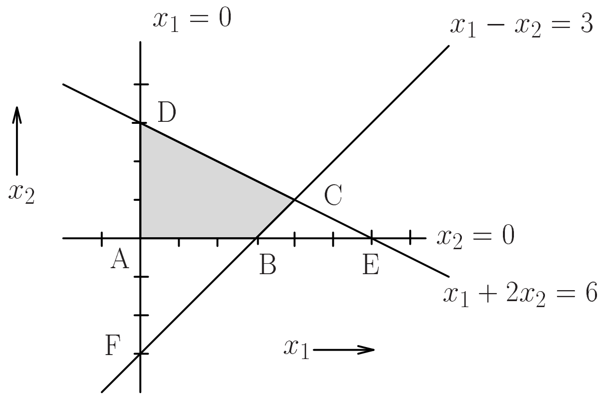
Feasible set for P



$$
\begin{aligned}
\text{P:}\quad & \text{maximize} \quad x_1 + x_2 \\
& \text{subject to} \quad x_1 + 2x_2 \le 6 \\
& \qquad\qquad x_1 - x_2 \le 3 \\
& \qquad\qquad x_1, x_2 \ge 0
\end{aligned}
$$

Example: set partitioning

Given n positive numbers $a_1, \dots, a_n$, partition them into two disjoint sets, A and $\bar{A}$, such that the sums of the numbers in the two sets are as equal as possible.

Example: job scheduling

Given n tasks of lengths $a_1, \dots, a_n$, process them on two machines operating in parallel so as to complete all n tasks in the minimal time possible.

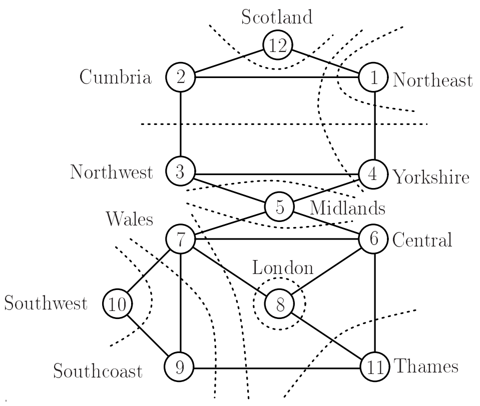
These problems can be formulated as an ILP

$$
\begin{aligned}
& \text{minimize} \quad x_0 \\
& \text{subject to} \quad \textstyle\sum_i x_i a_i \le x_0 \\
& \qquad\qquad \textstyle\sum_i (1 - x_i) a_i \le x_0 \\
& x_i \in \{0, 1\},\ i = 1, \dots, n,\ \text{and } x_0 \ge 0
\end{aligned}
$$

Here x_i is 1 or 0 as $i \in A$ or $i \in \bar{A}$. The first two inequalities require the sum of the items in A and $\bar{A}$ to each be less that x_0 (which is unconstrained).

Hence x_0 is minimized when the sums of the numbers in the two sets are a nearly equal as possible.

Example: optimal power generation and distribution



Demand for electricity at node i is d_i. Node i has k_i generators, that can generate electricity at costs of $a_{i1}, \dots, a_{ik_i}$, up to amounts $b_{i1}, \dots, b_{ik_i}$. Capacity for transmission from node i to j is c_{ij} $(= c_{ji})$. Let

y_{ij} = MW generated by generator j at node i.
x_{ij} = MW carried $i \to j$.

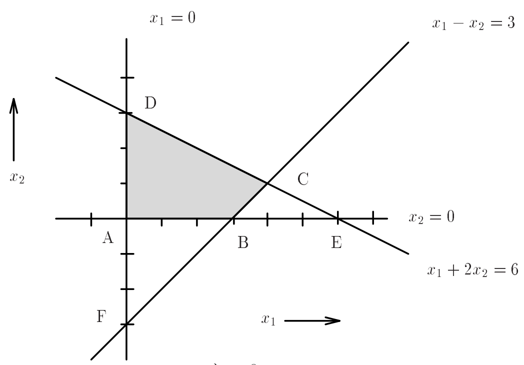
The LP is

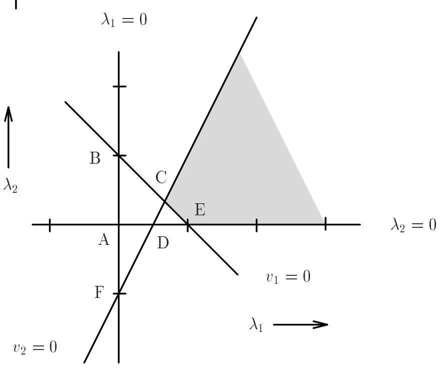
$$
\text{minimize} \sum_{ij} a_{ij} y_{ij}
$$

$$
\text{subject to} \sum_j y_{ij} - \sum_j x_{ij} + \sum_j x_{ji} = d_i,
$$

$$
0 \le x_{ij} \le c_{ij},\ 0 \le y_{ij} \le b_{ij}.
$$

Feasible sets:





Basic solutions:

Primal:

	x_1	x_2	z_1	z_2	f
A	0	0	6	3	0
B	3	0	3	0	3
C	4	1	0	0	5
D	0	3	0	6	3
E	6	0	0	-3	6
F	0	-3	12	0	-3

Dual:

	v_1	v_2	λ_1	λ_2	f
A	-1	-1	0	0	0
B	0	-2	0	1	3
C	0	0	$\frac{2}{3}$	$\frac{1}{3}$	5
D	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	3
E	0	1	1	0	6
F	-2	0	0	-1	-3

Example: insects as optimizers

A colony of insects consists of **workers** and **queens**, of numbers $w(t)$ and $q(t)$ at time t.

