All Meals for a Dollar and other vertex enumeration problems

David Avis

April 5, 2012

Hot news: Lovasz wins Kyoto Prize

INAMORI FOUNDATION http://www.inamori-f.or.jp/e_kp_lau_thi.html

稲盛財団
INAMORI FOUNDATION Home E-mail Back to English Top

Inamori Foundation | The Kyoto Prize
Inamori Foundation | Philosophy | About The Kyoto Prize | Laureates

The Kyoto prize / Laureates / The 2010 Kyoto Prize Laureates

Advanced Technology / Prize Field: Biotechnology and Medical Technology

Shinya Yamanaka

Japan / September 4, 1962
Medical Scientist
Professor, Kyoto University

| Profile | Achievements | Press page |

"Development of Technology for Generating Induced Pluripotent Stem (iPS) Cells"

By introducing just four transcription factor genes into dermal fibroblasts, Dr. Yamanaka succeeded in producing induced pluripotent stem (iPS) cells, which exhibit a pluripotency similar to that of embryonic stem (ES) cells. The iPS cell technology is now expected not only to expand the possibilities of regenerative medicine, but also to make significant contributions to the rapid progress of medical science in general.

Basic Sciences / Prize Field: Mathematical Sciences (including Pure Mathematics)

László Lovász

Hungary, U.S.A. / March 9, 1948
Mathematician
Professor, Eötvös Loránd University

| Profile | Achievements | Press page |

"Outstanding Contributions to Mathematical Sciences Based on Discrete Optimization Algorithms"

Through his advanced research on discrete structures, Dr. Lovász has provided a link among various branches of mathematics in terms of algorithms, thereby influencing a broad spectrum of the mathematical sciences - including discrete mathematics, combinational optimization and theoretical computer science. In so doing, Dr. Lovász has made outstanding contributions to the advancement of both the academic and technological possibilities of the mathematical sciences.

Arts and Philosophy / Prize Field: Arts (Painting, Sculpture, Craft, Architecture, Design)

Modelling and optimization

- Modelling refers to building an abstract mathematical model of real situation, typically one involving making decisions under constraints with a certain objective.

Modelling and optimization

- Modelling refers to building an abstract mathematical model of real situation, typically one involving making decisions under constraints with a certain objective.
- The decisions are modelled as decision variables

Modelling and optimization

- Modelling refers to building an abstract mathematical model of real situation, typically one involving making decisions under constraints with a certain objective.
- The decisions are modelled as decision variables
- The constraints and the objective are stated in terms of the decision variables

Modelling and optimization

- Modelling refers to building an abstract mathematical model of real situation, typically one involving making decisions under constraints with a certain objective.
- The decisions are modelled as decision variables
- The constraints and the objective are stated in terms of the decision variables
- If the constraints and objective are linear functions, it is called a linear program

Diet problem

- Situation: You need to choose some food in the supermarket to feed yourself properly for just \$1 per day.

Diet problem

- Situation: You need to choose some food in the supermarket to feed yourself properly for just $1 per day.
- Decison variables: How much of each product you will buy.

Diet problem

- Situation: You need to choose some food in the supermarket to feed yourself properly for just $1 per day.
- Decison variables: How much of each product you will buy.
- Constraints: There are minimum daily requirements for calories, vitamins, calcium, etc. There is a maximum amount of each food you can eat.

Diet problem

- Situation: You need to choose some food in the supermarket to feed yourself properly for just $1 per day.
- Decison variables: How much of each product you will buy.
- Constraints: There are minimum daily requirements for calories, vitamins, calcium, etc. There is a maximum amount of each food you can eat.
- Objective Eat for less than $1.

Sample data

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

The decision variables are $x_1, x_2, \ldots, x_6$.

Fractional servings are allowed.

From *Linear Programming*, Vasek Chvátal, 1983

Linear programming formulation for diet problem

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

$$
\begin{aligned}
\min z &= 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 \\
\text{s.t.} \quad & 110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 \geq 2000 \\
& 4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 \geq 55 \\
& 2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 \geq 800 \\
& 0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 3, \quad 0 \leq x_3 \leq 2, \\
& 0 \leq x_4 \leq 8, \quad 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 2
\end{aligned}
$$

General linear programming problem

$$
\begin{aligned}
max\ z \ = \ & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
s.t. \quad & a_{11}x_1 + a_{12}x_2 + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + a_{2n}x_n \leq b_2 \\
& \ldots \\
& a_{m1}x_1 + a_{m2}x_2 + a_{mn}x_n \leq b_m \\
& x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0
\end{aligned} \tag{1}
$$

