Decision Making and Optimization

Master in Data Analytics for Business

Lisbon School of Economics & Management
Universidade de Lisboa

2025-2026

Multicriteria Analysis

Multicriteria decisions

Until now, a single criterion has been used to make better decisions: minimize cost, maximize profit, minimize time, maximize payoff

Techniques suitable for situations where the decision maker needs to consider multiple criteria to arrive at the best overall decision.

Usually multicriteria decision problems involve conflicting goals.

Multiobjective Linear Programming

General model

$\max f_1(x)$
$\max f_2(x)$
$\dots$
$\max f_p(x)$

subject to $x \in S$

$\max z_1 = \sum_{j=1}^{n} c_{1j} x_j$
$\max z_2 = \sum_{j=1}^{n} c_{2j} x_j$
$\dots$
$\max z_p = \sum_{j=1}^{n} c_{pj} x_j$

subject to $x \in S$ where
$S = \{x : Ax \geq b, x \geq 0\}$

or

$$\max\{Z = Cx : x \in S\}$$

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ & \cdots & \\ c_{p1} & \cdots & c_{pn} \end{bmatrix}$$

Example

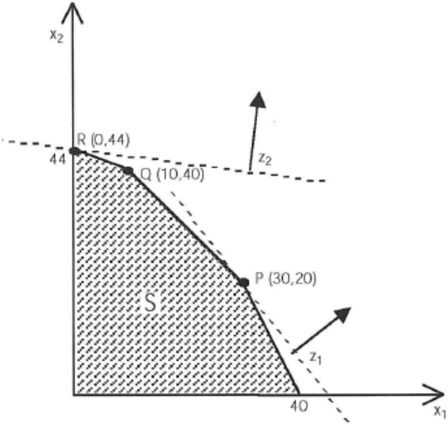
A small workshop produces two different products: P1 and P2. The manufacture of these products requires the use of 3 different types of machines: A, B and C. Each unit of P1 requires 1 hour on type A machines, 2 hours on type B machines and 2 hours on type C machines. Each unit of P2 requires 1 hour on type A machines, 1 hour on type B machines and 5 hours on type C machines. The workshop has several machines of these 3 types, which allow a maximum weekly use of 50 hours on the A machines, 80 hours on type B machines and 220 hours on type C machines. It is known that the profit from 1 unit of each of the products is 25 monetary units (m.u.) for P1 and 20 m.u. for P2, respectively. It is assumed that all production is sold. The reliability of the production system, as a function of the quantities produced of products P1 and P2, x_1 and x_2 respectively, is given by the function $z_2 = x_1 + 8x_2$. What should the weekly production be in order to maximize the resulting profit and maximize the reliability of the production system?

Example and pictures in

J.N. Clímaco, C.H. Antunes, M.J. Alves. "Programação Linear Multiobjectivo", Coimbra - Imprensa da Universidade. 2003.

Example

$$\begin{aligned} \max\ z_1 = &\ 25x_1 + 20x_2 \\ \max\ z_2 = &\ x_1 + 8x_2 \\ s.t. \quad &\ x_1 + x_2 \leq 50 \\ &\ 2x_1 + x_2 \leq 80 \\ &\ 2x_1 + 5x_2 \leq 220 \\ &\ x_1, x_2 \geq 0 \end{aligned}$$



there is no solution that simultaneously optimizes both objective functions:

$\max z_1$ is P

$\max z_2$ is R

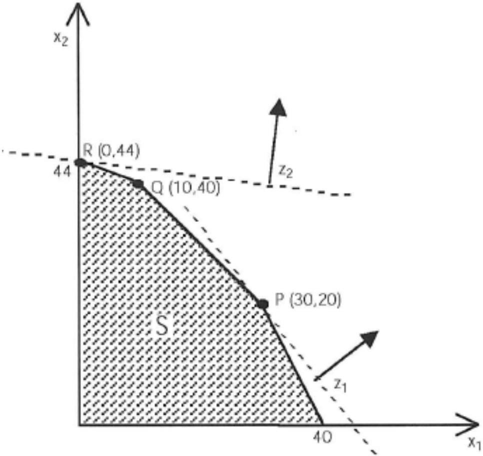
The two sets in multiobjective programming