A time-dependent proportion $u(t)$ of the colony's effort may be put into producing workers, $0 \le u(t) \le 1$. The problem is

$$\begin{aligned} &\text{maximize} && q(T) \\ &\text{subject to} && dq/dt = c(1-u)w \\ & && dw/dt = auw - bw \\ & && 0 \le u \le 1. \end{aligned}$$

where a, b, c are constants, with $a > b$.

This optimization problem is substantially more complicated than the LP problem. Firstly, there are effectively an infinite number of variables and constraints (since the differential equations must hold at every time t, $0 \le t \le T$). Secondly, the constraints between the variables w, q and u are non-linear.

There exists a beautiful method to solve this type of problem (taught in *Optimization and Control* IIB). The optimal policy is to produce only workers up to some moment, and produce only queens thereafter. This is what such insects do in practice!

Example: bin packing

Given n rational numbers $a_1, \dots, a_n \in (0,1)$, partition them into the minimum number of subsets such that the sum of the numbers in each subset is ≤ 1.

This is a *bin packing problem* in which items of sizes $a_1, \dots, a_n$ are to be packed into the minimum number of bins of size 1. It can be formulated as an ILP problem in decision variables $x_{ij},\ y_j$,

$$\begin{aligned} &\text{minimize} && \textstyle\sum_j y_j \\ &\text{subject to} && \textstyle\sum_i x_{ij} a_i \le y_j \\ & && x_{ij}, y_j \in \{0,1\},\ i,j \in \{1,\dots,n\} \end{aligned}$$

Here y_j is 1 or 0 as a jth bin is or is not used and x_{ij} is 1 or 0 as item i is or is not placed in bin j. Clearly, no more than n bins are needed and $\sum_j y_j$ is minimized when the items are packed into the minimum number of bins.

Here the *problem size* is the number of bits of data required to specify an *instance* of the problem (i.e., to specify n and $a_1, \dots, a_n$). All known algorithms have running time which grows essentially like $e^{(\textit{problem size})}$.

Klee and Minty's Example

Consider the problem with n variables and n constraints:

$$\begin{aligned} &\text{maximize} \\ &\quad 2^{n-1}x_1 + 2^{n-2}x_2 + \cdots + 2x_{n-1} + x_n \\ &\text{subject to} \quad x_1 \le 5 \\ &\quad 4x_1 + x_2 \le 25 \\ &\quad 8x_1 + 4x_2 + x_3 \le 125 \\ &\quad \vdots \\ &\quad 2^n x_1 + 2^{n-1}x_2 + \cdots + 4x_{n-1} + x_n \le 5^n \\ &\quad x_1, \dots, x_n \ge 0 \end{aligned}$$

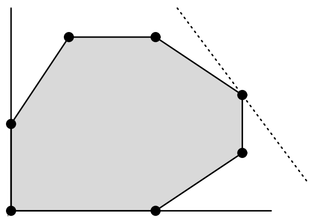
This can be shown to require $2^n - 1$ iterations if we start at $x_1 = \cdots = x_n = 0$ and use the rule of always choosing the largest number in the bottom row to select the pivot column. This example shows that the simplex algorithm can have exponential running-time in the worst case.

In fact, the optimal solution is $x_1 = \cdots = x_{n-1} = 0$, $x_n = 5^n$, and this can be reached in just one pivot operation from an initial basic feasible solution of $x_1 = \cdots = x_{n-1} = x_n = 0$ by pivoting in the column with the *smallest* entry in the bottom row!

The Hirsch Conjecture

There is the unresolved conjecture that for an LP with m constraints (in addition to the $x \geq 0$ constraints) an optimal vertex is always connected to any other vertex by at most m edges.

This is clearly true if $n = 2$. For example, here $m = 5$ and no vertex is more than 3 pivot steps from an initial b.f.s. of $(0, 0)$.



If the conjecture is true, then there always exists some way to reach the optimum in at most m pivots.

