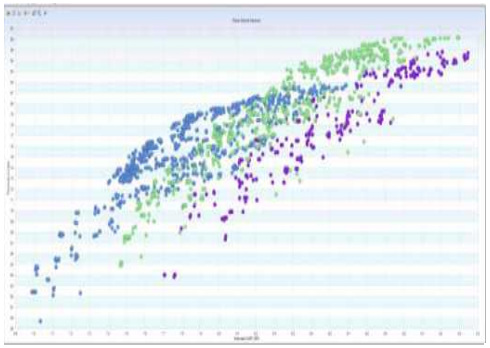
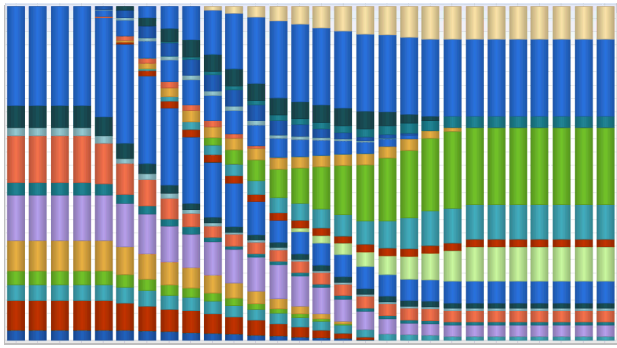
SAND2017-11444C





Multi-Objective Optimization for Task Mode Selection and Scheduling Utilizing a Genetic Algorithm

Alex Dessanti

INFORMS Annual Meeting

24 October 2017

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. 2017-XXXXP

Overview

- Problem Background
- Approach
- Capability Overview
- Results
- Conclusion

Challenge

Sandia National Laboratories

- Scheduling is the process of timing a series of tasks to complete a job
- Real-world problems involve many *competing objectives* with tradeoffs to be considered
- Often *many options* to achieve desired outcomes (both how and when)
- Creating schedules that meet delivery dates and minimize project duration with *limited resources* is a significant challenge
- Problem-specific formulations are effective, but require significant effort and optimization modeling expertise

Schedule Optimization is Important

Sandia National Laboratories

- Schedule optimization is impactful, spanning many industries
  - Defense – roll-out/stand-up of defense systems
  - Energy – execution of grid modernization plans
  - Homeland Security – scheduling resources for border monitoring
  - Cyber Security – deployment of IT system upgrades
  - Manufacturing – production lines, testing facilities
  - Airlines – flight schedules, fleet assignment, crew scheduling
  - Healthcare – nurse rostering, operating room surgical schedules

- Common scheduling objectives
  - Minimize cost
  - Minimize time to complete set of tasks

Schedule Optimization is Difficult

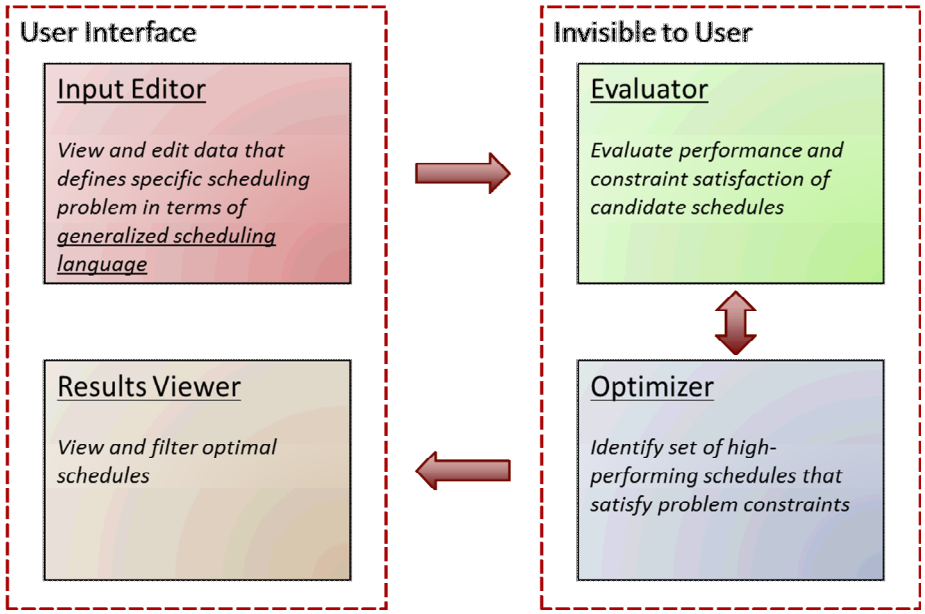
Sandia National Laboratories

- Scheduling problems have many characteristics
  - Resource constraints
  - Task precedence
  - Multiple ways (modes) to complete a task
  - Multiple concurrent projects can share resources
  - Inventory constraints
  - Production, consumption, and flow of materials
  - Multiple objectives to achieve

