

Small problem: If an element in constant-column is zero  $\rightarrow$  degenerate feasible vector.

May be necessary to exchange degenerate l.h.s. variable for right hand variable.

Transforming the general linear programming problem  
to restricted normal form.

An example of a general problem:

$$Z = x_1 + x_2 + 3x_3 - \frac{x_4}{2}$$

$$x_i \geq 0, \quad i = 1, \dots, 4$$

$$\begin{aligned} x_1 + 2x_3 &\leq 740 \\ 2x_2 - 7x_4 &\leq 0 \end{aligned} \quad \left. \right\} m_1 = 2$$

$$x_2 - x_3 + 2x_4 \geq \frac{1}{2} \quad m_2 = 1$$

$$x_1 + x_2 + x_3 + x_4 = 9 \quad m_3 = 1$$

We introduce slack variables.

This will make the  $m_1 + m_2$  inequalities among the constraints into equalities.

We need  $m_1 + m_2$  of them.

In our example:

$y_1, y_2, y_3$

$$x_1 + 2x_3 + y_1 = 740$$

$$2x_2 - 7x_4 + y_2 = 0$$

$$x_2 - x_3 + 2x_4 - y_3 = \frac{1}{2}$$

Sign reflects  
the type of  
inequality.

$$y_i \geq 0 \quad i=1, \dots, 3.$$

In addition, we introduce

$$M = M_1 + M_2 + M_3$$

Artificial variables.  $z_1, \dots, z_M$ .

$$z_1 = 740 - x_1 - 2x_3 - y_1$$

$$z_2 = -2x_2 + 7x_4 - y_2$$

$$z_3 = \frac{1}{2} - x_2 + x_3 - 2x_4 + y_3$$

$$\begin{aligned} z_i &\geq 0 \\ i &= 1, \dots, 4 \end{aligned}$$

$$z_4 = 9 - x_1 - x_2 - x_3 - x_4$$

We also introduce the  
Auxiliary object function

$$z' = -z_1 - z_2 - z_3 - z_4$$

$$\begin{aligned} (= -749\frac{1}{2} + 2x_1 + 4x_2 + 2x_3 - 4x_4 \\ + y_1 + y_2 - y_3) \end{aligned}$$

$z'$  is maximal for  $z_1 = z_2 = z_3 = z_4 = 0$ .

We generate an initial  
 feasible vector:

	$x_1$	$x_2$	$x_3$	$x_4$	$y_1$	$y_2$	$y_3$
$Z$	0	1	1	3	$-\frac{1}{2}$	0	0
$Z_1$	$240$	-1	0	-2	0	-1	0
$Z_2$	0	0	-2	0	7	0	-1
$Z_3$	$\frac{1}{2}$	0	-1	1	-2	0	0
$Z_4$	9	-1	-1	-1	-1	0	0
$Z'$	$-249\frac{1}{2}$	2	4	2	-4	1	1

→ Make all coefficients along this row negative.

Then, all the  $Z_i$  variables have become right hand variables and may be not equal to zero.

Then, pivot among the  $x_i$  and  $y_i$  variables to minimize  $Z$ :

Result:

	$x_1$	$y_2$	$y_3$	$z_1$
$z$	17.03	-0.95	-0.05	-1.05
$x_2$	3.33	-0.35	-0.15	0.35
$x_3$	4.73	-0.55	0.05	-0.45
$x_4$	0.95	-0.10	0.10	0.10
$y_1$	730.55	0.10	-0.10	0.90
$z'$	---	---	---	---

All coefficients negative

Along this row: We have  
the solution.

If the algorithm fails,

the constraints are not

consistent and no

solution exists.

## Non-linear programming

Optimize  $F(x)$

given the constraints

$$\begin{cases} g_i(x) = 0 & i=1, \dots, m_1 \\ h_j(x) \geq 0 & j=m_1+1, \dots, m_1+m_2. \end{cases}$$

If  $F(x)$  is a quadratic form  $\Rightarrow$

Quadratic programming.

