

2012 NDIA Joint Armaments Conference

Goal-Programming and Traditional Force-on- Force Simulations

Jon Peoble

Jim Rodrigue

May 16, 2012

Agenda

- Goal Programming
 - Concepts
 - Defining Optimal Mixes
 - Limitations

- Constructive Simulations

- Connectivity to Constructive Simulations

- Summary

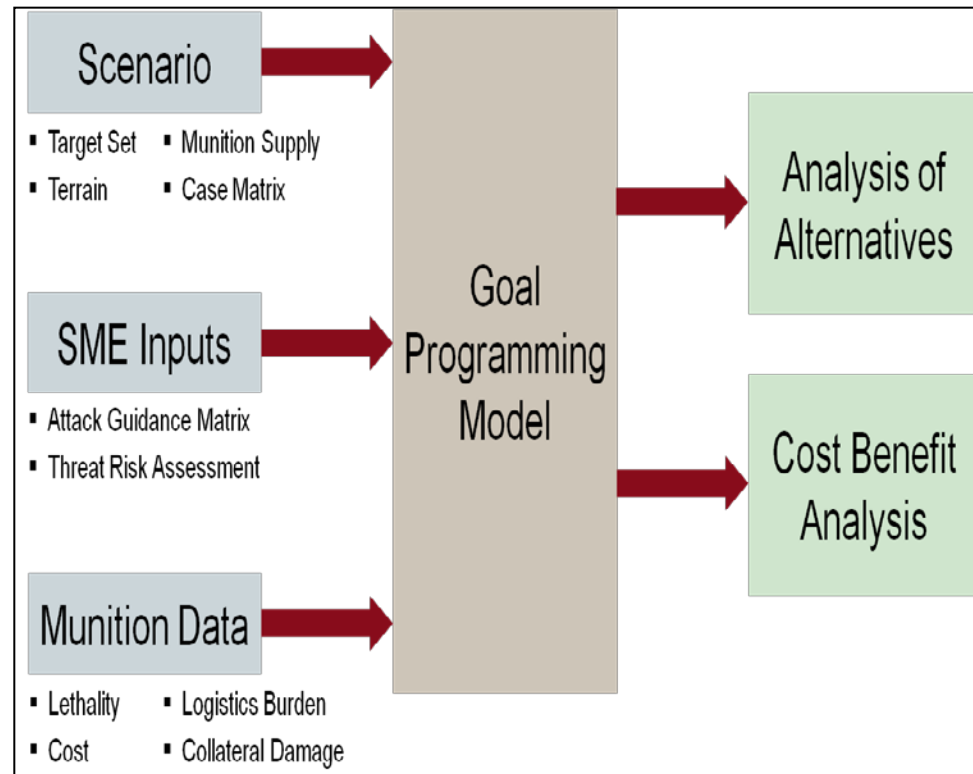
Goal Programming Model

■ Goal-Programming (GP) Model

- Math based, linear programming
- Model analyzes alternatives based on a set of goals and constraints:
 - Cost, lethality, collateral damage, logistics, etc.
 - Weights determine the importance of each goal
- Model identifies the “optimal” solution for the given inputs
 - Eliminate as many targets as possible while adhering to goals and constraints

■ Benefits of GP Model

- Very quick
- Ability to quantify a large number of metrics
- Ability to analyze large case matrices
- Scenario definitions are flexible and easily modified