In multiobjective programming we consider two spaces and sets:

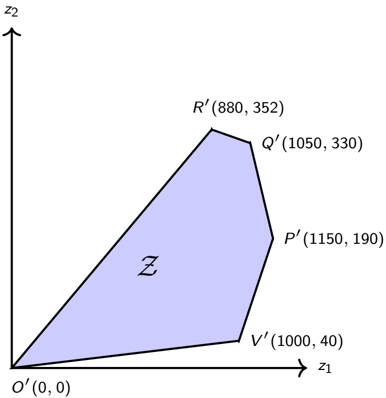
- in the space of the decision vectors, the feasible set or the set of feasible solutions, $x = (x_1, \ldots, x_n) \in S$
- in the space of the criterion vectors, the criterion set $z = (z_1, \ldots, z_p) \in Z$

Exercise. In the corresponding spaces, draw the feasible and the criterion sets of:
$S = conv(\{(0,0),(4,0),(4,1),(3,3),(0,4.5)\})$ and $c_1 = [1,0], c_2 = [3,2]$.

Example



in the variables space



in the criteria space

Efficiente or Pareto optimal solutions

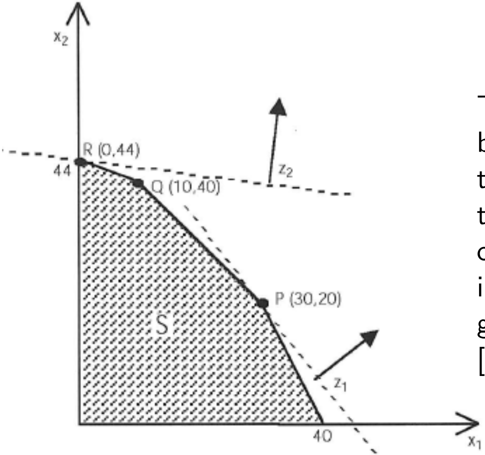
Efficiente or Pareto optimal solutions

A feasible solution $\bar{x} \in S$ is called **efficient** or **Pareto optimal**, if there is no other $x \in S$ such that $C\bar{x} \leq Cx$ and $C\bar{x} \neq Cx$.

The set of all efficient solutions is called efficient set or Pareto-optimal set.

Note: these definitions are not unique in literature and vary according to the references

Efficiente or Pareto optimal solutions



The solutions P, Q and R are said to be efficient because, for any of them, there is no other feasible solution that is equal to or better in both objective functions and strictly better in at least one of them. The same goes for any solution of the $[PQ]$ and $[QR]$ edges.

Detecting efficient solutions

Let $C^{\geq}$ denote the semi-positive polar cone generated by the gradients of the p objective functions:

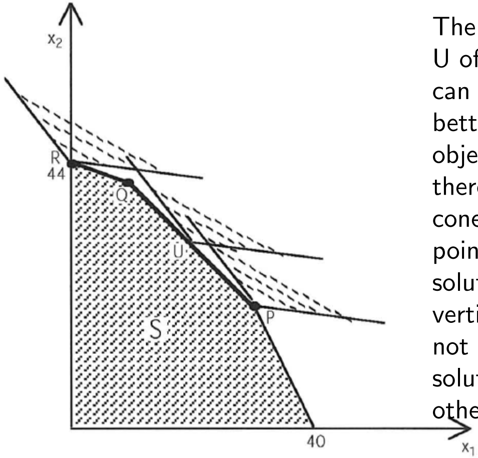
$$C^{\geq} = \{y \in \mathbb{R}^p : Cy \geq 0, Cy \neq 0\} \cup \{0\}$$