If  $F(x)$  is non linear enough,

multiple extremal values.

- global optimization.

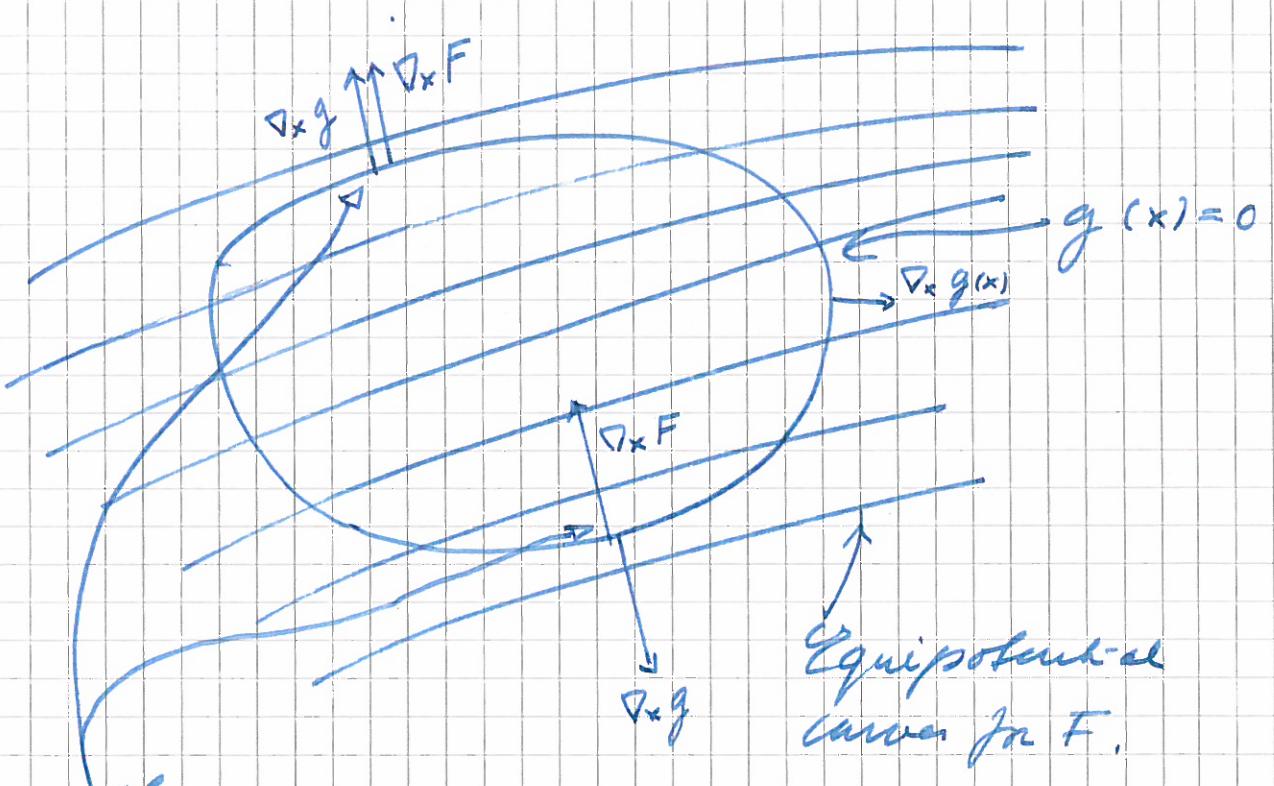
At optimal  $x$ :

$$\left\{ \begin{array}{l} \nabla_x L = \nabla_x F(x) + \lambda \nabla_x g(x) = 0 \\ \nabla_\lambda L = g(x) = 0 \end{array} \right.$$

Why this?

Hence, optimize  $L(x, \lambda)$  with respect to  $(x, \lambda)$ .

How does this actually work?