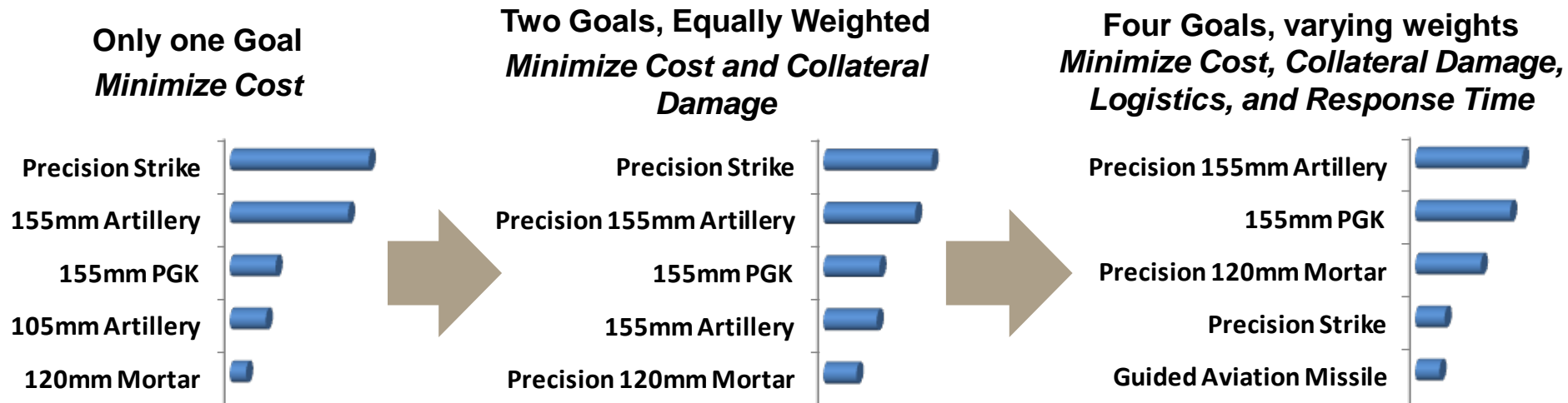


Goal-Programming results in different “optimal” solutions based on the user-defined scenario, goals, and constraints

Identifying Optimal Mixes

- Model determines an optimal solution based on the user-defined goals
 - The “weights” determine the importance of each goal
 - Weights must be matched to real-world considerations to produce valid solutions
- Constraints can be placed on the model to enforce scenario-specific limitations / preferences on munitions
 - Example: Prevent a munition from firing on certain targets due to Commander Intent / preference
- Below illustrates how goals, their weighting, and constraints can drastically change the optimal solution determined by the model

Top 5 Munitions in Different “Optimal Mixes”



Optimal mixes generated by the model vary greatly based on the inputs

Limitations of GP

- Model very subject to “garbage in, garbage out”
 - Answer only valid for given inputs
 - If inputs are incomplete or weighted incorrectly, answer is meaningless
 - Example: Defeat of a structure target