The domination set is defined as $D_{\bar{x}} = \{\bar{x}\} + C^{\geq}$. This set contains all points whose criterion vectors dominate the criterion vector $\bar{x} \in S$
This can be rewritten as:

$$D_{\bar{x}} = \{x \in \mathbb{R}^n : x = \bar{x} + y, Cy \geq 0, Cy \neq 0.\}$$

$\bar{x} \in S$ is efficient if and only if $D_{\bar{x}} \cap S = \{\bar{x}\}$

Detecting efficient solutions



The figure shows, for P, R and for point U of the [PQ] edge, the cones (which can be called cones of dominance) where better solutions would be located in both objective functions. As it can be seen, there is no intersection between these cones and the feasible region, apart from points P, R and U. It follows that the solutions of $[PQ] \cup [QR]$ (including the vertices) are efficient because they are not dominated by other feasible solutions. The same is not true of any other solution in the feasible region.

Nondominated points

Nondominated points

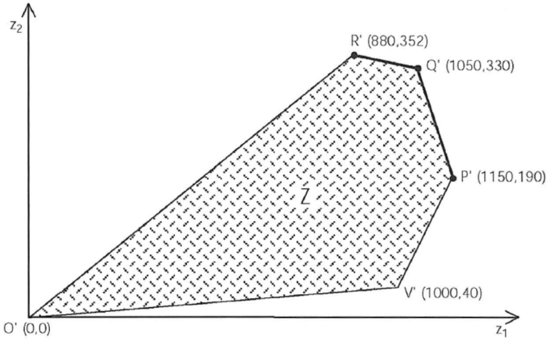
If $\bar{x}$ is efficient, $z = C\bar{x}$ is called **nondominated point**.

The set of all nondominated points is called the nondominated set or Pareto-optimal front or just Pareto front.

$\bar{x}$ dominates x iff $\bar{x}$ is no worse than x in all objectives and $\bar{x}$ is strictly better than x in at least one objective

Note: these definitions are not unique in literature and vary according to the references

Nondominated points



To represent the feasible region of the problem in the space of objectives Z, we start by determining the images of the vertices of S, which will also be vertices in Z.
The points of $[P'Q'] \cup [Q'R']$ (including the vertices) are nondominated, because the corresponding solutions are efficient.

Weak efficiency

Weak efficiency

A feasible solution $\bar{x} \in S$ is called weak-efficient if there does not exist another $x \in S$ such that $C\bar{x} > Cx$.

Note: these definitions are not unique in literature and vary according to the references

Example:

$$
\begin{array}{lll}
\max & z_1 = x_1 & \\
\max & z_2 = x_2 & \\
\text{s.to} & x_2 & \leq 4 \\
 & x_1 + x_2 & \leq 10 \\
 & 2x_1 + x_2 & \geq 10 \\
 & x_1, x_2 \geq 0 &
\end{array}
$$

Detecting efficient solutions

Example

$$\begin{array}{lll}
\max & z_1 = 3x_1 + 2x_2 & \\
\max & z_2 = x_1 & \\
\text{s.to} & x_1 + x_2 & \leq 6 \\
 & x_1 & \leq 4 \\
 & 2x_1 + x_2 & \leq 9 \\
 & x_1, x_2 & \geq 0
\end{array}$$

Detecting efficient solutions

Example

$$
\begin{array}{lll}
\min & 3x_1 + x_2 & \\
\max & x_1 + x_2 & \\
\text{s.to} & x_2 & \leq 3 \\
 & 3x_1 - x_2 & \leq 6 \\
 & x_1, x_2 & \geq 0
\end{array}
$$

Detecting efficient solutions

Example:

$$
\begin{array}{lll}
\max & z_1 = -x_1 + 3x_2 & \\
\max & z_2 = 3x_1 + 3x_2 & \\
\max & z_3 = x_1 + 2x_2 & \\
\text{s.to} & x_2 & \leq 4 \\
 & x_1 + 2x_2 & \leq 10 \\
 & 2x_1 + x_2 & \leq 10 \\
 & x_1, x_2 \geq 0 &
\end{array}
$$

Methods for detecting efficient solutions

There are several methods:

1. Optimizing one of the objective functions and transforming the remaining objectives into constraints
2. Optimizing a weighted sum of objective functions
3. Minimizing the Tchebycheff distance to a reference point

General approaches to multiobjective optimization

- A priori approaches. The decision maker needs to specify additional preferences prior to the optimization, e.g., weights of the objectives
- A posteriori approaches. Typically these methods generate efficient solutions. The decision maker must state his preferences a posteriori, i.e. after being informed about the trade-offs between non-dominant solutions
- Interactive approaches. The prioritization of objective functions is refined by requesting user feedback on preferences at multiple points in time during the execution of an algorithm

Goal Programming

Technique developed to deal with multi-criteria situations within the general framework of linear programming.

The goal programming approach was developed for decision problems involving two conflicting goals. Goal programming can be used to identify a solution that comes closest to achieving both goals.
Before applying the methodology, the decision maker must determine which, if either, goal is more important.

Goal Programming

Goal Programming is an *a priori* preference articulation approach where the decision maker:

- establishes a target (numeric goal) for each of the objectives
- formulates an objective function for each objective
- searches for a solution that minimizes the (weighted) sum of the deviations of these objective functions from their respective goals

Goal Programming

Goal programming problems can be categorized according to how the goals compare in importance:

- nonpreemptive goal programming (*programação com metas não hierarquizadas*) when all the goals are comparable in importance
- preemptive goal programming (*programação com metas hierarquizadas*) when there is a hierarchy of priority levels for the goals, so that the goals of primary importance receive first priority attention, those of secondary importance receive second-priority attention, and so forth.

Goal Programming

Each constraint associated to a goal is written as follows:

$$\sum_{j=1}^{n} a_{ij}x_j + d_i^- - d_i^+ = b_i$$

where b_i is the target.

- If the goal is to attain the target b_i, then both deviations d_i^- and d_i^+ should be minimized
- If the goal is to attain the target b_i or exceed it, then deviation d_i^- should be minimized
- If the goal is to not exceed target b_i, then deviation d_i^+ should be minimized

Goal Programming - Preemptive

There is a hierarchy of priority levels for the goals . Such a case arises when one or more of the goals clearly are far more important than the others

The initial focus should be on achieving as closely as possible these first priority goals

After we find an optimal solution with respect to the first priority goals, we can break any ties for the optimal solution by considering the second priority goals.

Any ties that remain after this reoptimization can be broken by considering the third priority goals, and so on.

Goal Programming - Preemptive

There are two basic methods based on linear programming for solving preemptive goal programming problems:

- sequential procedure (*Método Sequencial*)
- streamlined procedure (*Método Simplificado*)

Example: Goal Programming - Preemptive

Nicolo Investment Advisor company faces the following problem. A client has $80 000 to invest and, as an initial strategy, would like the investment portfolio restricted to two stocks:

Stock	Price/Share	Estimated Annual Return/Share	Risk Index/Share
U.S. Oil	$25	$3	0.50
Hub Properties	$50	$5	0.25

U.S. Oil, which has a return of $3 on a $25 share price, provides an annual rate of return of 12%, whereas Hub Properties provides an annual rate of return of 10%.

The client agreed that an acceptable level of risk would correspond to portfolios with a maximum total risk index of 700.

Another goal of the client is to obtain an annual return of at least $9000.

Example: Goal Programming - portfolio risk

The risk index per share, 0.50 for U.S. Oil and 0.25 for Hub Properties, is a rating Nicolo assigned to measure the relative risk of the two investments. Higher risk index values imply greater risk; hence, Nicolo judged U.S. Oil to be the riskier investment. By specifying a maximum portfolio risk index, Nicolo will avoid placing too much of the portfolio in high-risk investments.

