Types of Optimization Models

Model Formulations

- **Different types of optimization model formulations exist:**
  - Classical non-linear formulation
  - Linear Programming formulation
  - Baseline model formulation
  - Goal Programming formulation
  - Compromise Decision Support Problem formulation
  - etc.
- **Basic classifications are:**
  - Constrained versus unconstrained
  - Linear versus non-linear
  - Single objective versus multi-objective
- **Another classification can be made by variables:**
  - continuous/discrete/mixed-integer

Introduction to Multi-Objective Optimization

Single versus Multi-Objective

What should you use and what is available?

Most important rule:

Never restrict yourself in modeling!

- **Design is multi-objective by nature, so we will look at some <u>multi-objective</u> formulations first.**

Different formulations

- There are different ways to formulate a multi-objective optimization model
- Some covered are:
  - Baseline model
  - Goal Programming (GP) model
  - Compromise Decision Support Problem model
- Others exist

Baseline Model

Ignizio proposes the following "baseline" model:

Find: The vector of problem variables $\underline{X}$

Satisfy: The goals

$$A_t(\underline{X}) \begin{matrix} \geq \\ = \\ \leq \end{matrix} T_t \qquad \text{for all } t$$

Maximize: $A_r(\underline{X})$ for all r

Minimize: $A_s(\underline{X})$ for all s

- **He argues that this is one of the most generic and flexible mathematical programming models.**
- ***What is missing?***

Goal Programming (GP)

- Another multiobjective mathematical “programming” technique is **<u>Goal Programming</u>** (GP)

- The term "<u>goal programming</u>" is used by its developers to indicate the search for an "optimal" program (i.e., a set of policies to be implemented) for a **<u>mathematical model that is composed solely of goals</u>**.

- Developers argue that any mathematical programming model may find an equivalent representation in GP.

- “GP provides an alternative representation that often is more effective in capturing the nature of real world problems.”

Difference between Objectives and Goals

In Goal Programming a distinction is made between an objective and a goal:

- **<u>Objective</u>:** In mathematical programming, an objective is a function that we seek to optimize, via changes in the problem variables.

  The most common forms of objectives are those in which we seek to maximize or minimize. For example,

  $$\text{Minimize } Z = A(\underline{X})$$

- **<u>Goal</u>:** In short, a goal is an objective with a "right hand side".

  This right hand side (T) is the target value or aspiration level associated with the goal. For example,

  $$A(\underline{X}) \geq T$$

Solving Multi-objective Models

- **Solving multi-objective models is NOT standard practice (yet).**

- **Often, a first step in solving these models is a *<u>model transformation</u>* into a model that CAN be solved using an existing algorithm/solver.**

- **How do we solve such a baseline model?**

- **For example, for solving a baseline model, we can convert it to:**
  - single objective nonlinear programming (NLP) problem.
  - multi-objective Goal Programming (GP) problem.
  - multi-objective compromise Decision Support Problem (DSP).
  - other...

Some Model Transformation Basics

Transforming a baseline model into a GP model

Here are steps how to transform a “baseline model” into a “GP” model:

Step 1: Transform all objectives into goals by establishing associated aspiration levels based on the belief that a real world decision maker can usually cite (initial) estimates of his or her aspiration levels. Hence,

$$\text{maximize } A_r(\underline{X}) \text{ becomes } A_r(\underline{X}) \geq T_r \text{ for all } r$$

$$\text{minimize } A_s(\underline{X}) \text{ becomes } A_s(\underline{X}) \leq T_s \text{ for all } s.$$

where T_r and T_s are the respective aspiration levels (targets).

Step 2: Rank-order each goal according to its perceived importance. Hence, the set of hard goals (i.e., constraints in traditional math programming) is always assigned the top priority or rank.

Step 3: All the goals must be converted into equations through the addition of deviation variables

Going from Inequalities to Equalities

Converting Inequalities - Standard Approach

- Note: A computer does not “like” inequalities.
- Thus: Inequalities have to be converted to equalities

- In general, converting equalities to inequalities is achieved by adding variables.

- This is a VERY common practice in optimization.

