## Problem Statement

What mix provides the best trade-off between reward and costs of flying, while:

1. Completes satisfying demands
2. Meeting capacity constraints
3. Meeting ability requirements
4. Meeting platform payload feasible combinations
5. Meeting interoperability requirements
6. Meeting non-schedule overlap restrictions

This should be indicated by an integer variable specifying the mixes desired over time and for what demands they are available and best able given their ability.

## Sets

|  |  |  |
| --- | --- | --- |
| Platforms |  | Only 5 platforms exist |
| Payloads |  | Only 10 payloads exist |
| Demands |  | Only 20 demands exist |
| Time |  | Only 72 scenario hours exist |

## Parameters

|  |  |  |
| --- | --- | --- |
| Reward |  | *Reward* payoff for a platform i, payload j, satisfying a unique mission demand d. |
| Cost |  | General *Cost* associated with a platform I, payload j, satisfying a unique mission demandd. |
| Interoperability |  | *Boolean* interoperability matrix defining 1 if platform i is interoperable with payload j. |
| Demand |  | *Demand* of a specific mission demand at a specific time in the scenario. |
| Ability Throttle |  | *Ability Throttle* defining how able a payload j is at satisfying a mission demand d. |
| Back-order Penalty |  | Backorder *Penalty* associated with not having a payload j to satisfy a mission demand d. |
| Back-order Acceptance Delay |  | *Acceptance* delay for how many time steps in the future we wish to allow a backorder to fall. |
| Fixed Charge |  | *Fixed Charge*, if you decide to fly a platform I payload j, there is a one-time fee for doing so in this scenario. |

## Decision Variables

|  |  |
| --- | --- |
|  | 1 if platform i payload j at time t is used to satisfy a demand d, 0 otherwise. |
|  | 1 if platform i payload j is used ever. |

## Objective

|  |  |
| --- | --- |
| English | Maximize Reward – General Cost – Backorder Penalty – Fixed Charge |
| Math | Maximize - - - |

## Constraints

|  |  |
| --- | --- |
| Total demands a mix satisfied at a time t can only be > 0 if |  |
|  |  |
|  |  |
|  |  |