To illustrate how to use the risk index per share to measure the total portfolio risk, suppose that Nicolo chooses a portfolio that invests all \$80 000 in U.S. Oil, the higher risk, but higher return, investment.
Nicolo could purchase $\$80\,000/\$25 = 3200$ shares of U.S. Oil, and the portfolio would have a risk index of $3200 \times 0.50 = 1600$.
Conversely, if Nicolo purchases no shares of either stock, the portfolio will have no risk, but also no return.
Thus, **the portfolio risk index will vary from 0 (least risk) to 1600 (most risk)**.

Nicolo's client would like to avoid a high-risk portfolio; thus, investing all funds in U.S. Oil would not be desirable.

However, the client agreed that **an acceptable level of risk would correspond to portfolios with a maximum total risk index of 700**.

Thus, considering only risk, one goal is to find a portfolio with a risk index of 700 or less.

Example: Goal Programming - portfolio return

Another goal of the client is to obtain an annual return of at least \$9 000.

This goal can be achieved with a portfolio consisting of 2000 shares of U.S. Oil, at a cost of $2000\times$ \$25 $=$ \$50 000, and 600 shares of Hub Properties, at a cost of $600\times$\$50 $=$ \$30 000; the annual return in this case would be $2000\times$\$3 $+$ $600\times$\$5 $=$ \$9000.

Note, however, that the portfolio risk index for this investment strategy would be $2000\times0.50 + 600\times0.25 = 1150$; thus, this portfolio achieves the annual return goal but does not satisfy the portfolio risk index goal.

Example: Goal Programming - the conflicting goals

The portfolio selection problem is a multicriteria decision problem involving two conflicting goals: one dealing with risk and one dealing with annual return

The goal programming approach was developed precisely for this kind of problem.

Goal programming can be used to identify a portfolio that comes closest to achieving both goals.

Before applying the methodology, the client must determine which, if either, goal is more important.

Example: Goal Programming - the conflicting goals

Primary Goal (Priority Level 1)
Goal 1: Find a portfolio that has a risk index of 700 or less.

Secondary Goal (Priority Level 2)
Goal 2: Find a portfolio that will provide an annual return of at least \$9000.

In goal programming terminology, they are called **preemptive priorities** because the decision maker is not willing to sacrifice any amount of achievement of the priority level 1 goal for the lower priority goal

Example: Goal Programming - the conflicting goals

The portfolio risk index of 700 is the target value for the priority level 1 (primary) goal, and the annual return of \$9000 is the target value for the priority level 2 (secondary) goal.

The difficulty in finding a solution that will achieve these goals is that only \$80 000 is available for investment.

Example: Goal Programming - the decision variables

define the decision variables:

x_1 is the number of shares of U.S. Oil purchased

x_2 is the number of shares of Hub Properties purchased

Example: Goal Programming - the constraints

this portfolio problem has one constraint that corresponds to the funds available

Because each share of U.S. Oil costs \$25 and each share of Hub Properties costs \$50, the constraint representing the funds available is

$$25x_1 + 50x_2 \leq 80\ 000$$

To complete the formulation of the problem, we must develop a **goal equation** for each goal:

Example: Goal Programming - the goal equations

for Goal 1

Depending on the values of x_1 and x_2, the portfolio risk index may be less than, equal to, or greater than the target value of 700

To represent these possibilities mathematically, we create the goal equation

$$0.5x_1 + 0.25x_2 = 700 + d_1^+ - d_1^-$$

where
d_1^+ represents the amount by which the portfolio risk index exceeds the target value of 700
d_1^- represents the amount by which the portfolio risk index is less than the target value of 700

d_1^+ and d_1^- are called **deviation variables**

Example: Goal Programming - the goal equations

for Goal 2

To develop a goal equation for the secondary goal, we begin by writing a function representing the annual return for the investment:
Annual return $= 3x_1 + 5x_2$

now define the two **deviation variables**
d_2^+ represents the amount by which the annual return for the portfolio is greater than the target value of \$9000
d_2^- represents the amount by which the annual return for the portfolio is less than the target value of \$9000

Using these two deviation variables, we write the goal equation for goal 2 as

$$3x_1 + 5x_2 = 9000 + d_2^+ - d_2^-$$

Example: Goal Programming - the objective function

The objective function in a goal programming model calls for **minimizing** a function of the deviation variables

In the portfolio selection problem, the most important goal, denoted P1, is to find a portfolio with a risk index of 700 or less.