Example: Project assignment in CUED

Third year Engineering students are required to do 2 projects during the Easter Term, out of a choice of 32 projects. The process of assigning projects to students is complicated. There are timetable constraints and there is a limit on the number of students that can do each project. The problem is solved by asking students to allocate preference scores for projects. The instructions are:

You should indicate your preferences for exactly eight projects by assigning scores to eight projects that satisfy the following rules

1. *Your eight scores should be precisely the numbers 4,4,3,3,2,2,1,1. A 4 indicates a project for which you have the strongest preference. You may score two projects with 4, two with 3, and so on.*
2. *No two projects in the same category should be given the same score ...*
3. *You may place a score against Fi1 only if ...*
4. *European projects ...*

Suppose there are n students. Let us define $32n$ variables of the form x_{ij} where

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases} \text{if student } i \begin{array}{c} \text{is} \\ \text{is not} \end{array} \text{ assigned project } j.$$

Suppose student i assigns a score of a_{ij} to project j. We seek to solve an optimization problem with objective:

$$\texttt{maximize} \sum_{i=1}^{n} \sum_{j=1}^{32} a_{ij} x_{ij},$$

over $x_{ij} \in \{0, 1\}$, subject to the following additional constraints.

(a) No more that c_j students are assigned to project j. This is expressed

$$\sum_{i=1}^{n} x_{ij} \leq c_j, \quad \text{for all } j = 1, \ldots, 32.$$

Here we have 32 constraints.

(b) Each student does exactly 2 projects:

$$\sum_{j=1}^{32} x_{ij} = 2, \quad \text{for all } i = 1, \ldots, n.$$

Here we have n constraints.

(c) Students are not assigned a pair of projects timetabled at the same time. Consideration of Table 2 shows that this can be written as,

$$\begin{aligned} \sum_{j=1}^{4} x_{ij} + \sum_{j=15}^{17} x_{ij} &\leq 1, \\ \sum_{j=5}^{7} x_{ij} + \sum_{j=18}^{20} x_{ij} + \sum_{j=31}^{32} x_{ij} &\leq 1, \\ \sum_{j=8}^{10} x_{ij} + \sum_{j=21}^{23} x_{ij} + \sum_{j=31}^{32} x_{ij} &\leq 1, \\ \sum_{j=11}^{14} x_{ij} + \sum_{j=24}^{26} x_{ij} &\leq 1, \\ \sum_{j=27}^{30} x_{ij} + \sum_{j=31}^{32} x_{ij} &\leq 1, \end{aligned}$$

for all $i = 1, \ldots, n$. This is system of $5n$ constraints.

(d) Students are not assigned more that one computer project or one design project.

$$\sum_{j=1}^{14} x_{ij} \le 1, \quad \text{for all } i = 1, \dots, n,$$
$$\sum_{j=15}^{26} x_{ij} \le 1, \quad \text{for all } i = 1, \dots, n.$$

Here we have $2n$ constraints.

Thus the problem has a total of $32n$ variables and $8n + 32$ constraints, with the additional constraint that each $x_{ij} \in \{0, 1\}$. For $n = 250$ we have a integer LP with 8,000 variables and 2,032 constraints.

This is a large ILP, but the relaxed LP version, in which we require only $0 \le x_{ij} \le 1$ can be solved. In fact, this gives an integer solution. (Can you see why it must?) The method provides a very satisfactory allocation of projects to students. A computer package called LPSOLVE is used to do this, which runs in about 1 minute.

Results:

The bottom line is that once all the students have responded it is possible to have the allocations ready to hang in CUED's foyer in less than 4 minutes. All 248 students were assigned either their first and second choices.

assigned ranking	weights		
combinations	4,3,2,1	8,6,2,1	20,6,2,1
1st 1st	161	158	162
1st 2nd	75	81	74
2nd 2nd	10	9	9
1st 3rd	2		1
2nd 3rd			1
1st 4th			1

It was decided to use the weights 8,6,2,1.

Comparison of final tableaus for P and D

final tableau for P:

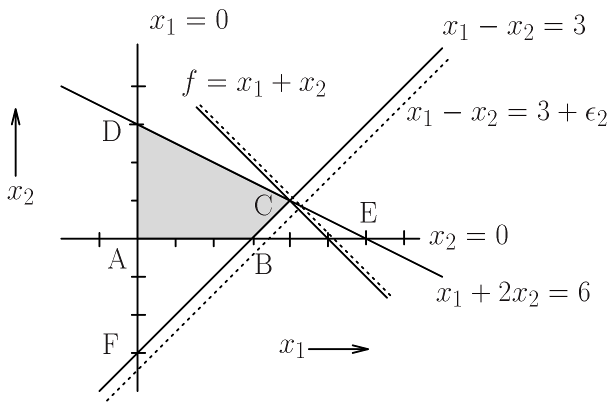
$$\begin{array}{cccc|c} x_1 & x_2 & z_1 & z_2 & \\ \hline 0 & 1 & \frac{1}{3} & -\frac{1}{3} & 1 \\ 1 & 0 & \frac{1}{3} & \frac{2}{3} & 4 \\ \hline 0 & 0 & -\frac{2}{3} & -\frac{1}{3} & -5 \end{array}$$

final tableau for D:

$$\begin{array}{cccc|c} \lambda_1 & \lambda_2 & v_1 & v_2 & \\ \hline 0 & 1 & -\frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \\ \hline 0 & 0 & -4 & -1 & 5 \end{array}$$

Note the positions of the primal and dual variables. Given a free choice of problems, either the primal or dual may be the easier one to solve. Once we have a final tableau for either problem we can read off the solutions to both.

