Overview

- **This is a project I assigned a few years ago**
- **Read the project description**
  - What are the objectives?
  - What is to be decided – what are the variables?
  - Which parameters are set? Which parameters will have to be varied (sensitivity analysis)?
- **Look at the provided input data**
  - Note that it gives most of the necessary sets and data
  - It is in another language, so you'll have to translate it to MPL
  - There's a fair amount if it, though, and it has multiple dimensions

First, Let’s Answer the Questions (1 of 2)

- **What are the objectives?**
  - Minimize aircraft losses
  - Minimize the number of days to kill the target set
  - Meet investment limit (which is subject to discussion)
  - We will have to decide how to trade between these three objectives, so we have a goal program
- **What are the variables?**
  - The number of weapons to buy
  - The assignment of weapons to targets in each scenario
  - The assignment of sorties (one aircraft flying to one target) in each scenario
  - We might need other variables too

First, Let's Answer the Questions (2 of 2)

- **Which parameters will have to be varied?**
  - Overall investment: opinion ranges from \$35M - \$200M
  - Probabilities of the 3 scenarios
  - Fortunately, people seem to agree on everything else
  - However, the fact that certain things have to be varied may affect the design of the model

How to Start?

- **What's the general form of the model?**
  - Multiple scenarios => two-stage recourse model
  - Multiple goals => some sort of goal program
  - Final model will have to combine goals with two-stage recourse formulation
- **However, we need to work on some things with the basic 1-scenario problem**
  - How do we determine the length of the bombing campaign?
  - How do we enforce all the conditions on using certain bombs in certain weather conditions?
  - What variables will we need to represent all this?

Campaign Length and Weather (1 of 3)

- **This is probably the hardest part of this project**
- **Take the SSC scenario**
  - It has bad weather 30% of the time (proportion 0.3)
  - We can fly 90 sorties per day in this scenario
  - If we need to fly 270 sorties in bad weather, it will take 270 / (90*0.3) = 10 days on average to do it
- **But why fly in bad weather at all?**
  - We still want to minimize the time to conduct the campaign
  - Not flying in bad weather increases campaign length by at least 30% (and gives the enemy an unearned advantage)

Campaign Length and Weather (2 of 3)

- **Here's a question the students raised in this project**
  - There are 6 target types (3 collateral damage X 2 hardness)
  - Does each target type have to be killed in proportion to the weather?
  - Example: SCC has 120 soft targets with strict collateral damage requirements. Do we have to kill 40% in good weather (48), 30% in fair weather (36), and 30% (36) in bad weather?
- **Answer**
  - No, these are fixed targets (e.g., buildings)
  - We can attack them whenever we want
  - We do NOT need to constrain the number attacked to weather proportions
  - However, we still need to track the TOTAL number of sorties flown in various weather conditions
- **An aside**
  - You could argue that you need to constrain attacks to weather, because the enemy might use certain buildings on certain days

Campaign Length and Weather (3 of 3)

- **So here's the sub-model**
  - days required for scenario >= total sorties flown in weather condition / (sorties per day in scenario X proportion of time in weather condition)
  - We need this constraint for every weather condition
  - So, days required will be the maximum
- **Another question: can sorties assigned be fractional?**
  - Answer: yes, we are working with expected values for kills and attrition
  - Example: A GPS PK = 0.6 => 1/0.6 = 1.67 bombs *on average* required for kill
  - 2 GPS bombs per sortie / 1.67 bombs required => 1.2 *sorties* required *on average* to kill the target
  - Since those numbers are fractional, it is OK to use fractional (continuous) sortie assignments
  - We are treating the sortie assignments as expected values

Enforcing Weapon-Target Limitations

- **Certain bombs only work in certain weather states**
  - LGB (laser-guided bomb) requires good weather
  - GPS bomb works in all weather states, but is less accurate and requires more on average to get a kill
- **Certain bombs have unacceptable collateral damage**
  - Enormous consideration in modern warfare
  - Unguided weapons can have large miss distances due to wind and often hit unintended targets
  - However, guided weapons are much, much more expensive
- **So, the assignment variables ...**
  - Must be a function of scenario, target type (hardness and collateral damage), and weather

Next Step: Start Formulating

- I'll show you this via MPL code
- As usual, the first step is to write the indexes

```
INDEX

  e := (MTW1,MTW2,SSC)           {theater}
  b := (soad,gps,lgb,unguided) { weapon type }
  c := (strict, medium, none ) {collateral damage category
}
  h := (hard, soft )             { target hardness }
  w := (good, fair, bad )        { weather state }
```

Multidimensional Sets

- I wanted the students to use multidimensional sets to define allowable combinations of things
- Here are the sets I defined, in MPL:

```
{ allowable weapon and weather combinations }

 wxw[b,w] := (soad.good, soad.fair, soad.bad,
             gps.good,gps.fair,gps.bad,lgb.good,
             unguided.good, unguided.fair );

 { allowable weapon and collateral damage combinations }

 cda[b,c] := (lgb.strict, lgb.medium, lgb.none,
             soad.medium, soad.none,
             gps.medium, gps.none,
             unguided.none )
```

