Prof. Friedrich Eisenbrand
Martin Niemeier

Due Date: March 25, 2010
Discussions: March 11, March 18

Discrete Optimization

Spring 2010

Solutions 2

You can hand in written solutions for up to two of the exercises marked with (*) or (Δ) to obtain bonus points. The duedate for this is March 25, 2010, before the exercise session starts. Math students are restricted to exercises marked with (*). Non-math students can choose between (*) and (Δ) exercises.

Exercise 1
A factory produces two different products. To create one unit of product 1, it needs one unit of raw material A and one unit of raw material B. To create one unit of product 2, it needs one units of raw material B and two units of raw material C. Raw material B needs preprocessing before it can be used, which takes one minute per unit. At most 20 hours of time is available per day for the preprocessing. Raw materials of capacity at most 1200 can be delivered to the factory per day. One unit of raw material A, B and C has size 4, 3 and 2 respectively.

At most 130 units of the first and 100 units of the second product can be sold per day. The first product sells for 6 CHF per unit and the second one for 9 CHF per unit.

Formulate the problem of maximizing turnover as a linear program in two variables.

Solution
We introduce 5 variables. Variables x and y model the amount of units of product one and two respectively that should be created. Variables a, b and c model the amount of raw material A, B and C needed.

One unit of product one sells for 6 CHF and one unit of product two sells for 9 CHF per unit. Thus the turnover is $6x+9y$. This is our objective function.

To create one unit of product one we need one unit of A. Thus $x \leq a$ has to hold for each feasible solution. For each unit of product one and two we need one unit of B. Thus $x+y \leq b$ must hold. Similarly we get the constraint $2y \leq b$.

Raw material B needs preprocessing that takes 1 minute per unit. At most 20 hours are available for preprocessing. This limits the maximum amount of b: $b \leq 60 \cdot 20 = 1200$.

The total size of raw materials we can have per day is limited by 1200. One unit of raw material A, B and C has size 4, 3 and 2 respectively. This gives the constraint $4a+3b+2c \leq 1200$.

At most 130 units of product 1 and 100 units of product two can be sold per day, thus $x \leq 130$ and $y \leq 100$ must hold. Finally all variables must be nonnegative since we cannot produce a negative amount of products or have a negative stock of raw material: $x, y, a, b, c \geq 0$.

This yields the following linear program:

$$
\begin{array}{rll}
\max & 6x+9y & \\
\text{subject to} & x & \le a \\
& x+y & \le b \\
& 2y & \le c \\
& b & \le 1200 \\
& 4a+3b+2c & \le 1200 \\
& x & \le 130 \\
& y & \le 100 \\
& x, y, a, b, c \ge 0 &
\end{array}
\tag{1}
$$

This model has 5 variables, and we will now eliminate three of them. Consider an optimal solution x^*, y^*, a^*, b^*, c^* of the linear program (1). Observe that if we decrease a^*, b^* and c^* such that the first three constraints are satisfies with equality, the solution stays feasible and the objective value does not change. Thus this modified solution is optimal as well.

This justifies that we consider the modified LP instead where the fist three inequalities are replaced by equalities, i.e.

$$
\begin{array}{rll}
\max & 6x+9y & \\
\text{subject to} & x & = a \\
& x+y & = b \\
& 2y & = c \\
& b & \le 1200 \\
& 4a+3b+2c & \le 1200 \\
& x & \le 130 \\
& y & \le 100 \\
& x, y, a, b, c & \ge 0
\end{array}
$$

By substituting a, b and c in the remaining inequalities we get a linear program with 2 variables

$$
\begin{array}{rll}
\max & 6x+9y & \\
\text{subject to} & 2y & \le 1200 \\
& 4x+3(x+y)+2\cdot 2y & \le 1200 \\
& x & \le 130 \\
& y & \le 100 \\
& x, y & \ge 0
\end{array}
$$

which we can simplify to

$$
\begin{array}{rll}
\max & 6x+9y & \\
\text{subject to} & 2y & \leq 1200 \\
& 7x+7y & \leq 1200 \\
& x & \leq 130 \\
& y & \leq 100 \\
& x,y & \geq 0
\end{array}
$$

Exercise 2

Consider a school district with I neighborhoods, J schools and G grades at each school. Each school j has a capacity of C_{jg} for grade g. In each neighborhood i, the student population of grade g is S_{ig}. Finally the distance of school j from neighborhood i is d_{ij}.

