ISE 426
Optimization models and applications

Lecture 13 — October 15, 2015

- MinMax
- Goal programming

- Winston & Venkataramanan, pages 191-194

Minimizing the maximum of a set of linear functions

Consider an optimization problem of the form

$$\begin{aligned} \min \quad \max\{ \quad & -2x_1 + 2, \\ & -x_1 + 1, \\ & x_1 - 3 \\ & 2x_1 - 7\}. \end{aligned}$$

Minimizing the maximum of a set of linear functions

Consider an optimization problem of the form

$$\begin{aligned} \min \quad \max\{ \quad & -2x_1 + 2, \\ & -x_1 + 1, \\ & x_1 - 3 \\ & 2x_1 - 7\}. \end{aligned}$$

- This is convex nonlinear problem
- How to reformulate it into a linear problem
- Create a new variable y
- y is the **maximum** of all quantities $-2x_1 + 2$, $-x_1 + 1$, $x_1 - 3$, $2x_1 - 7$.

Minimizing the maximum of a set of linear functions

In general, consider an optimization problem of the form

$$\begin{array}{lll} \min & \max_{k=1,2\ldots,H} & \left(\sum_{j=1}^{n} a_{kj}x_j + b_k\right) = \\ \\ \min & \max & \{\sum_{j=1}^{n} a_{1j}x_j + b_1, \\ & & \sum_{j=1}^{n} a_{2j}x_j + b_2, \\ & & \vdots \\ & & \sum_{j=1}^{n} a_{Hj}x_j + b_H\}. \end{array}$$

- This is nonlinear (there's a max term in the objective)[1].

[1] AMPL won't complain, but CPLEX will refuse to solve the problem.

Minimizing the maximum of a set of linear functions

In general, consider an optimization problem of the form

$$\min \quad \max_{k=1,2\ldots,H} \quad \left(\sum_{j=1}^n a_{kj}x_j + b_k\right) =$$

$$\min \quad \max \quad \begin{aligned}&\{\textstyle\sum_{j=1}^n a_{1j}x_j + b_1,\\ &\textstyle\sum_{j=1}^n a_{2j}x_j + b_2,\\ &\vdots\\ &\textstyle\sum_{j=1}^n a_{Hj}x_j + b_H\}.\end{aligned}$$

- This is nonlinear (there's a max term in the objective)[1].
- However, the model easily becomes linear:
- Create a new variable y
- y is the **maximum** of all quantities $\sum_{j=1}^n a_{1j}x_j + b_1$, $\sum_{j=1}^n a_{2j}x_j + b_2 \ldots, \sum_{j=1}^n a_{Hj}x_j + b_H$.

[1] AMPL won't complain, but CPLEX will refuse to solve the problem.

Minimizing the maximum of a set of linear functions

- Easy... if y is the maximum of all those quantities, then it must be greater than each of them:

$$
\begin{aligned}
y &\geq \textstyle\sum_{j=1}^{n} a_{1j}x_j + b_1, \\
y &\geq \textstyle\sum_{j=1}^{n} a_{2j}x_j + b_2, \\
&\vdots \\
y &\geq \textstyle\sum_{j=1}^{n} a_{Hj}x_j + b_H.
\end{aligned}
$$

- each of these constraints is linear!

  (re-write as $\sum_{j=1}^{n} a_{1j}x_j - y \leq -b_1, \ldots$)
- objective function is y. The linear model is:

$$
\begin{aligned}
\min \quad & y \\
& y \geq \textstyle\sum_{j=1}^{n} a_{kj}x_j + b_k \qquad \forall k = 1, 2 \ldots, H
\end{aligned}
$$

Why do we minimize y?

- The constraints above only say that y is **at least** the maximum of all those linear functions.

$\Rightarrow$ They don't guarantee that y is **exactly** the maximum of all those linear functions.