Effect of perturbing a constraint



The final tableau is

$$\begin{array}{cccc|c} 0 & 1 & \frac{1}{3} & -\frac{1}{3} & 1 - \frac{1}{3}\epsilon_2 \\ 1 & 0 & \frac{1}{3} & \frac{2}{3} & 4 + \frac{2}{3}\epsilon_2 \\ \hline 0 & 0 & -\frac{2}{3} & -\frac{1}{3} & -5 - \frac{1}{3}\epsilon_2 \end{array}$$

We retain the same basis provided $-6 \le \epsilon_2 \le 3$.

Some practical examples of LP

Military logistics planning

The problem is concerned with the feasibility of supporting military operations overseas during a crisis.

The aim is to determine if materials can be transported overseas within strict time windows. The LP includes capacities at embarkation ports, capacities of the various aircraft and ships that carry the movement requirements and penalties for missing delivery dates. One problem that has been solved resulted in an LP with 20,500 constraints and 520,000 variables (solved in 75 minutes on a mini-supercomputer.)

Yield management at American Airlines

Critical to an airline's operation is the effective use of its reservation inventory. American Airlines has developed a series of OR models that effectively reduce the large problem to three smaller problems: overbooking, discount allocation and traffic management. They estimate the quantifiable benefit at \$1.4 billion over the last three years and expect annual revenue contribution of over \$500 million.

How large an LP can we solve?

The key factor is the number of constraints.

A typical workstation/mainframe LP package (e.g., OSL) has the capacity of 2 billion variables and 16 million constraints. However, this overstates what can be done in practice.

Code	Number of constraints	Number of variables	Time	Computer
OSL	105,000	155,000	240 mins	IBM 3090
	750	12,000,000	27 mins	IBM 3090
OB1	10,000	233,000	12 mins	IBM 3090
Cplex	145	1,000,000	6 mins	Cray Y-MP
	41,000	79,000	3 mins	Cray 2

OSL = Optimization Subroutine Library from IBM
OB1 (Roy Marsten, Georgia Institute of Technology)
CPlex (Bob Bixby, Rice University)

Example: On-line bin packing

An infinite sequence of items are to be packed into bins of size 5. Each successive item is equally likely to be of sizes 1, 2 or 3. One possible packing algorithm is *Best Fit* (BF), which puts each item into the smallest gap into which it will fit amongst the existing gaps in partially full bins, or if there is no gap large enough, the item goes into an empty bin.

The state at time t is written $x = (x_1, x_2, x_3, x_4)^T$ where x_i is the number of partially full bins with a gap of size i. Then, for example,

$$(0,4,0,0) \longrightarrow \begin{array}{lcll} (1,3,0,0) & & 1 & \frac{1}{3} \\ (0,3,0,0) & \text{on arrival of a} & 2 \text{ w.p.} & \frac{1}{3} \\ (0,5,0,0) & & 3 & \frac{1}{3} \end{array}$$

Let $d(x) = E[x(t+1) - x(t) \mid x(t) = x]$, be the *expected drift*, e.g., $d(0,4,0,0) = \left(\frac{1}{3}, -\frac{1}{3}, 0, 0\right)^T$.

Problem: Let $E|x(t)| = x_1 + x_2 + x_3 + x_4$ be the number of partially full bins. Does $E|x(t)| \to \infty$ or is $E|x(t)|$ bounded as $t \to \infty$?

Practical considerations suggest that it is better if $E|x(t)|$ is bounded than if $E|x(t)| \to \infty$.

Theorem *$E|x(t)|$ is bounded as $t \to \infty \iff \exists$ $\delta < 0$ and potential function $\phi(x) \geq 0$ such that*

$$E[\phi(x(t+1)) - \phi(x) \mid x(t) = x] \leq \delta < 0 \textit{ for all } x.$$

This theorem makes sense because it says that from every possible starting state the value of the potential function is drifting (on average) towards 0.

Suppose we try $\phi(x) = \sum_{i=1}^{4} a_i x_i$ and hunt for a_i that will work by considering the LP

$$\begin{aligned} &\text{minimize} \quad \delta \\ &\text{subject to} \quad d(x)^T a \leq \delta, \quad \text{for all } x \\ &\qquad\qquad\quad a_1, a_2, a_3, a_4 \geq 0. \end{aligned}$$

If the optimal solution is $\delta < 0$ this proves that $E|x(t)|$ is bounded as $t \to \infty$.

This is what happens for our example. The LP has only 6 constraints.

For packing items of sizes $1, \ldots, 8$ into bins of size 14 a version of this method gives a LP with 415,953 constraints and takes 24 hours to construct and solve!