We want to solve the following problem: Assign all students to schools, in such a way that this assignment does not exceed the capacity of the schools. Moreover, the total distance travelled by all students should be minimal.

1. Try to formulate this as a linear program. What difficulties arise?
2. Download an instance for this problem here:

   `http://disopt.epfl.ch/webdav/site/disopt/shared/Opt2010/schools_inst.zmpl`

3. Model the problem using ZIMPL and solve the instance using an LP solver of your choice. What property has the optimal solution for the linear program? What does this tell you about an optimal solution for the problem?

Solution

1. Let x_{ijg} be the amount of students from neighborhood i of grade g travelling to school j. Then for an assignment of students to schools, the total distance travelled by all students is given as

   $$\sum_{i\in I}\sum_{j\in J}\sum_{g\in G} d_{ij}x_{ijg}.$$

   For a feasible assignment, every student of every neighborhood and grade must be assigned to a school, this gives the constraint

   $$\sum_{j\in J} x_{ijg} = S_{ig} \quad \forall i \in I, g \in G.$$

The number of students each school can take of the respective grades is bounded by C_{jg}, thus

$$\sum_{i\in I} x_{ijg} \leq C_{jg} \quad \forall j \in J, g \in G$$

must hold. Finally there can be no negative numbers of assignments: $x \geq 0$. This gives the following linear program:

$$\begin{array}{rll} \min & \sum_{i\in I}\sum_{j\in J}\sum_{g\in G} d_{ij}x_{ijg} & \\ \text{subject to} & \sum_{j\in J} x_{ijg} = S_{ig} & \forall i \in I, g \in G \\ & \sum_{i\in I} x_{ijg} \leq C_{jg} & \forall j \in J, g \in G \\ & x \geq 0 & \end{array}$$

2.
```
set neighborhoods := { 1 to 6 };
set grades := { 1 to 5 };
set schools := { 1 to 4 };

param nh_demands[ neighborhoods * grades ] :=
<1,1> 37, <1,2> 12, <1,3> 22, <1,4> 25, <1,5> 24,
<2,1> 11, <2,2> 7, <2,3> 34, <2,4> 19, <2,5> 21,
<3,1> 6, <3,2> 27, <3,3> 12, <3,4> 73, <3,5> 22,
<4,1> 11, <4,2> 24, <4,3> 49, <4,4> 13, <4,5> 35,
<5,1> 25 ,<5,2> 23, <5,3> 45, <5,4> 17, <5,5> 28,
<6,1> 16, <6,2> 3, <6,3> 17, <6,4> 22, <6,5> 13;

param school_capacities[ schools * grades] :=
<1,1> 32, <1,2> 40, <1,3> 60, <1,4> 53, <1,5> 34,
<2,1> 22, <2,2> 33, <2,3> 70, <2,4> 96, <2,5> 56,
<3,1> 15, <3,2> 15, <3,3> 20, <3,4> 44, <3,5> 12,
<4,1> 67, <4,2> 40, <4,3> 102, <4,4> 86, <4,5> 44;

param distances[ neighborhoods * schools ] :=
<1,1> 10, <1,2> 33, <1,3> 21, <1,4> 11,
<2,1> 15, <2,2> 23, <2,3> 33, <2,4> 25,
<3,1> 17, <3,2> 10, <3,3> 26, <3,4> 22,
<4,1> 30, <4,2> 7, <4,3> 11, <4,4> 16,
<5,1> 25 ,<5,2> 23, <5,3> 19, <5,4> 12,
<6,1> 27, <6,2> 22, <6,3> 11, <6,4> 23;


# The variables of the linear program
var assign[neighborhoods * schools * grades] >= 0 <= infinity;

# The objective function
minimize cost: (sum <i,j,g> in neighborhoods * schools * grades : distances[i,
```

```
# Assignment constraints
subto ass: forall<i,g> in neighborhoods * grades do
   (sum <j> in schools : assign[i,j,g]) == nh_demands[i,g];
# Capacity constraints
subto cap: forall<j,g> in schools * grades do
   (sum <i> in neighborhoods : assign[i,j,g]) <= school_capacities[j,g];
```

3. The observation is that the variable values for the optimal solution are integer. In fact, we can prove that for any instance to this problem, there is an integer optimal solution. The reason is that the matrix encoding the linear program is totally unimodular, because it is the node-edge incidence matrix of a bipartite graph. You will learn the theory behind this later in the course.