Deviation Variables - "Distance to target"

- In Goal Programming and other approaches (like compromise Decision Support Problem) "deviation" variables are used to convert inequalities to equalities.
- The deviation variable d is (then) defined as:

$$d = T_i - A_i(\underline{X})$$

- Note: Mathematically, the deviation variable d can be negative, positive, or zero.
- From a reality point of view, a deviation variable represents the <u>distance</u> (deviation) between the <u>aspiration level (target)</u> and the <u>actual attainment</u> of the goal.

Two Deviation Variables instead of One

- The deviation variable d can be replaced by two variables:

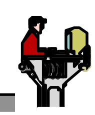
$$d = d_i^- - d_i^+$$

where $d_i^- \bullet d_i^+ = 0$ and $d_i^-, d_i^+ \geq 0$

- ***Why?*** Many optimization algorithms do not "like" negative numbers and the preceding ensures that the deviation variables <u>never</u> take on negative values.
- The product constraint ensures that one of the deviation variables will always be zero.
- The goal formulation (now) becomes:

$$A_i(X) + d_i^- - d_i^+ = T_i; \qquad i = 1,2, \ldots , m$$

subject to $d_i^- \bullet d_i^+ = 0$ and $d_i^-, d_i^+ \geq 0$



Values of Deviation Variables

Note that a goal is always expressed as an equality:

$$A_i(\underline{X}) + d_i^- - d_i^+ = T_i; \qquad i = 1,2, \ldots, m$$

And when considering this equality, the following will be true:

if $A_i(\underline{X}) < T_i$ is true, then ($d_i^- > 0$ AND $d_i^+ = 0$) must be true;

if $A_i(\underline{X}) > T_i$ is true, then ($d_i^- = 0$ AND $d_i^+ > 0$) must be true;

if $A_i(\underline{X}) = T_i$ is true, then ($d_i^- = 0$ AND $d_i^+ = 0$) must be true.

When in doubt, just use a numerical example.

“Desired” Values of Deviation Variables

Again, note that a goal is always expressed as an equality.

$$A_i(X) + d_i^- - d_i^+ = T_i; \qquad i = 1,2, \ldots, m$$

To achieve a goal (i.e., reach the target), 3 situations are possible:

1. To satisfy $A_i(\underline{X}) \leq T_i$, we must ensure that the deviation variable d_i^+ is zero.
   - The deviation variable d_i^- is a measure of how far the performance of the actual design is from the goal.

2. To satisfy $A_i(\underline{X}) \geq T_i$, the deviation variable d_i^- must be made equal to zero.
   - In this case, the degree of “overachievement” is indicated by the positive deviation variable d_i^+.

3. To satisfy $A_i(\underline{X}) = T_i$, <u>both</u> deviation variables, d_i^- and d_i^+ must be zero.

Question: How would this change if we only had a single d_i that can be positive or negative?

Thus, to achieve a target, we must <u>minimize</u> the unwanted deviation(s)!

Minimizing deviations

Consider the preceding three situations again.

To achieve a goal (i.e., reach the target), 3 situations are possible:

1. To achieve $A_i(\underline{X}) \leq T_i$, we must *minimize* (d_i^+)

2. To achieve $A_i(\underline{X}) \geq T_i$, we must *minimize* (d_i^-)

3. To achieve $A_i(\underline{X}) = T_i$, we must *minimize* ($d_i^- + d_i^+$).

(How would this change if we only had a single d_i that can be positive or negative?_

Big Question: What if you have more than one goal?

That is, how do you minimize <u>multiple</u> deviation variables?

Prioritizing Goals

Two Approaches to Prioritizing Goals

Goals are not equally important to a decision maker.

How do we represent our preferences?

Two approaches are:

1) **<u>Assign weights</u>** and calculate the sum of the deviation variables ('distance to target') multiplied by their individual weights.

2) **<u>Rank order</u>** goal deviations in priority levels, often referred to as a preemptive formulation. The preemptive formulation does not exclude the assignment of weights.

Note: Other techniques exist, but right now we focus on the above two.

Weighted Sum Approach

- **Assigning weights, or weighted sum approach, is one of the most common ways of converting multi-objective/multi-goal problems into a single objective problem.**

- Min $z = (w_1 d_1^- + w_2 d_2^+ + \ldots.) = \Sigma\,(w_i d_i^- + w_k d_k^+)$

- **The weights (w) can be any value, in principle.**
  - In case the sum of the weights equals 1, then we speak of an <u>archimedean formulation</u>.