General linear programming problem

$$
\begin{aligned}
max\ z = \ & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
s.t.\ \ & a_{11}x_1 + a_{12}x_2 + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + a_{2n}x_n \leq b_2 \\
& \ldots \\
& a_{m1}x_1 + a_{m2}x_2 + a_{mn}x_n \leq b_m \\
& x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0
\end{aligned} \tag{1}
$$

- $x_1,\ x_2,\ \ldots,\ x_n$ are the decision variables

General linear programming problem

$$
\begin{aligned}
max\ z \ = \ & c_1x_1 \ + \ c_2x_2 + \ldots + c_nx_n \\
s.t. \quad & a_{11}x_1 \ + \ a_{12}x_2 \ + \ a_{1n}x_n \ \leq \ b_1 \\
& a_{21}x_1 \ + \ a_{22}x_2 \ + \ a_{2n}x_n \ \leq \ b_2 \\
& \ldots \\
& a_{m1}x_1 \ + \ a_{m2}x_2 \ + \ a_{mn}x_n \ \leq \ b_m \\
& x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0
\end{aligned} \tag{1}
$$

- $x_1,\ x_2,\ \ldots,\ x_n$ are the decision variables
- $c_1,\ c_2,\ \ldots,\ c_n,\ b_1,\ b_2,\ \ldots, b_m$ and $a_{11},\ \ldots,\ a_{ij},\ \ldots, a_{mn}$ are input data

General linear programming problem

$$
\begin{aligned}
max\ z &= c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
s.t.\quad & a_{11}x_1 + a_{12}x_2 + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + a_{2n}x_n \leq b_2 \\
& \qquad \ldots \\
& a_{m1}x_1 + a_{m2}x_2 + a_{mn}x_n \leq b_m \\
& x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0
\end{aligned} \tag{1}
$$

- $x_1,\ x_2,\ \ldots,\ x_n$ are the decision variables
- $c_1,\ c_2,\ \ldots,\ c_n,\ b_1,\ b_2,\ \ldots, b_m$ and $a_{11},\ \ldots,\ a_{ij},\ \ldots, a_{mn}$ are input data
- The constraints (1) define a *convex polyhedron*

Simplex Method

- George Dantzig invented the simplex method to solve linear programs during WWII.

Simplex Method

- George Dantzig invented the simplex method to solve linear programs during WWII.
- "In terms of widespread application, Dantzig's algorithm is one of the most successful of all time: Linear programming dominates the world of industry..." (Top 10 Algorithms of the 20th century)

Simplex Method

- George Dantzig invented the simplex method to solve linear programs during WWII.
- "In terms of widespread application, Dantzig's algorithm is one of the most successful of all time: Linear programming dominates the world of industry..." (Top 10 Algorithms of the 20th century)
- It gave rise to the field of Operations Research (OR).

Operations Research faculty at Stanford (1969)

George Dantzig is on the far left, then Alan Manne, Frederick Hillier, Donald Iglehart, Arthur Veinott Jr., Rudolf E. Kalman, Gerald Lieberman, Kenneth Arrow and Richard Cottle.

Sensei and Seito

Vasek Chvátal

Another OR graduate from Stanford

Hatoyama file:///C:/cygwin/home/avis/talks/allmeals/hatoy

informs online
Institute for Operations Research and the Management Sciences

In The Media

PM with OR degree steps down

"Japan's former prime minister, Yukio Hatoyama, could not apply math modeling to solving two pressing political problems......."

"Before entering politics, Hatoyama in the 1970s received a Ph.D in engineering in a field called operations research, which employs applied mathematics to solve complex problems, at Stanford University."

Linear programming solution

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

Linear programming solution

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

- $x_1 = 4$(oatmeal) $x_4 = 4.5$(milk) $x_5 = 2$(pie) cost=92.5 ¢

Linear programming solution

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

- $x_1 = 4$(oatmeal) $x_4 = 4.5$(milk) $x_5 = 2$(pie) cost=92.5 ¢
- Where are the chicken, eggs and pork?