Exercise 3

A cone $C \subseteq \mathbb{R}^n$ is *pointed* if it does not contain a line: There are no vectors $x \in C$, $v \in \mathbb{R}^n$ such that $x + \lambda v \in C$ for all $\lambda \in \mathbb{R}$.

Prove the following variant of Carathéodory's theorem: Given some set $X \subseteq \mathbb{R}^n$, $|X| > n$ such that cone(X) is pointed. For any $x \in$ cone(X), there exist at least two different subsets $X_1, X_2 \subseteq X$ with $|X_1| = |X_2| = n$ such that $x \in$ cone$(X_1) \cap$ cone(X_2).

Solution

With Carathéodory's theorem we know that there is a subset $X_1 \subseteq X$, $|X_1| = n$ and $x \in$ cone(X_1). If the vectors of X_1 are linearly dependent, again with Carathéodory's theorem there is a set $X' \subset X$ with $x \in$ cone(X') and $|X'| = n-1$. Thus for some vector $u \in X \setminus X_1$, setting $X_2 = X' \cup \{u\}$ shows the claim.

Thus assume that the vectors of X_1 are linearly independent. Let $X_1 = \{x_1, \ldots, x_n\}$ and let $x = \sum_{i=1}^n \lambda_i x_i$ with $\lambda_i \geq 0$ for each i.

Let $u \in X \setminus X_1$. Since the vectors of X_1 are a basis of $\mathbb{R}^n$, there is a vector $d \in \mathbb{R}^n$ with $u = \sum_{i=1}^n d_i x_i$. Assume for the moment that $d_j > 0$ for some $j \in \{1, \ldots, n\}$. Let $\mu \geq 0$ be maximal such that $\lambda_i - \mu d_i \geq 0$ for all $i = 1, \ldots, n$. The maximum exists since $\mu \leq \lambda_j / d_j$.

Then

$$x = \sum \lambda_i x_i = \sum \lambda_i x_i - \mu u + \mu u = \sum (\lambda_i - \mu d_i) x_i + \mu u.$$

By construction, there is a $x_j \in X_1$ such that $\lambda_j - \mu d_j = 0$. Thus the above shows that x can be written as a conic combination of $X_2 := (X_1 \setminus \{x_j\}) \cup \{u\}$.

Recall that we assumed that $d_j > 0$ for some $j \in \{1, \ldots, n\}$. For the case $d \leq 0$, observe that $-u = \sum_{i=1}^n -d_i x_i \in$ cone(X). Thus both u and $-u$ are contained in cone(X). Hence $\lambda u \in$ cone(X) for all $\lambda \in \mathbb{R}$, in contradiction to the assumption that the cone is pointed.

Exercise 4

A polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ *contains a line*, if there exists a nonzero $v \in \mathbb{R}^n$ and an $x^* \in \mathbb{R}^n$ such that for all $\lambda \in \mathbb{R}$, the point $x^* + \lambda \cdot v \in P$. Show that a nonempty polyhedron P contains a line if and only if A does not have full column-rank.

Solution

Assume that P contains a line $\{x : x = x^* + \lambda \cdot v, \lambda \in \mathbb{R}\}$. We claim that for all rows a_i of A we have $a_i^T v = 0$. Assume for contradiction that $a_i^T v = \beta \neq 0$. Then we can choose $\lambda \in \mathbb{R}$ such that $a_i^T x^* + \lambda\beta > b_i$. Thus for $x := x^* + \lambda v$ we have $x \notin P$ because $a_i^T x > \beta$. This is a contradiction to the fact that P contains the line $\{x : x = x^* + \lambda \cdot v, \lambda \in \mathbb{R}\}$.

Thus we have $a_i^T v = 0$ for all rows a_i. Thus the kernel of A is not empty, and A does not have full column rank.