100 conventional
155mm rounds



Single Precision
Rocket



Optimize Cost

100 x \$1000
\$100,000



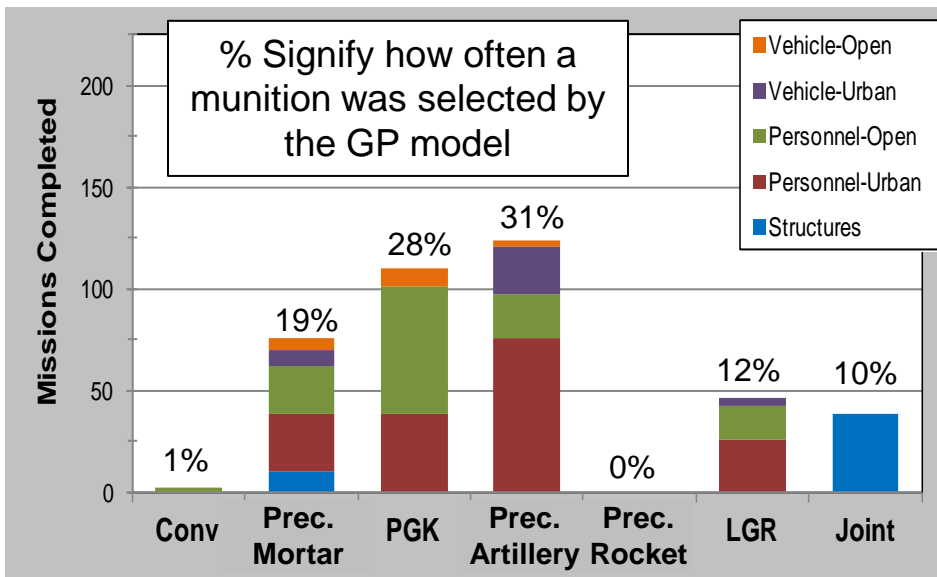
1 x \$110,000



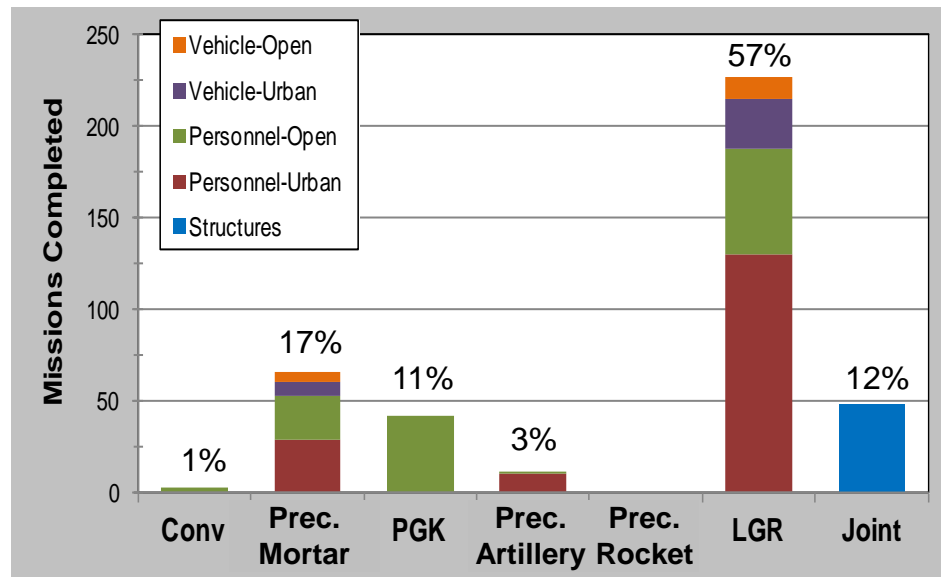
This is a correct
answer but is it
valid?

Limitations of GP

- Leaving out the slightest of considerations in GP inputs can have drastic repercussions in the outcome
- Example:
 - Not giving any weight to the goal of reducing “response time”
 - Analysis examines effects of adding a 2.75 Laser Guided Rocket (LGR) to the precision munition portfolio
 - LGR would be launched from RW; “response time” penalty for the RW to get on station



Results if “Minimize Response Time” goal is given modest weighting of 10%



Results if “Minimize Response Time” goal is not included in the model

Limitations of GP

- How do you generate realistic results in a GP model?
 1. Ensure as many real-life considerations as possible are modeled as goals
 - Logistics, collateral damage, response time, exposure, etc
 - Ensure weights match tactical considerations for specific scenario
 2. Include “Commander Preference” as a goal to force tactically sound choices
 - SMEs pair munitions and targets beforehand based on their expert experience
 - Model gives preference to these pairings
 - Problem: Can introduce biases into “optimal” mixes
 - If all considerations are represented as goals, Commander Preference is not required; let the model determine optimal pairings
- Very difficult to model operational benefits in GP model
 - How much better does this optimal mix make my force?
 - Survivability, op-tempo, etc