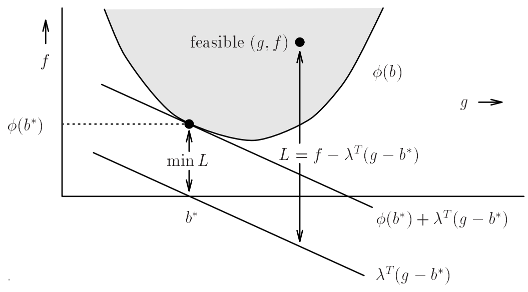
For packing $1, \ldots, 8$ into bins of size 11 the answer can be shown to be $E|x(t)| \to \infty$.

When Lagrangian methods work

Suppose f, g are convex, X is convex. Then

- $\phi(b) = \min\{f(x) : x \in X, g(x) = b\}$ is convex.
- $\exists$ a supporting hyperplane at b^*.
- The equation of this is $y = \phi(b^*) + \lambda^T(b - b^*)$.

We have the picture

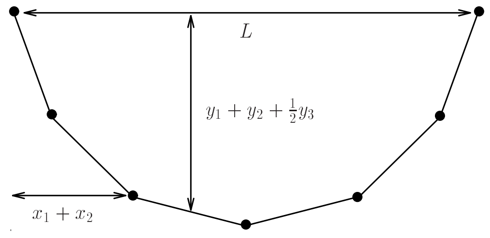


Notice that
$$\left.\frac{\partial \phi(b)}{\partial b}\right|_{b=b^*} = \lambda$$
.

Example: Hanging chain

A chain of n links, each of length 1, hangs between two points a distance L apart. To find the form in which the chain hangs we minimize the potential energy.

Let (x_i, y_i) be the displacement of the right hand end of the ith link from the right hand end of the $(i-1)$th link.



The potential energy is therefore
$\frac{1}{2}y_1 + (y_1 + \frac{1}{2}y_2) + \cdots + (y_1 + \cdots + y_{n-1} + \frac{1}{2}y_n)$.
We wish to minimize this subject to
$$\sum_{i=1}^n y_i = 0$$
$$\sum_{i=1}^n x_i = \sum_{i=1}^n \sqrt{1 - y_i^2} = L.$$

So the problem is

minimize $\sum_{i=1}^n (n - i + \frac{1}{2}) y_i$
subject to $\sum_{i=1}^n y_i = 0, \quad \sum_{i=1}^n \sqrt{1 - y_i^2} = L.$

Hence the Lagrangian is
$$L = \sum_{i=1}^n (n - i + \tfrac{1}{2}) y_i - \lambda \sum_{i=1}^n y_i - \mu \left(\sum_{i=1}^n \sqrt{1 - y_i^2} - L \right)$$
From $\partial L / \partial y_i = 0$ we find
$$y_i = \frac{-(n - i + \frac{1}{2} - \lambda)}{\sqrt{\mu^2 + (n - i + \frac{1}{2} - \lambda)^2}},$$
and λ and μ must be chosen to satisfy the two constraints.

Notice that the solution is not unique and that some stationary points give local minima that are not the global minimum. For example, consider a chain of 4 links which has local minimums in both V and W shapes.

Example: Airline luggage

On the Virgin Atlantic New York — London route the instructions given to economy class passengers read

> *2 pieces up to 32 kg/70 lbs maximum each piece. Dimensions when added together must not exceed 62" for one piece and 55" for the other.*

For a rectangular box with sides x_1,x_2,x_3 we

maximize $x_1 x_2 x_3$
subject to $x_1 + x_2 + x_3 \le c, \quad x_i \ge 0.$

The Lagrangian is
$$L = x_1 x_2 x_3 - \lambda(x_1 + x_2 + x_3 + z - c)$$
which has a maximum where $\lambda > 0$, $z = 0$ and
$$\frac{\partial L}{\partial x_i} = x_j x_k - \lambda = 0.$$
Hence the optimum occurs where $x_1 = x_2 = x_3 = c/3$ (as expected). $\lambda = c^2/9$.

In fact, this means we can carry 14,989 cubic inches, or about as much as 246 litres (if we use cube-shaped luggage)!

MacDiet (a LP with 30 variables, 19 constraints)

The LP is based on data for 22 MacDonald's foods.

Diet		1	2	3	4
Cost	£	1.30	4.83	5.23	5.99
Big Mac	.99	1.3	1		
Cheeseburger	.91				1
MacChicken sandwich	.99				1
Scambled eggs & muffin	.86			2	
French fries	.57		2	3	2
Tomato ketchup portion	.00	70.5	4	6	4
Banana milkshake	.90		3	2	2
Hot chocolate	.55				1
Orange juice	.60				1
Calories		2400	2308	2303	2303
Protein		55.0	59.8	66.2	63.8
Fat		41.4	76.2	76.3	75.4
Saturated fat		16.5	33.8	29.3	27.1
Carbohydrate		450.0	343.0	336.0	340.4
Sugars		326.2	204.4	148.2	190.1
Sodium		1.2	2.0	1.7	2.5
Fibre		4.9	15.3	22.0	19.1

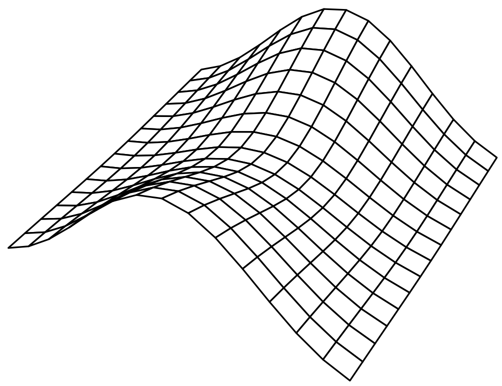
Diet 1. At least 2300 calories of which no more than 30% from fat. At least 55g protein and no more than 3g sodium.