Then  $\nabla_x g$  and  $\nabla_x F$  are parallel and we can have

$$\nabla_x F + \lambda \nabla_x g = 0.$$

Clearly, the two points where  $\nabla_x F$  and  $\nabla_x g$  are parallel along the curve

$g(x) = 0$  represent the optimal points for  $F$  given the constraint.

---

Example:

$$\text{minimize } F(\vec{x}) = x^2 + y^2$$

$$\text{given } g(\vec{x}) = x - y = 0$$

$$L(\vec{x}, \lambda) = x^2 + y^2 + \lambda(x - y)$$

$$\nabla_{\vec{x}} L = (2x + \lambda, 2y - \lambda) = (0, 0)$$

$$\nabla_{\lambda} L = 0 = x - y$$

$$2x + \lambda = 0$$

$$2y - \lambda = 0$$

$$x = y$$

$$x = y = 0$$

$$\underline{\lambda = 0}$$

## Genetic algorithms

We start with a population of strings (genes)

0 0 1 1 0 0 1 0 0 ... 0 1 0 0 1 1

We wish to optimize a fitness function whose inputs are the bit strings.

Optimization proceeds through a three-step evolution-like algorithm.

3 steps evolution:

1. Reproduction

Old population

→ New population.

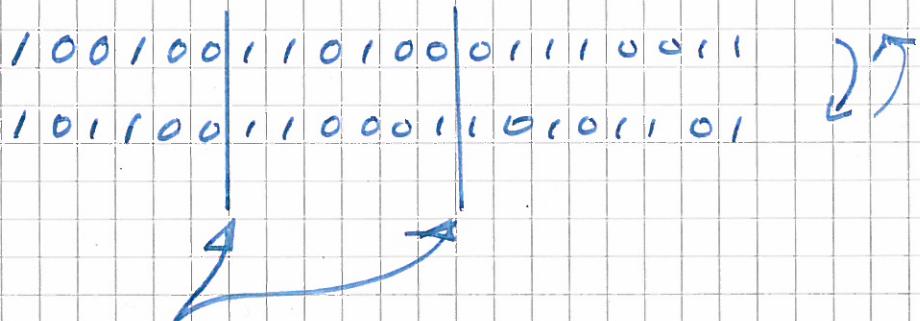
Every individual (string) is copied with probability proportional to its fitness function.

The number of individuals is kept fixed from generation to generation  $\Rightarrow$  individuals may be copied more than once.

## 2. Recombination.

Exchange of string bits.

Mating of 2 groups individuals in pairs:



position of cuts are chosen at random.

Exchange the cut-out pieces.

### 3. Mutation

Random reversal of bits  
with low probability.

... 0100110 ...  
 ↓  
 ... 0100010

### Simulated Annealing

Kirkpatrick et al.

Science, 220, 671 (1983).

We want to find the global  
minimum of the function  $E(x)$

Can be a very  
high-dimensional  
space.

Treat  $E$  as an energy in a thermal system, temperature  $T$ .

$$p(\vec{x}) \propto e^{-E(\vec{x})/k_b T}$$

Use the Monte Carlo to generate a sequence of configurations at a given temperature. Let the temperature fall towards zero.

→ minimize energy function.

But, "energy" does not have to be a physical energy.

Simulated annealing is a general optimization tool.

The temperature is lowered by the rule  $T \rightarrow CT$  where  $C < 1$ .

In practice, it will behave as

$$T \propto \frac{1}{\ln t}$$

↑  
time.

---

As  $t \rightarrow \infty$ , simulated annealing will find the global minimum with probability 1.

In practice, calculation ends after a finite time.

What is the best strategy to cool the system in a finite time?

Start by finding a criterion  
for what is optimal.

- (1) End energy  $E(t)$ .
- (2) BSF ("Best so far") energy,

$$E_{BSF}(t) = \min_{0 \leq t' \leq t} E(t')$$

The following adaptation scheme for temperature control worked well:

We need an ensemble.



Copy of system  
 whose energy  
 function we are  
 to minimize.

Each copy corresponds to  
a different configuration.