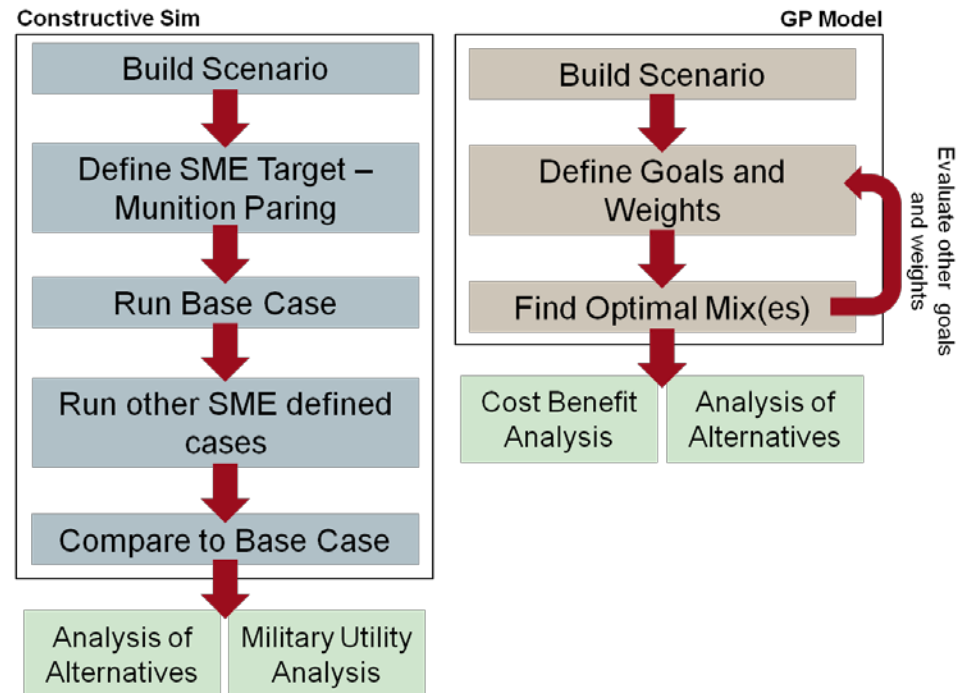
Goals, their weighting, and model constraints must all match real-life considerations to produce valid mixes

Constructive Simulations

- Constructive Simulations can provide the operational benefits piece that is missing from Goal Programming
 - Quantifies the benefits to the force in a force-on-force scenario
 - Often labor and time intensive to conduct analysis with

- Require a munition – target pairing for the scenario
 - Traditionally provided by SMEs
 - Define ideal munition pairings
 - Simulation adheres to pairing as close as possible while taking all dynamic battlefield states into consideration
 - Availability, organizational hierarchy, range, etc.