- That is,

$$y \geq \sum_{j=1}^{n} a_{kj}x_j + b_k \qquad \forall k = 1, 2 \ldots, H$$

only ensures that

$$y \quad \geq \quad \max_{k=1,2\ldots,H} \textstyle\sum_{j=1}^{n} a_{kj}x_j + b_k.$$

It does **not** ensure that

$$y \quad = \quad \max_{k=1,2\ldots,H} \textstyle\sum_{j=1}^{n} a_{kj}x_j + b_k.$$

Minimizing the maximum of a set of functions

However, this model works as we are minimizing y:

- Although for all feasible solutions $y \geq \max_{k=1,2\ldots,H} \sum_{j=1}^{n} a_{kj}x_j + b_k$,
- a solution $(\bar{x}_1, \bar{x}_2 \ldots, \bar{x}_n, \bar{y})$ with

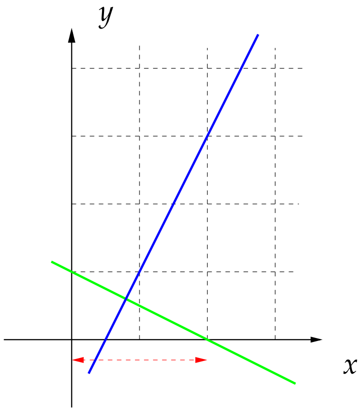
$$\bar{y} > \max_{k=1,2\ldots,H} \sum_{j=1}^{n} a_{kj}\bar{x}_j + b_k$$

  (strictly $>$) is feasible, but not optimal.
- Question: for an optimal solution $(\bar{x}_1, \bar{x}_2 \ldots, \bar{x}_n, \check{y})$ for how many k's $\check{y} = \sum_{j=1}^{n} a_{kj}\bar{x}_j + b_k$?

Another example

$$\min \quad \left(\max\{2x-1,\ -\tfrac{1}{2}x+1\}\right)$$
$$0 \le x \le 2$$

$$\begin{aligned} \min \quad & y \\ & y \ge 2x-1 \\ & y \ge -\tfrac{1}{2}x+1 \\ & 0 \le x \le 2 \end{aligned}$$



Example: bank loan

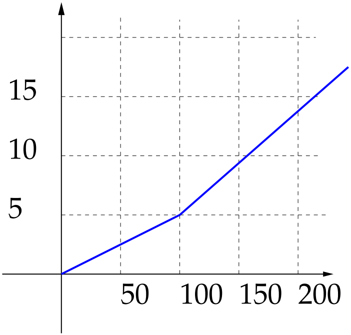
Our company wants to ask for a loan of 300k$ to two banks.
The interests paid to a bank depend on amount borrowed:

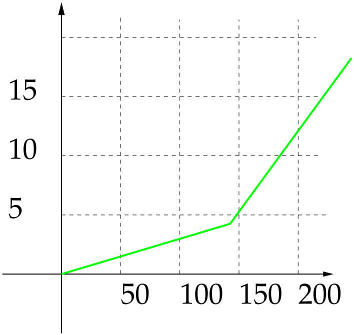
Bank 1:

- 5% of amt $\leq$ 100k$
- 8% of amt $\geq$ 100k$

Bank 2:

- 3% of amt $\leq$ 140k$
- 12% of amt $\geq$ 140k$





Example: bank loan

Determine how much to borrow from both banks in order to minimize the total interests paid.

- Variables: x_1 and x_2, amount borrowed from Bank 1 and Bank 2 (in k$)
- Constraints: $x_1 \geq 0$, $x_2 \geq 0$, and $x_1 + x_2 = 300$
- Objective function: sum of interests paid to the banks.

$$\min \; f_1(x_1) + f_2(x_2)$$

What are $f_1(x_1)$ and $f_2(x_2)$?