This problem has only two goals, and the client is unwilling to accept a portfolio risk index greater than 700 to achieve the secondary annual return goal.

Therefore, the secondary goal is denoted P2.

These goal priorities are referred to as **preemptive priorities** because the satisfaction of a higher level goal cannot be traded for the satisfaction of a lower level goal

Example: Goal Programming - model

$$
\begin{array}{lll}
\min & P1(d_1^+) \; + \; P2(d_2^-) & \\
\text{s.to} & 25x_1 + 50x_2 \leq 80\ 000 & \text{Funds available} \\
 & 0.5x_1 + 0.25x_2 - d_1^+ + d_1^- = 700 & \text{P1 goal} \\
 & 3x_1 + 5x_2 - d_2^+ + d_2^- = 9000 & \text{P2 goal} \\
 & x_1, x_2, d_1^+, d_1^-, d_2^+, d_2^- \geq 0 &
\end{array}
$$

Example: Goal Programming - preemptive priorities

Goal programming problems with preemptive priorities are solved by treating priority level 1 goals (P1) first in an objective function

The idea is to start by finding a solution that comes closest to satisfying the priority level 1 goals

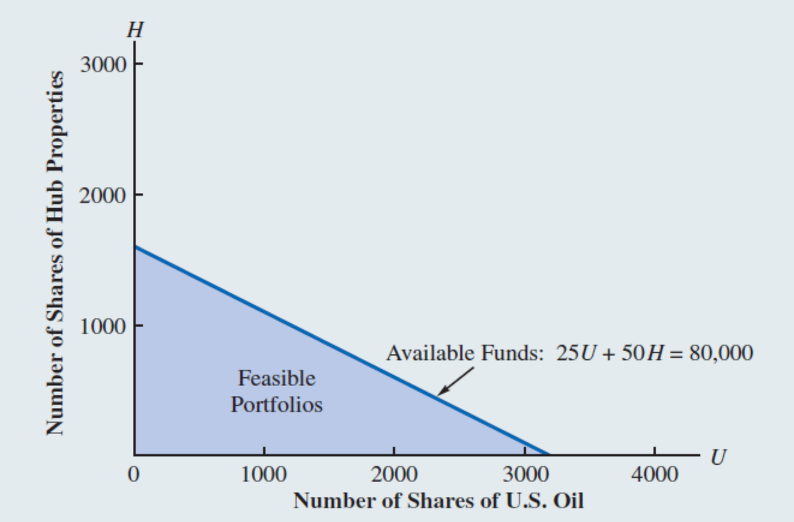
This solution is then modified by solving a problem with an objective function involving only priority level 2 goals (P2)

however, revisions in the solution are permitted only if they do not hinder achievement of the P1 goals

In general, solving a goal programming problem with preemptive priorities involves solving a sequence of linear programs with different objective functions; P1 goals are considered first, P2 goals second, P3 goals third, and so on. At each stage of the procedure, a revision in the solution is permitted only if it causes no reduction in the achievement of a higher priority goal.

Each linear program is obtained from the one at the next higher level by changing the objective function and adding a constraint.