Diet 2. Require an integer solution. No more than twice as many tomato ketcup portions as french fries.

Diet 3. No more than 2 banana milkshakes.

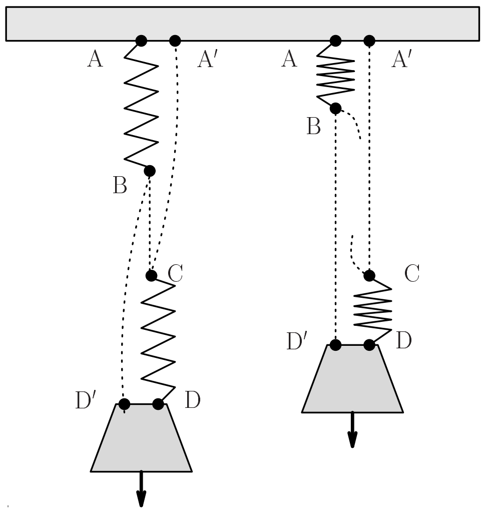
Diet 4. No more than 2 french fries and less than 64g protein.

Example: A saddle



$$\max_x \left\{ \min_y f(x,y) \right\} = \min_y \left\{ \max_x f(x,y) \right\}$$

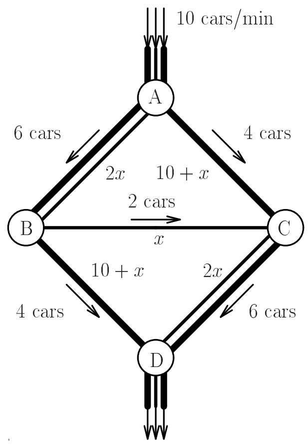
Springs puzzle



Two equal springs are stretched AB and CD. Dotted lines are strings. Strings A′C and BD′ are initially 'just slack'.

If string BC breaks the springs are now acting in parallel rather than in series. Each carries half the weight. They contract and the weight moves up (by one-half the distance AB).

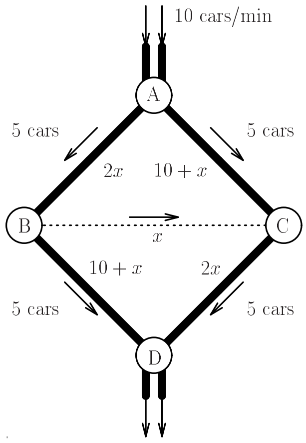
Braess's Paradox



The time to traverse links AB and CD is $2x$ minutes, where x is the number of cars carried per minute. On links AC and BD it is $10+x$ and on BC it is x.

A load of 10 cars/min splits into the equilibrium shown above. The journey along ABD and ACD takes $2(6)+10+(4)=26$ mins and the journey along ABCD also takes $2(6)+(2)+2(6)=26$ mins. No driver has any incentive to change his route.

Suppose BC is closed. The traffic readjusts to the equilibrium shown below.



The journey from A to D now takes $2(5)+10+(5) = 25$ mins.

Thus closing BC has reduced the journey time by 1 minute for all drivers!

If BC is now reopened the equilibrium is unstable. Some drivers will find it to their advantage to use BC and the equilibrium will readjust to the one in the first picture, in which the journey from A to D takes 26 minutes.

More practical examples of LP

Basin facility planning for AT&T

To determine where undersea cables and satellite circuits should be installed, when they will be needed, the number of circuits needed, cable technology, call routing, etc., over a 19 year planning horizon (an LP with 28,000 constraints and 77,000 variables.)

Military officer personnel planning

The problem is to plan US Army officer promotions (to Lieutenant, Captain, Major, Lieutenant Colonel and Colonel), taking into account the people entering and leaving the Army and training requirements by skill categories to meet the overall force structure (an LP with 21,000 constraints and 43,000 variables.)

Bond arbitrage

Many financial transactions can be modelled as LPs, e.g., bond arbitrage, switching amongst various financial instruments (bonds) so as to to make money. Typical problems have around1,000 constraints and 50,000 variables and can be solved in 'real-time'.

Example: The Rendezvous Problem

Two people have become separated. What can they do to minimize the expected time required to find one another? This is roughly the problem you face if you go to a shopping mall with a friend, become separated, and then want to find your friend in minimal time, knowing that he is also be hunting for you.

In a model of this, we think of each player as being in one of n rooms. At each discrete step a player can move to any of the rooms, or stay in the same room. Players have no common map of the rooms and everything looks symmetric. A player can only remember which rooms he has visited. They wish to meet in the minimal expected number of steps.

