathbf{X} & \mathbf{X} & -\mathbf{1} & \mathbf{1} & -\mathbf{I} \\ \mathbf{X} & -\mathbf{X} & \mathbf{1} & -\mathbf{1} & -\mathbf{I} \end{bmatrix} \mathbf{U} \leq \begin{bmatrix} -\mathbf{Y} \\ \mathbf{Y} \end{bmatrix}, \end{cases} \quad (5)$$

where $\mathbf{U} = [k_+, k_-, m_+, m_-, t_1, \dots, t_n]^T$, $\mathbf{c}^T = [0, 0, 0, 0, -\mathbf{1}^T]$, \mathbf{X} and \mathbf{Y} are the column vectors with entries x_i and y_i ($i = 1, \dots, n$), respectively, $\mathbf{1}$ is a column vector of length n whose entries are all 1 and \mathbf{I} is an identity matrix of size n .

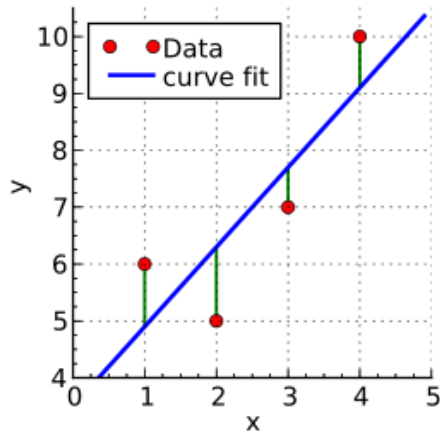


LUND
UNIVERSITY

Least sum of squares – L^1 regression

Example

- Let's try this out for the data set that we saw in the figure earlier:
- The data points are (1,6), (2,5), (3,7) and (4,10).
- To find the L^1 regression line, we need to solve the LP problem given on the next slide.
- In general, there will be $2n$ constraints and $n + 4$ variables if there are n data points, for the LP problem on standard form.



Least sum of squares – L^1 regression

Example

The LP problem on standard form is

$$\left\{ \begin{array}{l} \text{maximize } [0 \ 0 \ 0 \ 0 \ -1 \ -1 \ -1 \ -1] \mathbf{u} \\ \text{subject to} \\ \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 \\ -2 & 2 & -1 & 1 & 0 & -1 & 0 & 0 \\ -3 & 3 & -1 & 1 & 0 & 0 & -1 & 0 \\ -4 & 4 & -1 & 1 & 0 & 0 & 0 & -1 \\ 1 & -1 & 1 & -1 & -1 & 0 & 0 & 0 \\ 2 & -2 & 1 & -1 & 0 & -1 & 0 & 0 \\ 3 & -3 & 1 & -1 & 0 & 0 & -1 & 0 \\ 4 & -4 & 1 & -1 & 0 & 0 & 0 & -1 \end{bmatrix} \mathbf{u} \leq \begin{bmatrix} -6 \\ -5 \\ -7 \\ -10 \\ 6 \\ 5 \\ 7 \\ 10 \end{bmatrix}, \\ \mathbf{u} \geq 0. \end{array} \right.$$



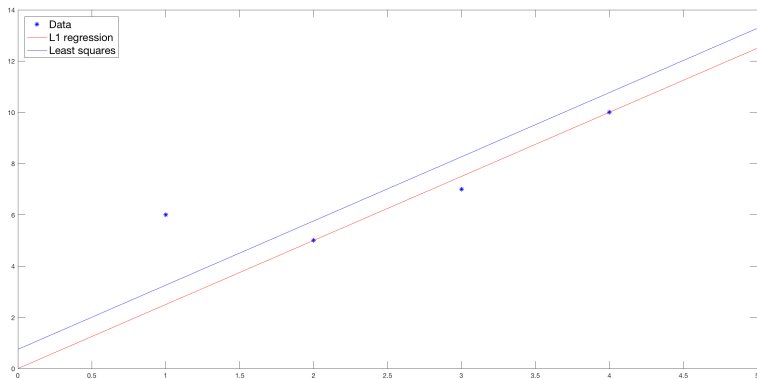
Simulated annealing

Genetic algorithms

**Applications of linear
programming: L^1
regression**

Piecewise linear
optimization

Example



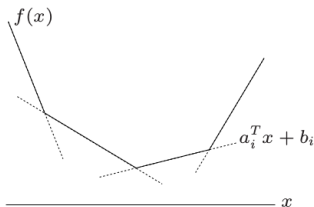
Piecewise linear optimization

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization



- Using the same idea, it is possible to rewrite any convex, piecewise linear minimization problem as an LP problem.
- A convex piecewise linear minimization problem is a problem whose feasible set is a convex polyhedron and whose objective function is convex and piecewise linear.
- A convex piecewise linear function can be written on the form

$$f(\mathbf{x}) = \max_{i=1,\dots,m} (\mathbf{a}_i^T \mathbf{x} + b_i).$$

A convex piecewise linear function



LUND
UNIVERSITY

Piecewise linear optimization

Simulated annealing

Genetic algorithms

Applications of linear programming: L^1 regression

Piecewise linear optimization

By introducing a new variable t , the optimization problem of *minimizing*

$$f(\mathbf{x}) = \max_{i=1,\dots,m} (\mathbf{a}_i^T \mathbf{x} + b_i)$$

subject to some linear inequality and equality constraints, is equivalent to

minimize t ,

$$\text{subject to } \begin{cases} \mathbf{a}_i^T \mathbf{x} + b_i \leq t, & i = 1, \dots, m, \\ + \text{ the original constraints.} \end{cases}$$

Note that this is an LP problem with m additional constraints and one extra variable.



LUND
UNIVERSITY