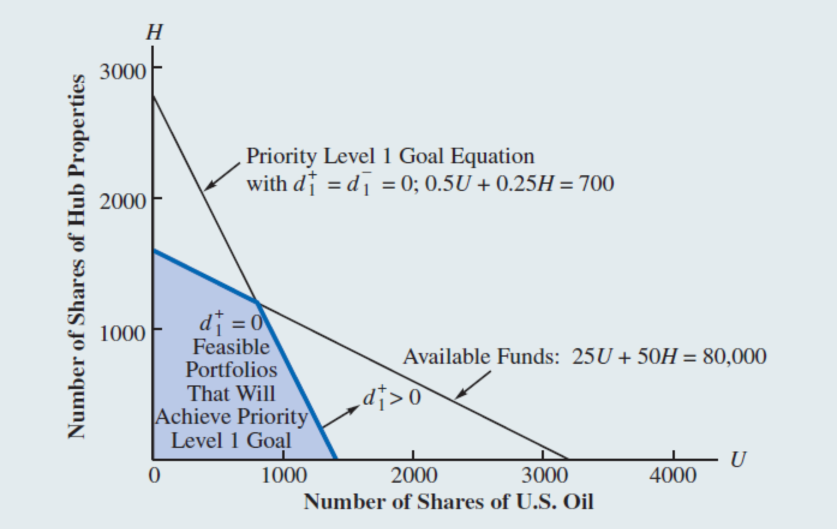
Example: Portfolios That Satisfy the Available Funds Constraint



Example: Goal Programming - P1 problem

$$
\begin{array}{lll}
\min & d_1^+ & \\
\text{s.to} & 25x_1 + 50x_2 \leq 80\ 000 & \text{Funds available} \\
& 0.5x_1 + 0.25x_2 - d_1^+ + d_1^- = 700 & \text{P1 goal} \\
& 3x_1 + 5x_2 - d_2^+ + d_2^- = 9000 & \text{P2 goal} \\
& x_1, x_2, d_1^+, d_1^-, d_2^+, d_2^- \geq 0 &
\end{array}
$$

Example: Portfolios That Satisfy the P1 Goal

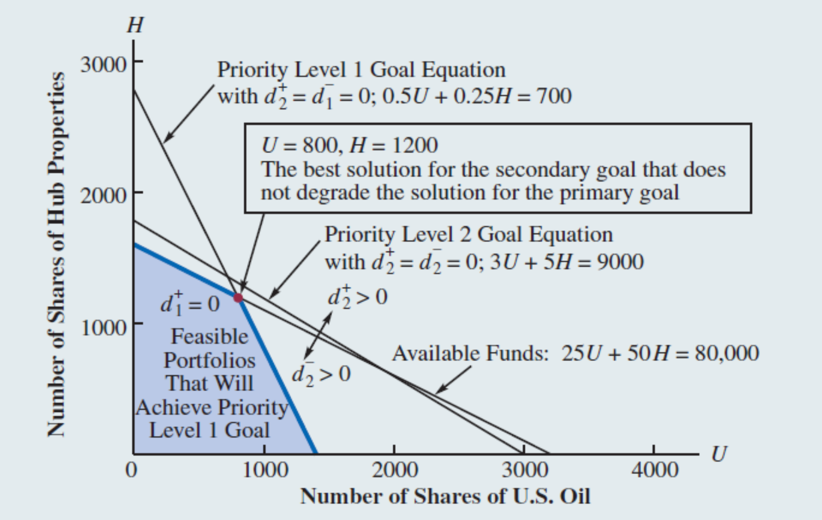


Example: Goal Programming - P2 problem

$$
\begin{array}{lll}
\min & d_2^- & \\
\text{s.to} & 25x_1 + 50x_2 \leq 80\,000 & \text{Funds available} \\
 & 0.5x_1 + 0.25x_2 - d_1^+ + d_1^- = 700 & \text{P1 goal} \\
 & 3x_1 + 5x_2 - d_2^+ + d_2^- = 9000 & \text{P2 goal} \\
 & d_1^+ = 0 & \text{maintain achiev. P1 goal} \\
 & x_1, x_2, d_1^+, d_1^-, d_2^+, d_2^- \geq 0 &
\end{array}
$$

Example: Best Solution with Respect to Both Goals

(Solution to P2 Problem)



Example: Goal Programming - the solution

the goal programming solution for the Nicolo Investment problem recommends that the $80,000 available for investment be used to purchase 800 shares of U.S. Oil and 1200 shares of Hub Properties.
Note that the priority level 1 goal of a portfolio risk index of 700 or less has been achieved.
However, the priority level 2 goal of at least a $9000 annual return is not achievable. The annual return for the recommended portfolio is $8400.