$f_1(x_1)$: $0.05 * x_1$ for $0 \leq x_1 \leq 100$,
$5 + 0.08 * (x_1 - 100)$ for $x_1 \geq 100$

$f_2(x_2)$: $0.03 * x_2$ for $0 \leq x_2 \leq 140$,
$4.2 + 0.12 * (x_2 - 140)$ for $x_2 \geq 140$

Example: bank loan

For this specific case[2], both $f_1(x_1)$ and $f_2(x_2)$ can be written as

$$
\begin{aligned}
f_1(x_1) &= \max\{0.05 * x_1,\ 5 + 0.08 * (x_1 - 100)\} \\
f_2(x_2) &= \max\{0.03 * x_2,\ 4.2 + 0.12 * (x_2 - 140)\}
\end{aligned}
$$

So the model is:

$$
\begin{array}{lll}
\min & & \max\{0.05 * x_1,\ 5 + 0.08 * (x_1 - 100)\} + \\
 & + & \max\{0.03 * x_2,\ 4.2 + 0.12 * (x_2 - 140)\} \\
 & & \\
 & & x_1 + x_2 = 300 \\
 & & x_1 \geq 0 \\
 & & x_2 \geq 0
\end{array}
$$

Nonlinear…

---

[2] both f_1 and f_2 are convex!

Example: bank loan

Linearization:

$$
\begin{aligned}
\min \quad & y_1 + y_2 \\
& y_1 \geq 0.05 * x_1 \\
& y_1 \geq 5 + 0.08 * (x_1 - 100) \\
& y_2 \geq 0.03 * x_2 \\
& y_2 \geq 4.2 + 0.12 * (x_2 - 140) \\
& x_1 + x_2 = 300 \\
& x_1 \geq 0 \\
& x_2 \geq 0
\end{aligned}
$$

Attaining minimum

- the auxillary variable “wants to” be at its lowest allowed value
- it should appear with a positive coefficient in a minimization problem or with a negative one in a max. problem

Symmetrically, it also works in “max-min” problems, that is, when maximizing the minimum of a set of functions
Caution! It does **not** work in general in other contexts, e.g.:

$$\max \max_{k=1,2\ldots,H}\left(\sum_{j=1}^{n} a_{kj}x_j + b_k\right)$$

or

$$\min \min_{k=1,2\ldots,H}\left(\sum_{j=1}^{n} a_{kj}x_j + b_k\right)$$

Example: job assignment

Problem:

- We have to assign m workers to m jobs. Everyone must be assigned to exactly one job, and all jobs have to be done.
- The degree of preference of a worker i to job j is defined by c_{ij}, for $i = 1, 2 \ldots, m$, $j = 1, 2 \ldots, m$.
- maximize the total preference, i.e. the sum of all preferences c_{ij} for assignments (i, j) worker-job.

Job assignment: model

Variables: x_{ij} for worker i and job j. Constraints:

- Every worker is assigned to **exactly** one job:

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i = 1, 2 \ldots, m$$

- Every job is done by **exactly** one worker:

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j = 1, 2 \ldots, m$$

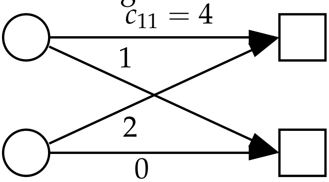
- Variables x_{ij} are binary (a yes/no decision)

Objective function: total preference

$$\sum_{i=1}^{m} \sum_{i=1}^{m} c_{ij} x_{ij}$$

Bad objective?

- the total preference $\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}x_{ij}$ does not provide a fair balance in assigning jobs: some worker may be very dissatisfied with their assignments.



- For a fair assignment, we may instead maximize the minimum assignment cost of each worker:
- How? The satisfaction of worker i is equal to $\sum_{j=1}^{m} c_{ij}x_{ij}$
- New objective function (still to be maximized):

$$\min_{i=1,2\ldots,m} \sum_{j=1}^{m} c_{ij}x_{ij}$$

- Look at least satisfied worker(s) (as it results from variables x_{ij}) and limit their dissatisfaction as much as possible

Job assignment: new model

$$\begin{array}{lll} \max & \min_{i=1,2\ldots,m} \sum_{j=1}^{m} c_{ij}x_{ij} & \\ & \sum_{j=1}^{m} x_{ij} = 1 & \forall i = 1,2\ldots,m \\ & \sum_{i=1}^{m} x_{ij} = 1 & \forall j = 1,2\ldots,m \\ & x_{ij} \in \{0,1\} & \forall i,j = 1,2\ldots,m \end{array}$$

It's nonlinear! Let's use the same trick, with different signs.