Linear programming solution

	Food	Serv. Size	Energy (kcal)	Protein (g)	Calcium (mg)	Price ¢	Max Serv.
x_1	Oatmeal	28g	110	4	2	3	4
x_2	Chicken	100g	205	32	12	24	3
x_3	Eggs	2 large	160	13	54	13	2
x_4	Milk	237ml	160	8	285	9	8
x_5	Cherry Pie	170g	420	4	22	20	2
x_6	Pork w. beans	260g	260	14	80	19	2
	Min. Daily Amt.		2000	55	800		

- $x_1 = 4$(oatmeal) $x_4 = 4.5$(milk) $x_5 = 2$(pie) cost=92.5 ¢
- Where are the chicken, eggs and pork?
- Do I have to eat the same food every day?

Problems with the solution

Problems with the solution

- Many desirable items were not included in the optimum solution

Problems with the solution

- Many desirable items were not included in the optimum solution
- We obtained a unique optimum solution, but ...

Problems with the solution

- Many desirable items were not included in the optimum solution
- We obtained a unique optimum solution, but ...
- ... people (and managers) like to make choices!

Ask the right question!

- Q: What are all the meals I can eat for at most \$1?

Ask the right question!

- Q: What are all the meals I can eat for at most $1?
- A: An infinite number! Add any small amount .....

Ask the right question!

- Q: What are all the meals I can eat for at most $1?
- A: An infinite number! Add any small amount .....
- Q: Can you give me some different meals at least?

Ask the right question!

- Q: What are all the meals I can eat for at most \$1?
- A: An infinite number! Add any small amount .....
- Q: Can you give me some different meals at least?
- A: Yes! In fact I can describe all allowable meals for under \$1

All meals for a dollar

Any solution to these inequalities is a meal for under $1:

$$\begin{aligned}
3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 &\leq 100 \\
110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 &\geq 2000 \\
4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 &\geq 55 \\
2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 &\geq 800 \\
0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 3, \quad 0 \leq x_3 \leq 2, & \\
0 \leq x_4 \leq 8, \quad 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 2 &
\end{aligned}$$

All meals for a dollar

Any solution to these inequalities is a meal for under \$1:

$$\begin{aligned}
3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 &\leq 100 \\
110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 &\geq 2000 \\
4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 &\geq 55 \\
2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 &\geq 800 \\
0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 3, \quad 0 \leq x_3 \leq 2, \\
0 \leq x_4 \leq 8, \quad 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 2
\end{aligned}$$

- But this is just a restatement of the problem ....

All meals for a dollar

Any solution to these inequalities is a meal for under \$1:

$$
\begin{aligned}
3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 &\leq 100 \\
110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 &\geq 2000 \\
4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 &\geq 55 \\
2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 &\geq 800 \\
0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 3, \quad 0 \leq x_3 \leq 2,& \\
0 \leq x_4 \leq 8, \quad 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 2&
\end{aligned}
$$

- But this is just a restatement of the problem ....
- ... how do I find these solutions?

A more useful solution

All menus for a $1

All (17) Extreme
Solutions to the Diet Problem with Budget $1.00

Cost	Oat-meal	Chicken	Eggs	Milk	Cherry Pie	Pork Beans
92.5	4.	0	0	4.5	2.	0
97.3	4.	0	0	8.	0.67	0
98.6	4.	0	0	2.23	2.	1.40
100.	1.65	0	0	6.12	2.	0
100.	2.81	0	0	8.	0.98	0
100.	3.74	0	0	2.20	2.	1.53
100.	4.	0	0	2.18	1.88	1.62
100.	4.	0	0	2.21	2.	1.48
100.	4.	0	0	5.33	2.	0
100.	4.	0	0	8.	0.42	0.40
100.	4.	0	0	8.	0.80	0
100.	4.	0	0.50	8.	0.48	0
100.	4.	0	1.88	2.63	2.	0
100.	4.	0.17	0	2.27	2.	1.24
100.	4.	0.19	0	8.	0.58	0
100.	4.	0.60	0	3.73	2.	0
100.	4.	0	1.03	2.21	2.	0.78