Optimal instruction manual

Each player has a copy of the *Hitchhikers' Guide to the Galaxy*, and looks up instructions what to do in this situation. What should these instructions be?

One possibility is the instruction *search at random*. Under this strategy the expected number of steps until the players meet is n.

Optimal instructions for $n = 3$

In the case $n = 3$ the optimal instructions are:

1. Move to at room a random (i.e., probability $1/3$ for each room).
2. If you have not met your friend then compare your present location with the location that you have just come from. If they are the same then stay there for the next step. If they differ then go to the third room, i.e., the one which you are not in now nor which you visited on the previous step.
3. If you have still not met your friend then restart the instructions at step 1.

Under this algorithm the expected number of steps to meet is $8/3$. This is less than the expected time of 3 which results under *search at random*.

There are many interesting unsolved questions:

- What are the optimal instructions for $n \geq 4$?
- Does the minimal expected time to meet increase in n?
- What are the optimal instructions if the players are allowed to leave notes behind in the rooms they have visited? What should the notes say?

Example: Optimal coding

Suppose words $w_1, \dots, w_m$ are to be coded into binary. Not all codes are decipherable. For example,

$$w_1 \longrightarrow 0 \qquad w_2 \longrightarrow 01 \qquad w_3 \longrightarrow 101$$

is no good because we wouldn't know whether 0101 is code for $w_1 w_3$ or $w_2 w_2$.

Suppose w_i is coded by a binary string of length s_i, where $1 \le s_i \le \bar{s}$. Then

$$\left(2^{-s_1} + 2^{-s_2} + \cdots + 2^{-s_m}\right)^n = \sum_{i=n}^{n\bar{s}} a_i 2^{-i},$$

where the coefficient a_i is the number of distinct ways that the binary codes for n words can be appended one after another to make a string which is i bits long. If these i-strings are to be uniquely decipherable, we must have no more than 2^i of them, so $a_i \le 2^i$. This gives

$$\left(2^{-s_1} + 2^{-s_2} + \cdots + 2^{-s_m}\right)^n \le n\bar{s}$$

for all n. Clearly $t^n < n\bar{s}$ for all n only if $|t| \le 1$. Hence we have the necessary condition

$$\boxed{2^{-s_1} + 2^{-s_2} + \cdots + 2^{-s_m} \le 1}$$

Now suppose word i occurs with probability p_i.

Problem: (minimize the mean code word length)

minimize $\sum_{i=1}^m p_i s_i$

subject to $\sum_{i=1}^m 2^{-s_i} \le 1$ and $s_i \in \{1, 2, \dots, \bar{s}\}$.

Solution:

We relax the constraint $s_i \in \{1, 2, \dots, \bar{s}\}$ to $s_i \ge 0$ and consider minimizing the Lagrangian, i.e.,

$$\min_{s_i, z \ge 0} \sum_{i=1}^m p_i s_i - \lambda \left(\sum_{i=1}^m 2^{-s_i} + z - 1 \right).$$

We need $\lambda < 0$ and $z^* = 0$. Differentiating with respect to s_i we have

$$p_i + \lambda (\log_e 2)\, 2^{-s_i^*} = 0.$$

The constraint $\sum_i 2^{-s_i^*} = 1$ implies $1 + \lambda^* \log_e 2 = 0$ and so $s_i^* = -\log_2 p_i$. The minimized value is

$$\sum_{i=1}^m p_i s_i^* = -\sum_{i=1}^m p_i \log_2 p_i \quad \text{(the source entropy)}.$$

This is a lower bound on the average word length under an optimal coding. In *Communication Theory* IIB we learn how to construct codes that (asymptotically) achieve this lower bound.

Example: Consumer behaviour

A consumer with £c to spend can purchase any of n goods at prices $p_1, \dots, p_n$. If the consumer purchases quantities $q_1, \dots, q_n$ her utility is $u(q_1, \dots, q_n)$. Hence she will seek to

maximize $u(q_1, \dots, q_n)$

subject to $\sum_i q_i p_i = c, \quad q_i \ge 0.$

The Lagrangian is

$$L(q, \lambda) = u(q_1, \dots, q_n) - \lambda \left(\sum_i q_i p_i - c \right)$$

which (under suitable conditions on u) has a maximum where

$$\frac{\partial L}{\partial q_i} = \frac{\partial u}{\partial q_i} - \lambda p_i = 0.$$

From this we derive the intuitively obvious result that at the optimum

$$\frac{1}{p_1}\frac{\partial u}{\partial q_1} = \cdots = \frac{1}{p_n}\frac{\partial u}{\partial q_n}.$$

In other words, at the optimum, an extra increment of spending increases the total utility by the same amount, no matter upon which good it is spent.

Option Trading

The owner of a

`British Airways 420 Jul put option`

has the right to sell one share of BA for 420p at anytime between today (3/5/95) and the option's expiry date at the end of July. Today's price of a BA share is 400p. So if exercised today, the owner can clear a profit of 20p (by buying one share on the market and then selling it at 420p). However, the quoted price for the option is 32p, which reflects the fact that the owner doesn't have to exercise the option today and can wait for BA shares to fall lower. The option has 12p (= 32 − 20p) of **time value**.