- New variable y (will be our objective function)
- y is $\min_{i=1,2\ldots,m} \sum_{j=1}^{m} c_{ij}x_{ij}$, for each $i = 1,2\ldots,m$.

$\Rightarrow$ y is smaller than **each** of these quantities:

$$\begin{array}{l} y \le \sum_{j=1}^{m} c_{1j}x_{1j} \\ y \le \sum_{j=1}^{m} c_{2j}x_{2j} \\ \vdots \\ y \le \sum_{j=1}^{m} c_{mj}x_{mj} \end{array}$$

- or, more compact: $y \le \sum_{j=1}^{m} c_{ij}x_{ij} \ \forall i = 1,2\ldots,m$

Job assignment: final linear model

$$
\begin{array}{lll}
\max & y & \\
& y \leq \sum_{j=1}^{m} c_{ij} x_{ij} & \forall i = 1, 2 \ldots, m \\
& \sum_{j=1}^{m} x_{ij} = 1 & \forall i = 1, 2 \ldots, m \\
& \sum_{i=1}^{m} x_{ij} = 1 & \forall j = 1, 2 \ldots, m \\
& x_{ij} \in \{0, 1\} & \forall i, j = 1, 2 \ldots, m
\end{array}
$$

Job assignment: alternative model

Let's reduce it to a minimization problem. The obj.f. changes sign, the problem becomes a minimization one:

$$\begin{aligned} & \max \quad \min_{i=1,2\ldots,m} \textstyle\sum_{j=1}^{m} c_{ij}x_{ij} = \\ = \quad & -\min \quad \left(-\min_{i=1,2\ldots,m} \textstyle\sum_{j=1}^{m} c_{ij}x_{ij}\right) = \end{aligned}$$

[apply the inverse rule inside the brackets. . .]

$$\begin{aligned} = \quad & -\min\left(\max_{i=1,2\ldots,m}\left(-\textstyle\sum_{j=1}^{m} c_{ij}x_{ij}\right)\right) = \\ = \quad & -\min\left(\max_{i=1,2\ldots,m} \textstyle\sum_{j=1}^{m}(-c_{ij})x_{ij}\right) \end{aligned}$$

Minimizing the maximum of absolute values

Consider now a system, of linear equations $Ax = b$, which does not have a solution. We want to find a solution x which violates the linear quations as little as possible. We can consider

$$\begin{array}{lll} \min & \max_{k=1,2\ldots,H} & (|\sum_{j=1}^{n} a_{kj}x_j - b_k| = \\ \\ \min & \max & \{|\sum_{j=1}^{n} a_{1j}x_j - b_1|, \\ & & |\sum_{j=1}^{n} a_{2j}x_j - b_2|, \\ & & \vdots \\ & & |\sum_{j=1}^{n} a_{Hj}x_j - b_H|\}. \end{array}$$

Minimizing the maximum of absolute values

Consider now a system, of linear equations $Ax = b$, which does not have a solution. We want to find a solution x which violates the linear quations as little as possible. We can consider

$$\begin{array}{lll} \min & \max_{k=1,2\ldots,H} & (|\sum_{j=1}^{n} a_{kj}x_j - b_k| = \\ & & \\ \min & \max & \{|\sum_{j=1}^{n} a_{1j}x_j - b_1|, \\ & & |\sum_{j=1}^{n} a_{2j}x_j - b_2|, \\ & & \vdots \\ & & |\sum_{j=1}^{n} a_{Hj}x_j - b_H|\}. \end{array}$$

$$\begin{array}{lll} \min & y & \\ & y \geq & \sum_{j=1}^{n} a_{kj}x_j - b_k \qquad \forall k = 1, 2 \ldots, H \\ & y \geq & -(\sum_{j=1}^{n} a_{kj}x_j - b_k) \qquad \forall k = 1, 2 \ldots, H \end{array}$$

The solution is likely to have n of the constraints from $Ax = b$ to have equal maximum violation. What if we do not like such a solution?