A more useful solution

All menus for a $1

All (17) Extreme
Solutions to the Diet Problem with Budget $1.00

Cost	Oat-meal	Chicken	Eggs	Milk	Cherry Pie	Pork Beans
92.5	4.	0	0	4.5	2.	0
97.3	4.	0	0	8.	0.67	0
98.6	4.	0	0	2.23	2.	1.40
100.	1.65	0	0	6.12	2.	0
100.	2.81	0	0	8.	0.98	0
100.	3.74	0	0	2.20	2.	1.53
100.	4.	0	0	2.18	1.88	1.62
100.	4.	0	0	2.21	2.	1.48
100.	4.	0	0	5.33	2.	0
100.	4.	0	0	8.	0.42	0.40
100.	4.	0	0	8.	0.80	0
100.	4.	0	0.50	8.	0.48	0
100.	4.	0	1.88	2.63	2.	0
100.	4.	0.17	0	2.27	2.	1.24
100.	4.	0.19	0	8.	0.58	0
100.	4.	0.60	0	3.73	2.	0
100.	4.	0	1.03	2.21	2.	0.78

A more useful solution

All menus for a $1

All (17) Extreme

Solutions to the Diet Problem with Budget $1.00

Cost	Oat-meal	Chicken	Eggs	Milk	Cherry Pie	Pork Beans
92.5	4.	0	0	4.5	2.	0
97.3	4.	0	0	8.	0.67	0
98.6	4.	0	0	2.23	2.	1.40
100.	1.65	0	0	6.12	2.	0
100.	2.81	0	0	8.	0.98	0
100.	3.74	0	0	2.20	2.	1.53
100.	4.	0	0	2.18	1.88	1.62
100.	4.	0	0	2.21	2.	1.48
100.	4.	0	0	5.33	2.	0
100.	4.	0	0	8.	0.42	0.40
100.	4.	0	0	8.	0.80	0
100.	4.	0	0.50	8.	0.48	0
100.	4.	0	1.88	2.63	2.	0
100.	4.	0.17	0	2.27	2.	1.24
100.	4.	0.19	0	8.	0.58	0
100.	4.	0.60	0	3.73	2.	0
100.	4.	0	1.03	2.21	2.	0.78

- Taking convex combinations of rows gives new meals

A more useful solution

All menus for a $1

All (17) Extreme
Solutions to the Diet Problem with Budget $1.00

Cost	Oat-meal	Chicken	Eggs	Milk	Cherry Pie	Pork Beans
92.5	4.	0	0	4.5	2.	0
97.3	4.	0	0	8.	0.67	0
98.6	4.	0	0	2.23	2.	1.40
100.	1.65	0	0	6.12	2.	0
100.	2.81	0	0	8.	0.98	0
100.	3.74	0	0	2.20	2.	1.53
100.	4.	0	0	2.18	1.88	1.62
100.	4.	0	0	2.21	2.	1.48
100.	4.	0	0	5.33	2.	0
100.	4.	0	0	8.	0.42	0.40
100.	4.	0	0	8.	0.80	0
100.	4.	0	0.50	8.	0.48	0
100.	4.	0	1.88	2.63	2.	0
100.	4.	0.17	0	2.27	2.	1.24
100.	4.	0.19	0	8.	0.58	0
100.	4.	0.60	0	3.73	2.	0
100.	4.	0	1.03	2.21	2.	0.78

- Taking convex combinations of rows gives new meals
- Eg. Taking half each of the last two rows gives a $1 meal with all foods

Two representations of a bounded polyhedron

Two representations of a bounded polyhedron

- H-representation (Half-spaces): $\{x \in R^n : Ax \leq b\}$

Two representations of a bounded polyhedron

- H-representation (Half-spaces): $\{x \in R^n : Ax \leq b\}$
- V-representation (Vertices): $v_1, v_2, ..., v_N$ are the vertices of P

$$x = \sum_{i=1}^{N} \lambda_i v_i$$

$$\text{where} \quad \sum_{i=1}^{N} \lambda_i = 1, \quad \lambda_i \geq 0, \; i = 1, 2, ..., N$$

Two representations of a bounded polyhedron

- H-representation (Half-spaces): $\{x \in R^n : Ax \leq b\}$
- V-representation (Vertices): $v_1, v_2, ..., v_N$ are the vertices of P

$$x = \sum_{i=1}^{N} \lambda_i v_i$$

$$\text{where} \quad \sum_{i=1}^{N} \lambda_i = 1, \quad \lambda_i \geq 0, \; i = 1, 2, ..., N$$

- Vertex enumeration: H-representation $\Rightarrow$ V-representation

Two representations of a bounded polyhedron

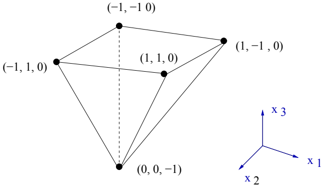
- H-representation (Half-spaces): $\{x \in R^n : Ax \le b\}$
- V-representation (Vertices): $v_1, v_2, ..., v_N$ are the vertices of P

$$x = \sum_{i=1}^{N} \lambda_i v_i$$

$$\text{where} \quad \sum_{i=1}^{N} \lambda_i = 1, \quad \lambda_i \ge 0, \ i = 1, 2, ..., N$$