- However, assigning weights without thought can cause problems.
  - *Can you name some?*

Rank Ordering

- In Rank Ordering, you prioritize one goal/objective above each other without giving explicit mathematical weights.
  - Basically, in words, Goal A has to be achieved before Goal B. I should not even think about Goal B yet if Goal A has not been achieved yet.

- One mathematical construct that is used in rank ordered formulations is the Lexicographic Minimum.

- The concept of a lexicographic minimum is used in several multi-objective formulations
  - Goal Programming
  - Compromise DSP

Lexicographic Minimum - Definition

LEXICOGRAPHIC MINIMUM Given an ordered array $f^{(i)} = (f_1, f_2, \ldots, f_n)$ of nonnegative elements f_k's, the solution given by $f^{(1)}$ is preferred to $f^{(2)}$ iff

$$f_k^{(1)} < f_k^{(2)}$$

and all higher order elements $(f_1, \ldots, f_{k-1})$ are equal. If no other solution is preferred to $f^{(1)}$, then $f^{(1)}$ is the lexicographic minimum.

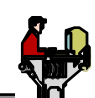
Examples?

Compromise Decision Support Problem formulation

- **Given:**
  - Relevant information
- **Find:**
  - System variables
  - Deviation variables
- **Satisfy:**
  - System constraints
  - System goals
- **Minimize:**
  - Deviation function

Arguably, just "another" multi-objective optimization model, but it is geared towards engineering design.

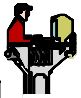
Note use of different "keywords" (given, find, satisfy, minimize).



The Effect of Selecting a Formulation

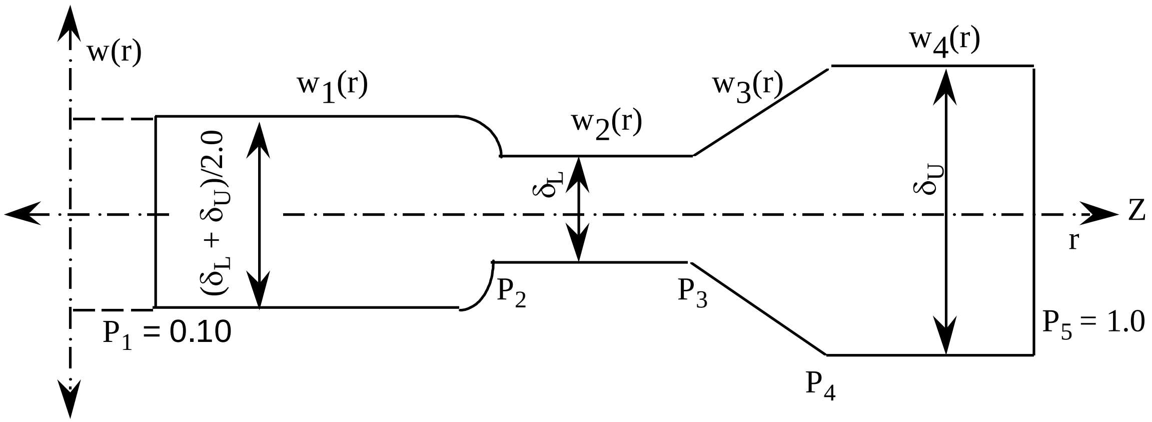
- ***It is important to note that differences in formulation CAN cause differences in results.***

- **The most influential factors are the choices of:**
  - **Objectives versus goals**
  - **Goal Priorities**
  - **Constraints versus goals (constraints are higher priority)**
  - **Goal targets**



Case Study/Example

Rotating Disk (Flywheel) Example



- Experimental evidence suggests that at high speeds the stresses are high near the hub of the rotating disk. For this reason, to get the stresses within safe limits, it is advisable to have more mass near the hub.
- The design criterion is to locate the points, P2 to P4 such that the kinetic energy is maximized and the mass of the rotating disk is minimized.

"Baseline" Model for Flywheel

Given

The relevant information for the disk:

Angular velocity of the disk	ω	= user input (rad/sec)
Lower limit of thickness	δ_L	= 0.01 (m)
Upper limit of thickness	δ_U	= 0.10 (m)
Location of the hub	P_1	= 0.05 (m)
Location of the rime	P_5	= 0.5 (m)
Slope of the linear portion	δ'	= 0.9
Density of the material of disk	ρ	=7830 (kg/m^3)
Yield stress of the material of disk	YS	=1.48E9 (N/m^2)

Relevant equations for the physics of the problem.