Minimizing the sum of absolute values

Consider now a system, of linear equations $Ax = b$, which does not have a solution. We want to find a solution x which violates the linear equations as little as possible in total. We can consider

$$\min \quad \sum_{k=1}^{H}(|\sum_{j=1}^{n} a_{kj}x_j - b_k| =$$

$$\begin{array}{ll}
\min & \sum_{k=1}^{H} |y_i| \\
& y_1 = \sum_{j=1}^{n} a_{1j}x_j - b_1, \\
& y_2 = \sum_{j=1}^{n} a_{2j}x_j - b_2, \\
& \vdots \\
& y_H = \sum_{j=1}^{n} a_{Hj}x_j - b_H.
\end{array}$$

Minimizing the sum of absolute values

Consider now a system, of linear equations $Ax = b$, which does not have a solution. We want to find a solution x which violates the linear equations as little as possible in total. We can consider

$$\min \quad \sum_{k=1}^{H} (|\sum_{j=1}^{n} a_{kj}x_j - b_k| =$$

$$\begin{aligned} \min \quad & \sum_{k=1}^{H} |y_i| \\ & y_1 = \sum_{j=1}^{n} a_{1j}x_j - b_1, \\ & y_2 = \sum_{j=1}^{n} a_{2j}x_j - b_2, \\ & \vdots \\ & y_H = \sum_{j=1}^{n} a_{Hj}x_j - b_H. \end{aligned}$$

Equivalent LP formulation

$$\begin{aligned} \min \quad & \sum_{k=1}^{H} (y'_k + y''_k) \\ & y'_k - y''_k = \quad \sum_{j=1}^{n} a_{kj}x_j - b_k \qquad \forall k = 1, 2 \ldots, H \end{aligned}$$

Goal programming

Consider the following management problem[3]:

- A company is introducing three new products, P1, P2, P3, and wants to know how many such products to make.
- The per-unit profit, per-unit employment level, and the per-unit capital investment for each product are as follows:

	P1	P2	P3
profit [M$]	12	9	15
employment lev. [$\times$100 wrks.]	5	3	4
capital inv. [M$]	4	7	8

The company:

- wants to have at least 125 M$ profit
- wants to keep its 4,000 employees, no more, no less
- wants to invest less than 55 M$

[3]Similar example on Winston & Venkataramanan, p. 191.

Optimization model

Variables: x_1, x_2, x_3.

Constraints:

- Profit: $12x_1 + 9x_2 + 15x_3 \geq 125$
- Employees: $5x_1 + 3x_2 + 4x_3 = 40$
- Investment: $5x_1 + 7x_2 + 8x_3 \leq 55$

Now we'd have to find an objective function.

However, a quick check with a dummy objective (e.g. x_1) tells us that the problem is infeasible: there is no value of (x_1, x_2, x_3) that satisfies **all** constraints.

Now who tells them the problem is infeasible?

The company now says that some constraints are **not** strict: they may be violated, but not too much.

They are **goals**: instead of focusing on one single objective function, try to make as many as possible to be satisfied as much as posible.

Previous estimations give the per-unit loss associated with violation of each constraint.

- 5 M$ per-unit loss for the long-run profit constraint ($<$125)

i.e. if we find a solution with $12x_1 + 9x_2 + 15x_3 = 122$, we'll incur losses for $5 \times (125 - 122) = 15$ M$

- 4 M$ per-unit loss when number of employees $<$40
- 2 M$ per-unit loss when number of employees $>$40
- 3 M$ per-unit loss when capital investment $>$55

Modify model: non-preemptive Goal Programming

- One or more constraint needs to be relaxed.
- Instead of ignoring them, penalize their **violation**:

$$\begin{array}{rrrcrrr} 12x_1 & +9x_2 & +15x_3 & \geq & 125 & & -y_1^- \\ 5x_1 & +3x_2 & +4x_3 & = & 40 & +y_2^+ & -y_2^- \\ 5x_1 & +7x_2 & +8x_3 & \leq & 55 & +y_3^+ & \end{array}$$

with $y_1^-, y_2^+, y_2^-, y_3^+ \geq 0$

- We'd like y_1^-, y_2^+, y_2^-, and y_3^+ to be all zero, but this is not possible as the problem would be infeasible.

⇒ try to make them as small as possible

Non-preemptive goal programming assumes all goals should be pursued (each with a weight).