- Vertex enumeration: H-representation $\Rightarrow$ V-representation
- Convex hull problem: V-representation $\Rightarrow$ H-representation

Example in R^3



H-representation:

$$\begin{aligned} 1 - x_1 \quad\quad + x_3 &\geq 0 \\ 1 \quad\quad - x_2 + x_3 &\geq 0 \\ 1 + x_1 \quad\quad + x_3 &\geq 0 \\ 1 \quad\quad + x_2 + x_3 &\geq 0 \\ - x_3 &\geq 0 \end{aligned}$$

V-representation:

$$v_1 = (-1,1,0), \quad v_2 = (-1,-1,0), \quad v_3 = (1,-1,0),$$

$$v_4 = (1,1,0), \quad v_5 = (0,0,-1)$$

Dictionary representation

Introduce slack variables:

$$\begin{aligned}
x_4 &= 1 - x_1 \quad + x_3 \geq 0\\
x_5 &= 1 \quad - x_2 + x_3 \geq 0\\
x_6 &= 1 + x_1 \quad + x_3 \geq 0\\
x_7 &= 1 \quad + x_2 + x_3 \geq 0\\
x_8 &= \quad - x_3 \geq 0
\end{aligned}$$

$B = \{4, 5, 6, 7, 8\}$, indices of *basic* variables.

$N = \{1, 2, 3\}$, indices of *co-basic* variables.

The dictionary with $N = \{4, 5, 8\}$ is *feasible* and represents the vertex $(1, 1, 0)$:

$$\begin{aligned}
x_1 &= 1 - x_4 \quad - x_8\\
x_2 &= 1 \quad - x_5 - x_8\\
x_3 &= \quad - x_8\\
\hline
x_6 &= 2 - x_4 \quad - 2x_8\\
x_7 &= 2 \quad - x_5 - 2x_8
\end{aligned}$$

Note: x_1, x_2, x_3 are basic and $x_6 \geq 0, \;\; x_7 \geq 0$

Adjacency: Pivoting

$$\begin{aligned} x_1 &= 1 - x_4 \quad\quad - x_8 \\ x_2 &= 1 \quad\quad - x_5 - x_8 \\ x_3 &= 0 \quad\quad\quad\quad - x_8 \\ \hline x_6 &= \mathbf{2} - x_4 \quad\quad - 2x_8 \\ x_7 &= \mathbf{2} \quad\quad - x_5 - 2x_8 \end{aligned}$$

B = { *1, 2, 3,* ***6, 7*** }

N = { ***4, 5 8*** }

Entering variable: ***4***

Leaving variable: ***6***

$$\begin{aligned} x_1 &= -1 \quad\quad + x_6 + x_8 \\ x_2 &= 1 - x_5 \quad\quad - x_8 \\ x_3 &= 0 \quad\quad\quad\quad - x_8 \\ \hline x_4 &= \mathbf{2} \quad\quad - x_6 - 2x_8 \\ x_7 &= \mathbf{2} - x_5 \quad\quad - 2x_8 \end{aligned}$$

B = { 1, 2, 3, ***4, 7*** }

N = { ***5, 6, 8*** }

x4 = 2 >= 0

x5 = 2 >= 0

Feasible Pivot!

Graph search for vertex enumeration

- Adjacency oracle for a feasible dictionary N:

$$Adj(N,i,j) = \begin{cases} \emptyset & N \cup \{i\}/\{j\} \text{ infeasible dictionary} \\ \emptyset & x_j \text{ has zero coefficient in row } x_i \\ N \cup \{i\}/\{j\} & \text{feasible dictionary} \end{cases}$$

Graph search for vertex enumeration

- Adjacency oracle for a feasible dictionary N:

$$Adj(N,i,j) = \begin{cases} \emptyset & N \cup \{i\}/\{j\} \text{ infeasible dictionary} \\ \emptyset & x_j \text{ has zero coefficient in row } x_i \\ N \cup \{i\}/\{j\} & \text{feasible dictionary} \end{cases}$$

- Find a starting feasible dictionary by solving an LP (Phase 1)