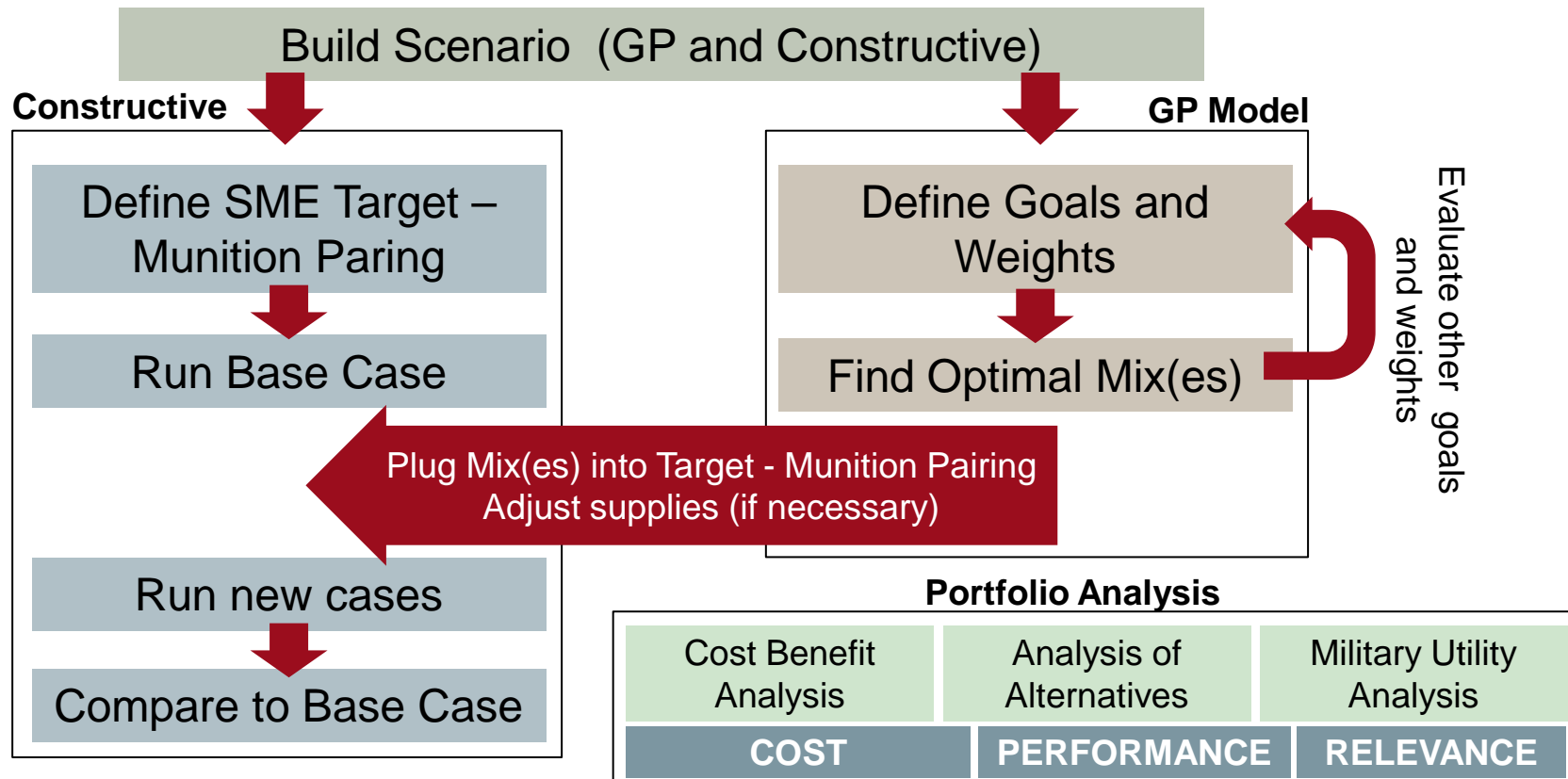
- How do we combine this with GP models to produce a portfolio analysis?



Constructive Simulations can define the operational benefits but traditionally, these model are run separately from GP Models

Linking GP Models to Constructive Simulations

- Proposal: GP Models replace the SME provided munition-target pairing
 - Examine the effects of the “optimal” mixes
 - Munition supply loads may also need to be adjusted



Integrating the GP model with the constructive sim allows for the optimal mixes to be validated in a realistic force-on-force scenario

Summary

- Goal Programming models are efficient tools that allow one to quickly identify optimal mixes of munitions
 - Very dependent on valid inputs; scenario, goals, weights, and constraints
 - GP models should be allowed to optimize the mix, without any outside biases added into the analysis
- Constructive Simulations are powerful tools that allow one to quantify the operational benefits of systems to the force
 - Powerful but time and resource intensive for large case matrices
- Combining the two allows for the strengths of each to be applied to the analysis
 - Evaluate the optimal mixes defined by the GP model in the constructive sims
 - Reduces size of constructive simulation case matrix
- Raytheon is exploring this topic further with the Fires Center of Excellence and FireSIM

A comprehensive portfolio analysis should include integrated results from both goal-programming and force-on-force simulations

Questions?

Jon Peoble
Raytheon Missile Systems
520.545.7841
Jon.Peoble@raytheon.com

Jim Rodrigue
Raytheon Missile Systems
520.794.1349
jmrodrigue@raytheon.com