Data

- There's a lot of data in this model
- One of the aims of the project was to teach students how to get higher-dimensional data into MPL
- See the MPL code for all of it; but here are examples

```
{ target data by scenario, collateral damage, hardness }

 TGTS[e,c,h] := [MTW1,strict,hard,200,
                 MTW1,strict,soft,400,
                 MTW1,medium,hard,400,

PK[b,e,h] := [soad,MTW1,soft,.86,
               soad,MTW1,hard,.60,
               soad,MTW2,soft,.77,

 ATR[e,b,w] := [ MTW1, soad, good, .0001,
                 MTW1, soad, fair, .0001,
                 MTW1, soad, bad, .0001,
                 MTW1, lgb, good, .005,
                 MTW1, lgb, fair, .007,
```

Variables

- **This is a recourse model, so we have**
  - Initial decisions: this is the number of bombs bought
  - Everything else: these are decisions made *in each scenario* (indexed by e)
- **Here are the variables I used**
  - Note the use of the multidimensional sets to limit allocation variables to allowed combinations
  - This is a good way to use the MPL “IN” operator

```
VARIABLES

 bought[b];       { Weapons bought }

 attr[e];         { Attrition by theater }

 days[e];         { Days to prosecute campaign by theater }

 { Sorties allocated by scenario, weapon, target damage/hardness, and weather }

 sorties[e,b,c in cda, h, w in wxw];
```

Modeling the Goals

- **This is the second-hardest part of the project**
- **And, there are several choices of how to do it**
- **There are 3 factors**
  - Total aircraft attrition (losses)
  - Expected days to complete the campaign
  - Money spent on weapons
- **I used a weighted objective, but:**
  - I knew I would make several runs
  - I could get a “near-preemptive” goal program by using large and small weights
  - I could control the budget by a simple constraint, and easily test many budgets

A Setup for a Run

- **What was I looking at here?**
  - Wanted mostly to minimize expected days (weight = 1)
  - Gave a small weight to attrition to make sure that it was considered (break ties among near-identical solutions)
  - I did not weight the cost; I handled that via a budget constraint
  - Note the use of the MPL MACRO function

```
DAYWGT := 1;
ATTRWGT := 0.0001;
COSTWGT := 0;

MACRO

  bcost:=sum(b: COST[b]*bought[b]);

MODEL

  Min  weighted = DAYWGT*SUM(e: PROB[e]*days[e]) +
                  ATTRWGT*SUM(e: PROB[e]*attr[e]) +
                  COSTWGT*bcost;

 bcost < BUDGET;
```

Constraints

```
SUBJECT TO

  kills[e,c,h] WHERE (TGTS[e,c,h] > 0):   { Kill constraints }

    SUM(b,w: PK[b,e,h]*LOAD[b]*sorties[e,b,c,h,w]) > TGTS[e,c,h];

  buys[e,b]:         { Buy and inventory constraints - by scenario }

    SUM(c,h,w: LOAD[b]*sorties[e,b,c,h,w]) < INV[b] + bought[b];

  expattr[e]:        { Expected attrition by scenario - passenger constraints }

    attr[e] = SUM(b,c,h,w: ATR[e,b,w]*sorties[e,b,c,h,w]);

  daysreq[e,w]:      { Days required by scenario - passenger constraints }

    SRTD[e]*WX[e,w]*days[e] > SUM(b,c,h: sorties[e,b,c,h,w]);

  bcost < BUDGET;    { Total spent on weapons }

BOUNDS

  bought[b] < MAXBUY[b];
```

Comments on Constraints

- **Remember what a “passenger variable” is**
  - Quantity computed as a convenience to make the model easier to understand
  - Could be substituted out
  - The “passenger constraints” are there to compute the passenger variables **attr[e]** and **days[e]**
  - You might be tempted to use the MPL MACRO function, but MPL does not allow macros to be indexed
- **Note also the daysreq constraints**
  - The constants are multiplied on the LHS, rather than divided on the RHS
  - Again, MPL doesn’t like dividing constants in equations

And This is the Whole Model!

- **Despite the frightening description, the model is:**
  - Fairly simple
  - Combines a goal program and a recourse model
  - Allows easy adjustments to the three goals to see how the answers change
- **But what was hard?**
  - Figuring out how to do weather and days required for the campaign
  - Getting the data into MPL
  - Getting MPL to limit weapon-target-weather assignments to allowed combinations
  - Coming up with a goal structure to allow different runs

Runs and Answers

- **The spreadsheet "Project Cases.xls" on Moodle shows the cases I ran initially**
  - 17 combinations of budget, scenario probabilities, and weights on attrition and days
  - This was more of an "exploratory analysis" to see broad trends
- **Large variations in answers**
  - 10 – 22 days for campaign, 16 – 24 aircraft lost for MTW-2
  - GPS bomb buys range from 0 – 2788
- **But some things don't change ...**
  - We never buy any new unguided weapons
  - Little variation in MTW-1 days for campaign, SSC attrition
- **Overarching conclusion: how much do you want to spend to improve MTW-2 outcomes?**

Some Questions for You ...

- **What other runs would you make?**
- **How would you present the results?**
- **Can you modify the model to compute worst-case probabilities for the scenarios?**
  - Note that the “worst case” depends on weights on the goals
  - So you could have *multiple* worst cases
  - Also, suppose each scenario had to have a minimum probability in the worst case. Any idea how to do that? (Ask me next semester)
- **Finally, this project, though dated, is *very* realistic**