Graph search for vertex enumeration

- Adjacency oracle for a feasible dictionary N:

$$Adj(N,i,j) = \begin{cases} \emptyset & N \cup \{i\}/\{j\} \text{ infeasible dictionary} \\ \emptyset & x_j \text{ has zero coefficient in row } x_i \\ N \cup \{i\}/\{j\} & \text{feasible dictionary} \end{cases}$$

- Find a starting feasible dictionary by solving an LP (Phase 1)
- Use your favourite graph traversal algorithm to find all feasible dictionaries

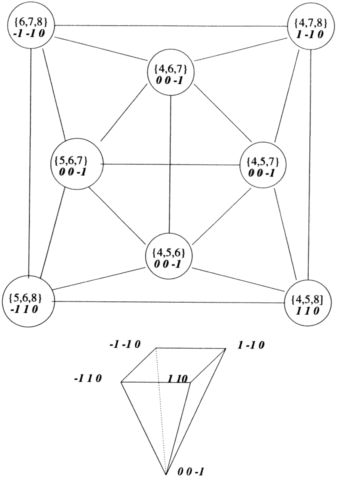
Graph search for vertex enumeration

- Adjacency oracle for a feasible dictionary N:

$$Adj(N,i,j) = \begin{cases} \emptyset & N \cup \{i\}/\{j\} \text{ infeasible dictionary} \\ \emptyset & x_j \text{ has zero coefficient in row } x_i \\ N \cup \{i\}/\{j\} & \text{feasible dictionary} \end{cases}$$

- Find a starting feasible dictionary by solving an LP (Phase 1)
- Use your favourite graph traversal algorithm to find all feasible dictionaries
- Eg. DFS or BFS

Graph of all feasible dictionaries



Problems with DFS/BFS algorithms

- DFS/BFS requires a database of visited vertices and a stack/queue

Problems with DFS/BFS algorithms

- DFS/BFS requires a database of visited vertices and a stack/queue
- A polyhedron with m inequalities and n dimensions may have $m^{\lfloor n/2 \rfloor}$ vertices !

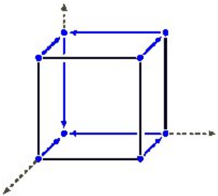
Problems with DFS/BFS algorithms

- DFS/BFS requires a database of visited vertices and a stack/queue
- A polyhedron with m inequalities and n dimensions may have $m^{\lfloor n/2 \rfloor}$ vertices !
- Often we just want to examine vertices, not keep them all

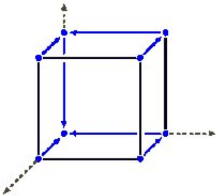
Problems with DFS/BFS algorithms

- DFS/BFS requires a database of visited vertices and a stack/queue
- A polyhedron with m inequalities and n dimensions may have $m^{\lfloor n/2 \rfloor}$ vertices !
- Often we just want to examine vertices, not keep them all
- Reverse search allows this to be done in $O(mn)$ space, ie. input size.

Reverse search algorithm Avis-Fukuda 1991

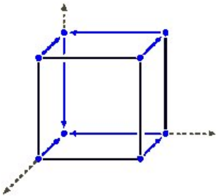


Reverse search algorithm Avis-Fukuda 1991



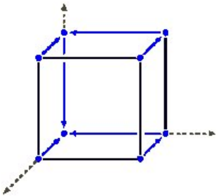
- Simplex method gives a path from any vertex to the optimum

Reverse search algorithm Avis-Fukuda 1991



- Simplex method gives a path from any vertex to the optimum
- The set of all such paths is a spanning tree of the polyhedron

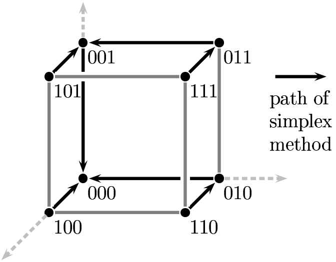
Reverse search algorithm Avis-Fukuda 1991



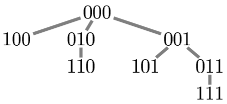
- Simplex method gives a path from any vertex to the optimum
- The set of all such paths is a spanning tree of the polyhedron
- Reverse search builds this tree starting at the origin, *reversing* the simplex method

Reverse search algorithm

http://cgm.cs.mcgill.ca/ avis/C/lrs.html



(a) The "simplex tree" induced by the objective $(-\sum x_i)$.



(b) The corresponding reverse search tree.