Follow the ensemble average of the  
energy,  $\langle E \rangle$ .

1) Set an initial temperature  $T_0$ .

2) Make  $M$  Monte Carlo steps, i.e.  
a Monte Carlo sweep. Repeat  
this  $j$  times.

$M$  is the number of degrees of  
freedom in the system.

3) Let  $\langle E \rangle_j$  be the ensemble  
average of the energy  $j$  Monte  
Carlo sweeps.

If  $\langle E \rangle_j > \langle E \rangle_{j-1}$ , return to 2.

If  $\langle E \rangle_j \leq \langle E \rangle_{j-1}$ , continue.

4) Adjust temperature  $T_{j+1} = T_j \cdot c$   
 where  $c < 1$ .

5) If maximum number of Monte Carlo sweeps has been attained,  
 stop.

Otherwise, return to 2.

Monte Carlo sweeps is the natural  
 "time" unit in Monte Carlo  
 simulations, making it possible  
 to compare systems of different  
 size.

Even though the temperature  
 is lowered by a fixed factor  $c$ ,  
 the algorithm is not

exponential in temperature decrease, as the number of sweeps varies from temperature to temperature.

---

Fluctuations mask the real behavior of the energy means as the ensemble is finite, and the criterion that is used will lower the temperature even though some of the ensemble members are stuck in local minima.

Sooner or later, the ensemble will no longer be in thermal equilibrium.

↑ By this we mean that the distribution of energies no longer follows the

241

Boltzmann distribution,  
 $e^{-E/k_B T}$ .

This leads us to

Simulated tempering.

Marinari and Parisi

Europhys. Lett. 19, 451 (1992).

Basic idea: Monte Carlo update  
 of temperature.

An Markov chain will not only  
 move in energy space but also  
 in temperature space.

We have a set of discrete  
 temperatures  $T_i \in \{T_1, \dots, T_m\}$ .

2/2

We define

$$\beta_i = \frac{1}{T_i}.$$

Boltzmann fctn:

$$w(E_\alpha) \propto e^{-\beta E_\alpha}$$

We generalize

$$w(E_\alpha, \beta_\alpha) \propto e^{-\beta_\alpha E_\alpha + g(\beta_\alpha)}$$

This is a function  
that controls the  
temperature distribution.

How should  $g$  be  
chosen?

Partition function

$$Z(\beta) = e^{-\beta f(\beta)} = \sum_{\alpha} e^{-\beta E_\alpha}.$$

*(Probability of states)*

243

Optimal choice:

$$g(\beta) = \beta f(\beta)$$

↑  
free energy.

Then, probability distribution for temperature irrespective of configuration,

$$w(\beta) \propto \sum_{\alpha} e^{-\beta E_{\alpha}} + g(\beta) = 1.$$

Any choice for  $g$  is possible.

One optimal choice: Better choice.

We can e.g. choose  $g(\beta) = 1$ .

We can estimate  $\beta f(\beta)$  by some initial runs for different temperatures.

Algorithm is then as follows.

For a given sample (in ensemble):

- 1) Trial move in temperature space:

$$\beta_i \rightarrow \beta_{i+1} \text{ or } \beta_{i-1}.$$

- 2) Calculate  $\Delta g_{i,i\pm 1} = g(\beta_{i\pm 1}) - g(\beta_i)$

$$\Delta g_{i,i-1} = g(\beta_{i-1}) - g(\beta_i)$$

- 3) If  $\Delta g_{i,i\pm 1} \geq 0$ , accept move.

- 4) Otherwise, generate random #  $r$ .

if  $e^{-\Delta g_{i,i\pm 1}} \geq r$ , accept move

if  $e^{-\Delta g_{i,i\pm 1}} < r$ , reject move.

5)

245

Perform standard Monte Carlo  
move in configuration space,

$$E_\alpha \rightarrow E_\beta \dots$$

Optimization; finding minimum  $E$ :  
follow energy of sample  
with smaller temperature.

---

Other optimization methods:

Potential Annealing