Conversely, if A does not have full column rank, let x^* be some feasible point of the polyhedron, and let v be a nonzero vector from the kernel of A. Then $x^* + \lambda \cdot v \in P$ for all $\lambda \in \mathbb{R}$. Hence P contains a line.

Exercise 5 (*)

Consider a linear program

$$\max\{c^T x : Ax \leq b\}.$$

Let B be a roof for the linear program.

1. Let $j \in B$, and consider the following system of equations:

$$\begin{aligned} a_j^T v &= -1 \\ a_i^T v &= 0 \quad \forall i \in B,\ i \neq j. \end{aligned} \tag{2}$$

   Prove the following statement: If x is a feasible solution for the *roof linear program* and v is a solution to system (2), then for all $\lambda > 0$, the vector $x + \lambda v$ is a feasible solution for the roof linear program as well.

2. Prove the following statement: The *vertex* of the root B is the *unique* optimal solution to the roof linear program if and only if c is a *strictly positive* conic combination of the vectors a_i of the roof.

Solution

Let A_B be the matrix composed of the rows of A corresponding to the roof B.

1. As x is feasible for the root linear program we have

$$A_B x \leq b_B.$$

   As v is a solution to (2), we have $A_B v \leq 0$. Now let $\lambda > 0$. We conclude

$$A_B(x + \lambda v) = A_B x + \lambda A_B v \leq b_B + 0 \leq b_B,$$

   which shows that $x + \lambda v$ is a feasible solution for the root linear program as well.

2. Since B is a roof, we know that there is a vector $y \geq 0$ with $c = A_B^T y$.

   We show the claim by proving the contraposition: We show that if $y_j = 0$ for some $j \in B$, then the vertex of the roof is not the unique optimal solution. Afterwards we show that if there are multiple optimal solutions, than $y_j = 0$ for some $j \in B$.

   To show the first part, assume that $y_j = 0$ for some $j \in B$. Let x^* be the vertex of the roof, i.e. $A_B x^* = b_B$. Let v be a solution of the system (2). As shown in the first part of the exercise, $x^* + v$ is a feasible solution to the roof LP. Observe that $y^T A v = 0$. The reason is that $a_i v = 0$ for all $i \neq j$, and $y_j = 0$. We conclude

   $$c^T(x^* + v) = y^T(A_B x^* + A_B v) = c^T x^* + \underbrace{y^T A_B v}_{=0} = c^T x^*.$$

   Since x^* is optimal, this shows that $x^* + v$ is another optimal solution.

   For the second part, assume that x' is another optimal solution to the roof LP other than the vertex solution x^*. Thus we have

   $$0 = c^T x^* - c^T x' = y^T(A_B x^* - A_B x') = y^T(b_B - A_B x').$$

   For the third equality we used the fact that x^* is the vertex of the roof. We know that $y \geq 0$. Moreover, since x' is a feasible solution we have $b_B - A_B x' \geq 0$ as well. Thus $y^T(b_B - A_B x')$ is a scalar product of nonnegative vectors of value 0. Thus for each $i \in B$, we have $y_i = 0$ or $b_i - a_i^T x' = 0$. Since x' is not a vertex solution, there is a $j \in B$ with $b_j > a_j^T x'$. We conclude that $y_j = 0$, which finishes the proof.

Exercise 6 (Δ)
Solve the following linear program using the simplex method:

$$\begin{array}{rll} \max & 6a + 9b + 2c & \\ \text{subject to} & a + 3b + c & \leq -4 \\ & b + c & \leq -1 \\ & 3a + 3b - c & \leq 1 \\ & a & \leq 0 \\ & b & \leq 0 \\ & c & \leq 0 \end{array}$$

Start with the roof given by the three last rows. For each iteration, give the vertex of the roof, the row which leaves the roof and the row which enters the roof (together with an argument why you can choose the rows as you did).

Solution

We start with the roof $\{4,5,6\}$ defined by the last three rows. The vertex solution is trivially $(a,b,c) = (0,0,0)$.

