Industrial Operations Research Practitioner’s Wish List

February 2016
Laura Kang
Three questions to ask with every project:

1. What is my problem? (in English)
2. What is my problem? (in mathematics)
3. Which tool/software do I use?

Customer needs

Expertise and resource availability
Never say “can’t be done” to customers

Make it Work!
What matter depends on problem size and complexity

- Formulation Performance
- Performance Speed
- Scheduling/Matching for hiring/rotation program
- Implementation Distribution
- Ply Shape and Stacking Sequence Optimization for Multi-Panel Laminates
Example 1: Interview Scheduling/Matching

Participiants → Managers

Participant-Manager Match

Interview Event

Participants/Managers Preference
Create a schedule for interviews for matching events

Integer Optimization

Participants

Positions

Max/Min number of interviews

Interview Schedules Optimization

Conflict Free Interview Schedule

Skill set information

Participant preferences

List of potential interviews

Optional input depending on how each customer prefers to schedule interviews
This problem is small and simple and successfully implemented

Linear objective function and constraints

Sparse constraint matrix (assignment-like)

Relatively small data set (100s participants with 100s managers)

Excel/CBC used to develop a customized tool for each customer
Create job-participant match based on preference
Minimum cost network flow problem

Min-cost Network Flow Problem

Matching Optimization

Position – Participant Match

Participants Preference

Manager Preference

Participant/Manager Weights

Prevented Match

Fixed Match

Sensitivity Analysis

Match restriction depending on customer preference
Successful implementation saves time and produces better job placements

Excel + In-house network optimization tool used to develop a customized tool for each customer

~80 hours per program per cycle saved

Better/fairer assignment

Solution time saving + Better schedule/match = Happy customers
Example 2: Composite Manufacturing
Definition and Benefits of Composites

Definition

• Dictionary: “made of different parts or elements”
• Aerospace industry: “fiber-reinforced plastics”

Benefits

▪ The fiber directions can be adjusted to the load
▪ The number of layers and the sequence can be adjusted
▪ Lower density than aluminum (lower lbs/in3)
▪ Easier integration of parts

Light Weight = Cost Saving

Composite wing
(http://www.compositesworld.com/blog/post/a-composite-wing-giant-is-born)
Composite structure is made by laying plies with different orientations on top of each other.

Ply is a thin layer of material and defined by a shape and an angle.

Plies can be laid up by hand or by automated layup equipment, such as tape layers or automated fiber placement machines.

Stacking plies in the right order (reg. angle and shapes) is essential for the strength of the material.
Simultaneously optimized ply shapes and stacking sequences
Using 2D Matrix Representation of 3D manufacturing problem

Matrix Representation

Each row is a layup sequence with associated fiber angle

Indices determine presence of a ply → ply shapes

Green = 45° ply
Blue = -45° ply
Red = 45° ply
Yellow = 45° ply

Schematic of wing skin

Ply Shapes and Stacking Sequence
Creative formulation + Integer Optimization = time saving + cost saving + possible design improvement

Objective = Minimize total boundaries of plies (QP)

Linear constraints – ill-conditioned constraints

With straightforward MIO, it takes 30 hours to get an ok solution for a sample problem (about 1/10 of actual size) with Gurobi

What makes the problem hard? Combinatorial nature of stacking sequence

Reformulated (still as IMO) to take stacking sequence combination as a preprocessing and reduce the complexity of problem significantly

New approach generates a reasonable solution for the whole wing in 5 minutes
If I had a magic wand – complete wish list

No licensing

Easy install and compatibility across platform

Ability to integrate with other platform/programming languages

Ability to handle big data set

Easy to learn with lots of examples and documentation

Ability to handle complicated optimization (combinatorial, nonlinear)

Fast and Furious(ly optimal)
What if I have to pick one? – It depends