Optimization problem: How can we extract the optimal value out of ownership of an option?

Let $V_t(x)$ = market price for the option when a BA share costs x and there are t days until expiry.

One approach is **dynamic programming**:

$$V_t(x) = \max \Big[\; 420 - x \; , \; E_\Delta V_{t-\delta}(x(1+\Delta)) \; \Big]$$

$$V_0(x) = \max \Big[\; 420 - x \; , 0 \; \Big]$$

But this requires knowledge of the distribution of Δ, the random percentage change in x over time δ.

Hedging

An alternative approach is to create a **hedged portfolio**, by buying h BA shares to hold alongside the option. Suppose that over a time interval of length δ the price is equally likely to change by a factor $1 + \mu\delta + \sigma\sqrt{\delta}$ or $1 + \mu\delta - \sigma\sqrt{\delta}$.

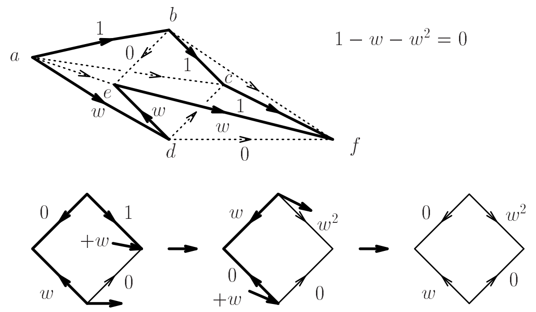
We can find $h > 0$ such that

$$V_{t-\delta}\left(x + \mu\delta x + \sigma\sqrt{\delta}x\right) + \left(1 + \mu\delta + \sigma\sqrt{\delta}\right)hx$$
$$=V_{t-\delta}\left(x + \mu\delta x - \sigma\sqrt{\delta}x\right) + \left(1 + \mu\delta - \sigma\sqrt{\delta}\right)hx.$$

In this case we don't care whether the BA price rises or falls. The resulting value of the portfolio is the same. In fact, equating the increase in the value of the portfolio to what the cost of the portfolio would have earned in a bank account implies what the value of $V_t(x)$ should be. In general, $V_t(x)$ and h will depend on x, t and σ (but not on μ). We must adjust h from moment to moment as x and t change. (See the IIA course *Stochastic Financial Models*.)

Notice that we both buy the put and *buy* an appropriate number of BA shares. In this case there is no risk. The 'Nick Leeson strategy' would be to buy puts and *short-sell* BA shares. Risky!

Example: Failure to stop when capacities and initial flows are not rational



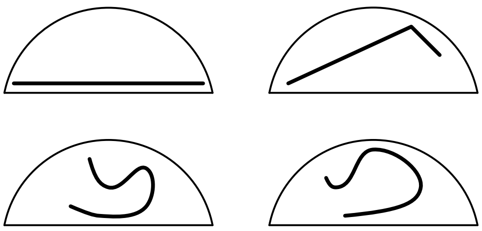
The network consists of a square $b\,c\,d\,e$ of directed arcs of capacity 1. The corners of the square are connected to a source at a and sink at f by arcs of capacity 10. The initial flow of $1 + w$ is shown in the top picture, where $w = (\sqrt{5} - 1)/2$, so $1 - w = w^2$. The first iteration is to increase flow by w along $a \to c \to b \to e \to d \to f$. The second increases it by w along $a \to d \to e \to b \to f$. The flow has increased by $2w$ and the resulting flow in the square is the same as at the start, but multiplied by w and rotated through 180^o. Hence the algorithm can continue in this manner forever without stopping and never reach the optimal flow of 40.

String covering

This problem is unsolved.

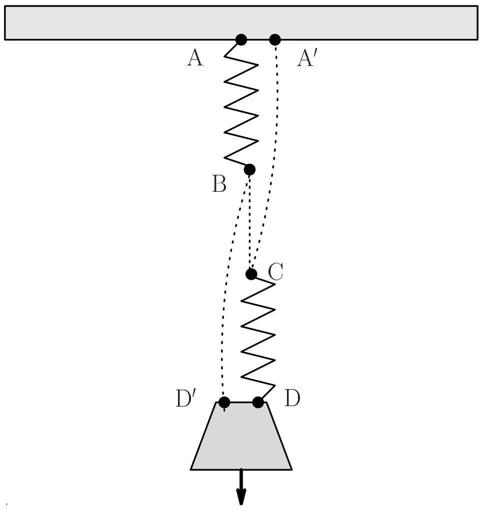
Optimization problem:

Find the shape of 'mat' of minimum area that can cover a piece of string of length 1 foot, no matter how this string is placed on the table.



We could use a semicircle with a base of length 1. But what is best?

Springs puzzle



Two equal springs are stretched AB and CD. Dotted lines are strings. Strings A′C and BD′ are initially 'just slack'.

What happens to the weight if string BC breaks?