Non-preemptive Goal Programming

$$
\begin{array}{lrrrrrrrr}
\min & 5y_1^- & +2y_2^+ & +4y_2^- & +3y_3^+ & & & & \\
\\
& 12x_1 & +9x_2 & +15x_3 & \geq & 125 & & -y_1^- \\
& 5x_1 & +3x_2 & +4x_3 & = & 40 & +y_2^+ & -y_2^- \\
& 5x_1 & +7x_2 & +8x_3 & \leq & 55 & +y_3^+ & \\
\end{array}
$$

$$y_1^-, y_2^+, y_2^-, y_3^+ \geq 0$$

- Result: $(x_1, x_2, x_3) = (\frac{25}{3}, 0, \frac{5}{3})$, and the only constraint being really relaxed is the second: $5x_1 + 3x_2 + 4x_3 = \frac{145}{3} = 48.333 > 40$.

i.e. $(y_1^-, y_2^+, y_2^-, y_3^+) = (0, 8.333, 0, 0)$

- Now at the company they start to think that maybe the second constraint **is** more important…

Preemptive Goal Programming

We still cannot satisfy all constraints, but we do prefer satisfying some rather than others.

Preemptive goal programming assumes some goals are more important than others, and satisfying the former should be a priority over the latter.

For the company, the main priorities are

- to preserve the total capital, and
- to keep employment level at most 40 (only one half of the second constraint), i.e. don't want to hire!

Once these are respected, we also care about the remaining two constraints:

- to do at least 125 M$ profit
- to keep employment level at least 40, i.e. don't want to fire

How do we model this?

Preemptive Goal Programming

For each **goal**, from most important to least important:

1. ignore (=relax) constraints at all lower levels
2. add penalization terms for this goal to objective function
3. solve
4. fix maximum violation of priorities at current level

Preemptive Goal Programming: stage 1

$$
\begin{array}{lrrrcrrr}
\min & 2y_2^+ & +3y_3^+ & & & & & \\
& 12x_1 & +9x_2 & +15x_3 & \geq & 125 & & -y_1^- \\
& 5x_1 & +3x_2 & +4x_3 & = & 40 & +y_2^+ & -y_2^- \\
& 5x_1 & +7x_2 & +8x_3 & \leq & 55 & +y_3^+ & \\
& & y_1^-, y_2^-, & y_2^+, y_3^+ & \geq & 0 & &
\end{array}
$$

- Only the violations of the more important constraints (y_2^+ and y_3^+) appear in the objective

$\Rightarrow$ The others don't, their constraints are **ignored** (=relaxed)

Result: $(x_1, x_2, x_3) = (0, 0, 0)$ (oops...), but we managed to satisfy both "important" constraints ($y_2^+ = y_3^+ = 0$).

The first constraint was relaxed, so it's easy to select x_i such that the other two are satisfied.

Preemptive Goal Programming: stage 2

$$
\begin{array}{rrrrcrrr}
\min & 5y_1^- & +2y_2^+ & +4y_2^- & +3y_3^+ & & & \\
& 12x_1 & +9x_2 & +15x_3 & \geq & 125 & & -y_1^- \\
& 5x_1 & +3x_2 & +4x_3 & = & 40 & +y_2^+ & -y_2^- \\
& 5x_1 & +7x_2 & +8x_3 & \leq & 55 & +y_3^+ & \\
& & & y_2^+ & = & 0 & & \\
& & & y_3^+ & = & 0 & & \\
& & & y_1^-, y_2^- & \geq & 0 & &
\end{array}
$$

- violation is fixed to 0 for the important constraints
- violation of the secondary constraints appear in the objective

Result: $(x_1, x_2, x_3) = (5, 0, 3.75)$, only the first constraint is violated (profit is $125 - y_1^- = 125 - 8.75 = 116.25$).

Example

The city council is developing an equitable city rate tax table. Taxes come from a combination of four sources:

- Property taxes: (\$550M base)
- Food & Drugs: (\$35M base)
- Other Sales: (\$55M base)
- Gasoline: (Consumption: 7.5 million gallons/year)

They would like to come up with a “fair” city tax...