Find

System variables

They determine the profile of the rotating disk,

P_2, P_3, and P_4.

Satisfy

System constraints

The stress constraints,

$\sigma_R(r) \leq \sigma_y$,

$\sigma_T(r) \leq \sigma_y$,

where σ_R, σ_T, and σ_y are the radial stress, tangential stress and yield stress respectively.

The constraints on the geometry of the rotating disk,

$P_1 \leq P_2$,

$P_2 \leq P_3$,

$P_3 \leq P_4$,

$P_4 \leq P_5$.

Maximize

The kinetic energy (K) of the rotating disk is to be maximized.

Minimize

The weight (M) of the rotating disk is to be minimized.

What is wrong with this "Baseline" model?

Different single objective functions

Minimize

The kinetic energy (K) of the rotating disk is to be maximized,

$$f(P_2, P_3, P_4) = -K$$

where f is the objective function.

Minimize

The weight (M) of the rotating disk is to be minimized,

$$f(P_2, P_3, P_4) = M$$

where f is the objective function.

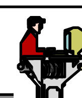
Weighted sum approach:

Minimize

The kinetic energy of the disk is to be maximized and its weight (M) is to be minimized,

$$f(P_2, P_3, P_4) = 0.6(-K) + 0.4M,$$

where f is the objective function.



Different Design Scenarios

The compromise DSP template is exercised in three ways,

- The deviation function is modeled in the preemptive form with the achievement of the kinetic energy goal as first priority.
- the deviation function is modeled in the preemptive form with the achievement of the weight goal as first priority.
- the deviation function is formulated for the Archimedean form giving a weight of 0.6 for the achievement of the aspiration level of the kinetic energy of the disk and a weight of 0.4 for the weight of the disk.

The traditional single-objective model is exercised in three ways,

- one with kinetic energy as objective function and mass of the disk as constraint,
- the other with mass of the disk as objective function and kinetic energy as constraint, and
- as a weighted sum of the two objectives.

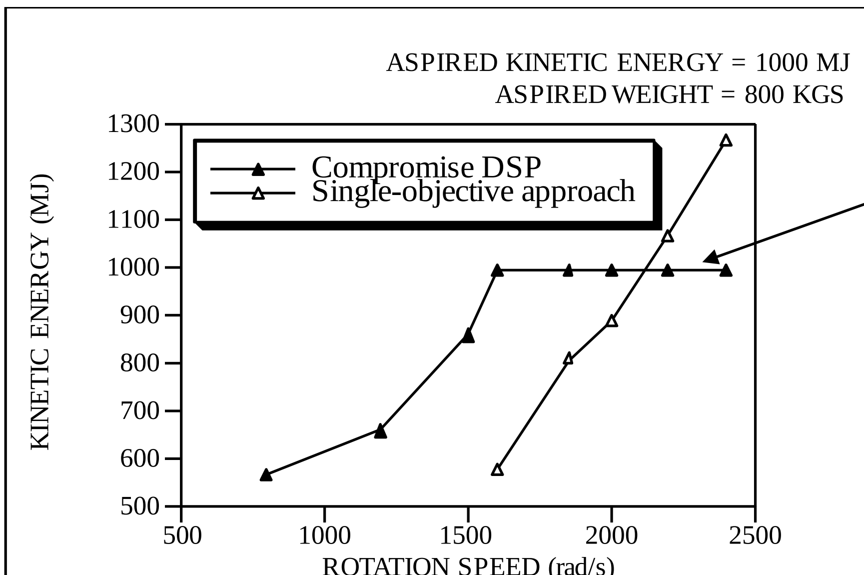
Differences in Results

- By way of illustrating the "power" of a preemptive formulation, a comparison of the results obtained is made for Scenario I:

  First priority: maximize the kinetic energy of the disk

  Second priority: minimize the mass.

- The aspiration levels for these objectives are set at 1000 MJ and 800 kgs respectively



Note that compromise DSP solution "sticks" to 1000 MJ while trying to minimize weight.

Question: *At what speed do you expect a minimum weight?*