It violates the first constraint, thus we need to perform an iteration of the simplex algorithm. To perform step 3 of the algorithm, we need to compute a conic combination of the objective vector using the rows of the roof, i.e. we need to solve the system

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 6 \\ 9 \\ 2 \end{pmatrix}.$$

The trivial solution is $x = (6,9,2)$. Moreover we want a nontrivial linear combination of the zero vector, using the new row vector with coefficient 1 and the others using the old row vectors: We solve the system

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix} y = 0$$
$$y_4 = 1$$

The trivial solution is

$$y = (-1,-3,-1,1).$$

We need to determine the maximum λ such that $x + \lambda y \geq 0$. In this case it is $\lambda = 2$. We get $x' := x + 2y = (4,3,0,2)$. Thus 6 leaves the roof and 1 enters the roof: Our new roof is $\{1,4,5\}$. Its vertex solution can be computed solving the system

$$\begin{aligned} a + 3b + c &= -4 \\ a &= 0 \\ b &= 0 \end{aligned}$$

and is $(a,b,c) = (0,0,-4)$.

We observe that the third constraint is violated, thus we need another iteration of the simplex method. Again we need a conic combination of the objective vector using the rows of our roof. Note that the vector x' from the last iteration already gives such a conic combination. We set $x = (2,4,3)$.

We also need to solve the system

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 3 & 0 & 1 & 3 \\ 1 & 0 & 0 & -1 \end{pmatrix} y = 0$$
$$y_4 = 1$$

The solution is $y = (1,-4,-6,1)$. We need to determine the maximum λ such that $x + \lambda y \geq 0$. In this case it is $\lambda = \frac{1}{2}$. We get $x' := x + \frac{1}{2} y = (\frac{5}{2}, 2, 0, \frac{1}{2})$. Hence row 5 leaves the roof and 3 enters the roof. Our new roof is $\{1,3,4\}$.

Its vertex solution can be computed solving the system

$$\begin{aligned} a+3b+c &= -4 \\ 3a+3b-c &= 1 \\ a &= 0 \end{aligned}$$

and is $(a,b,c) = (0, -\frac{1}{2}, -\frac{5}{2})$.

Observe that no constraint is violated by this vertex solution, thus it is optimal. The objective value is $6 \cdot 0 - 9 \cdot \frac{1}{2} - 2 \cdot \frac{5}{2} = -\frac{19}{2}$.

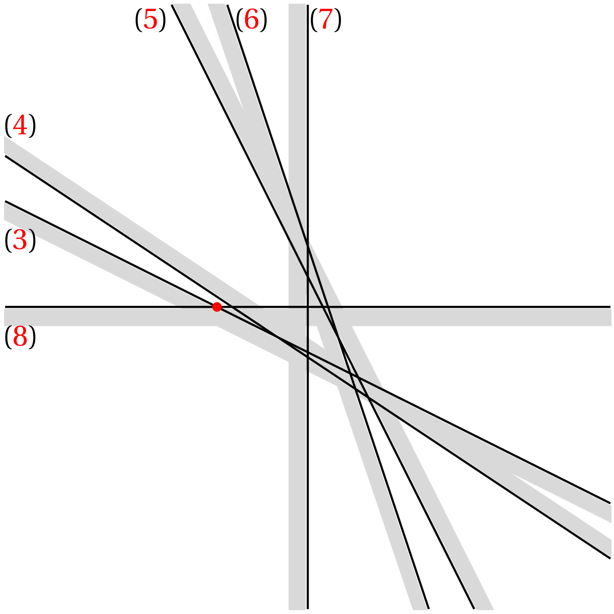
Exercise 7 (*)
Consider the problem

$$\begin{align} \max \quad & z \notag \\ \text{subject to} \quad & x+2y \le -3 \tag{3} \\ & -2x-3y \le 5 \tag{4} \\ & -2x-y+2z \le -1 \tag{5} \\ & 3x+y \le 2 \tag{6} \\ & x \le 0 \tag{7} \\ & y \le 0 \tag{8} \\ & z \le 0. \tag{9} \end{align}$$

Assume that we perform the simplex method, and at some point have the roof given by the rows (3), (8) and (9). The illustration below shows the situation in the 2-dimensional subspace given by the hyperplane $z = 0$.

Show that the simplex algorithm might not terminate, by giving a cycling sequence of roofs might be selected by the simplex method. Explain why your sequence is valid (it is sufficient to give drawings here, you do not need to compute the roof vertices explicitly).