- Tax revenues must be at least \$16M
- The property tax rate should be $\leq 1\%$.
- Food/drug taxes must be $\leq 10\%$ of all taxes collected
- Sales taxes must be $\leq 20\%$ of all taxes collected
- The gasoline tax must be $\leq \$0.02$/gallon.

Model

Variables:

r_p	Property tax rate	x_p	Property tax collected (in $)
r_f	Food/Drug tax rate	x_f	Food/Drug collected (in $)
r_s	Sales tax rate	x_s	Sales tax collected (in $)
r_g	Gas tax rate [$/gallon]	x_g	Gas tax collected (in $)
		T	Total taxes collected (in $)

Taxes (from rates) are based on the tax base

Model

Constraints (definition of x variables):

$$
\begin{aligned}
x_p &= 550r_p \\
x_f &= 35r_f \\
x_s &= 55r_s \\
x_g &= 7.5r_g \\
T &= x_p + x_f + x_s + x_g
\end{aligned}
$$

Requirements Constraints:

Revenue:	$T \geq 16$
Property Tax Rate:	$r_p \leq 0.01$
Food-Drug tax restriction:	$x_f \leq 0.1T$
Sales tax restriction:	$x_s \leq 0.2T$
Gas tax restriction:	$r_g \leq 0.02$

What's the objective?
It doesn't matter: The problem is infeasible!

Non-preemptive goal programming

Minimize the sum of violations altogether

$$
\begin{aligned}
\min \quad & e_p + e_f + e_s + e_g \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
& \\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p, e_f, e_s, e_g \geq 0
\end{aligned}
$$

Preemptive goal programming

Minimize each violation separately, in order. Suppose the order is (e_p, e_f, e_s, e_g). **Step 1**:

$$
\begin{aligned}
\min \quad & e_p \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
& \\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p, e_f, e_s, e_g \geq 0
\end{aligned}
$$

$\Rightarrow$ Result: $e_p = 0$

Preemptive goal programming: step 2

$$
\begin{aligned}
\min \quad & e_f \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
\\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p = 0, e_f, e_s, e_g \geq 0
\end{aligned}
$$

$\Rightarrow$ Result: $e_f = 0$

Preemptive goal programming: step 3

$$
\begin{aligned}
\min \quad & e_s \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
& \\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p = 0, e_f = 0, e_s, e_g \geq 0
\end{aligned}
$$

$\Rightarrow$ Result: $e_s = 0$

Preemptive goal programming: step 4

$$
\begin{array}{ll}
\min & e_g \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
& \\
& T \geq 16 \\
& r_p \leq 0.01 \qquad\qquad\qquad + e_p \\
& x_f \leq 0.1T \qquad\qquad\qquad + e_f \\
& x_s \leq 0.2T \qquad\qquad\qquad + e_s \\
& r_g \leq 0.02 \qquad\qquad\qquad + e_g \\
& e_p = 0, e_f = 0, e_s = 0, e_g \geq 0
\end{array}
$$

$\Rightarrow$ Result: $e_g = 0.74$

Min-max goal programming

Minimize the maximum violation

$$
\begin{aligned}
\min \quad & \max\{e_p, e_f, e_s, e_g\} \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
\\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p, e_f, e_s, e_g \geq 0
\end{aligned}
$$

Min-max goal programming

$$
\begin{aligned}
\min \quad & z \\
& z \geq e_p \\
& z \geq e_f \\
& z \geq e_s \\
& z \geq e_g \\
& x_p = 550 r_p \\
& x_f = 35 r_f \\
& x_s = 55 r_s \\
& x_g = 7.5 r_g \\
& T = x_p + x_f + x_s + x_g \\
& T \geq 16 \\
& r_p \leq 0.01 \qquad + e_p \\
& x_f \leq 0.1T \qquad + e_f \\
& x_s \leq 0.2T \qquad + e_s \\
& r_g \leq 0.02 \qquad + e_g \\
& e_p, e_f, e_s, e_g \geq 0
\end{aligned}
$$

Result: $e_p = e_f = e_s = e_g = 0.00991957$