Hint: Never let (9) *leave the roof. Then it is sufficient to consider the subspace as in the illustration.*

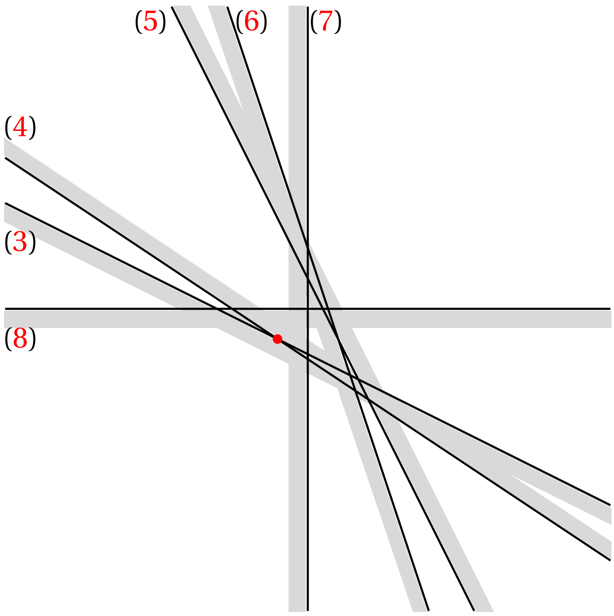


Solution

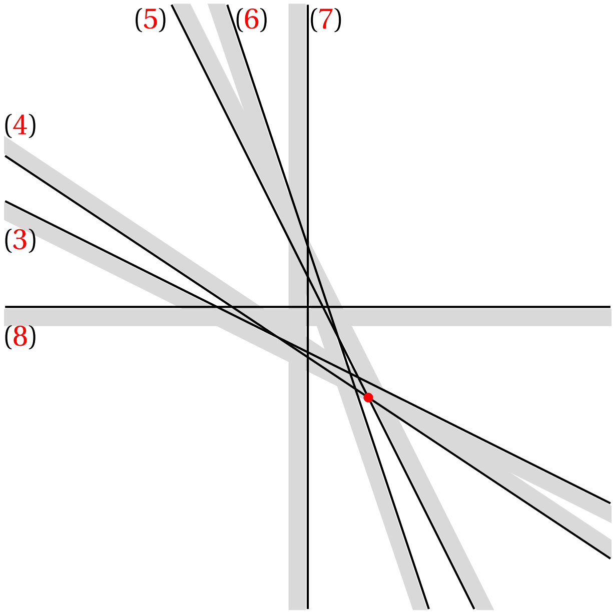
Recall that the high level idea for a simplex iteration is as follows: For each simplex iteration, we choose a constraint that violates the current vertex solution and bring it into the roof. We throw out another constraint in such a way that the new vertex solution is feasible for the old roof LP.

We show that the simplex algorithm might not terminate on this problem by giving a cycling sequence of roofs.

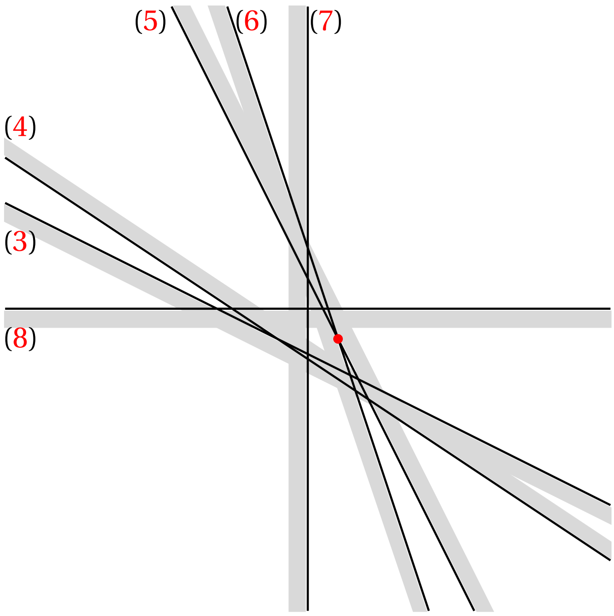
In the first iteration, we change the roof to $\{(3), (4), (9)\}$ and get the following vertex solution:



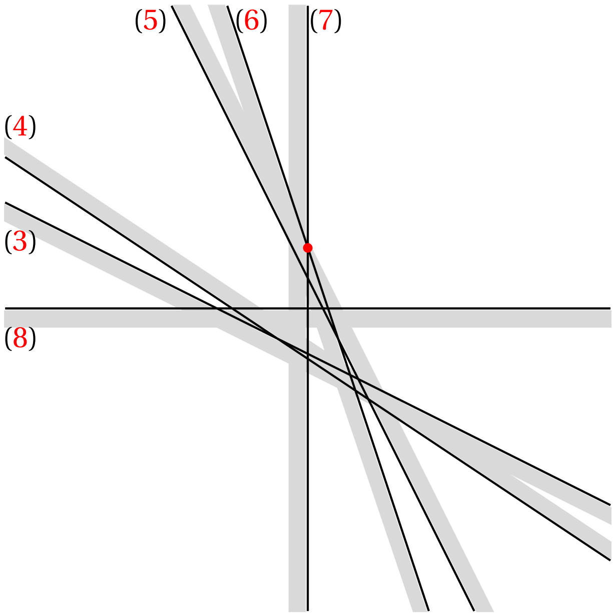
Next, we change the roof to $\{(4),(5),(9)\}$ and get the following vertex solution:



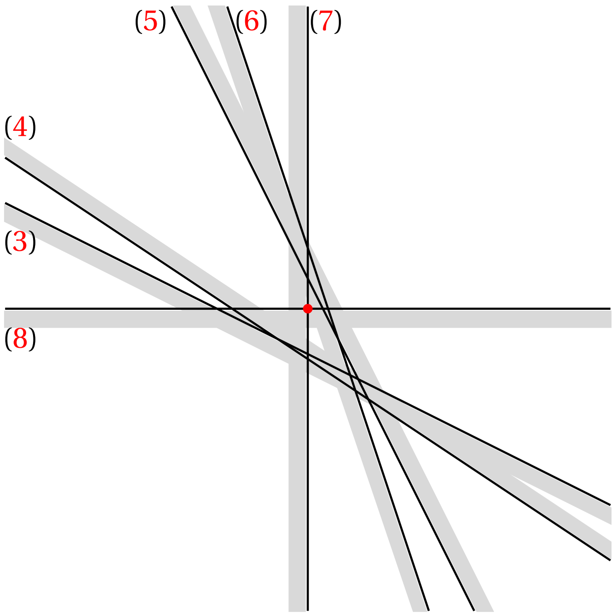
Next, we change the roof to $\{(5),(6),(9)\}$ and get the following vertex solution:



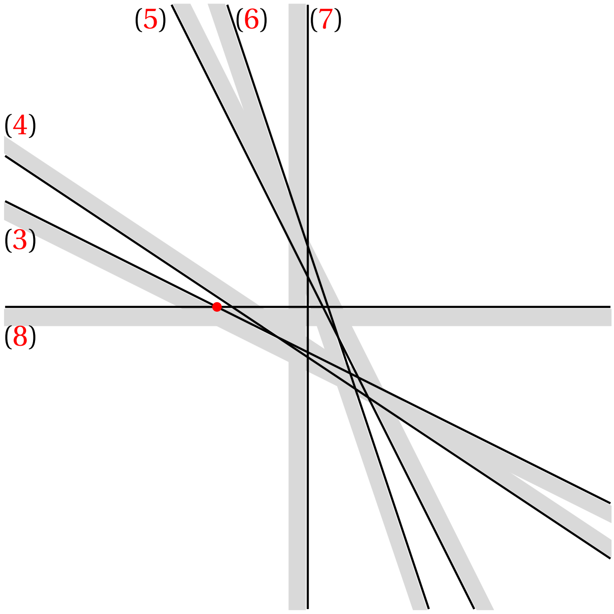
Next, we change the roof to $\{(6),(7),(9)\}$ and get the following vertex solution:



Next, we change the roof to {(7), (8), (9)} and get the following vertex solution:



Finally we change the roof to {(3), (8), (9)} and get the following vertex solution:



This is the roof we started with, and thus the simplex algorithm cycles.