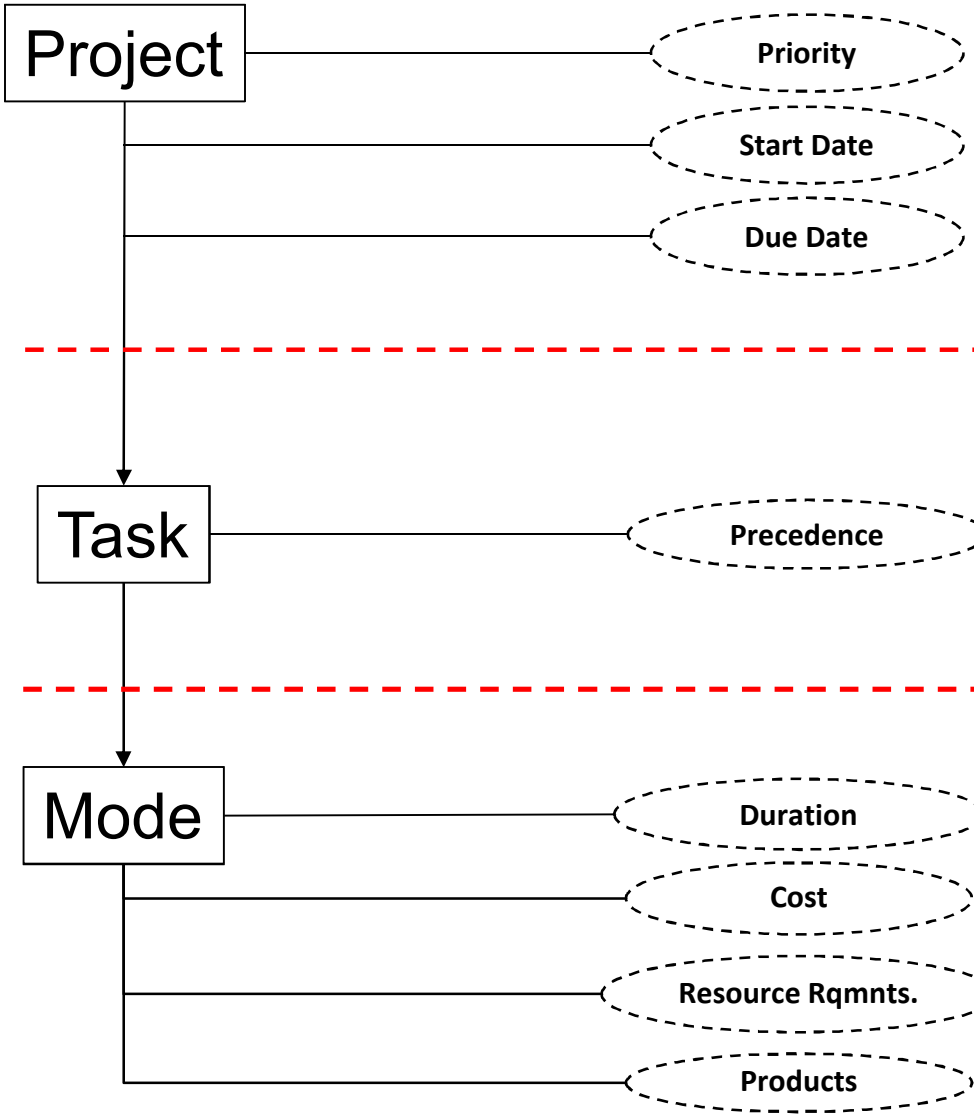
Generalized, user-friendly, extensible schedule optimization tool **for handling such challenges would benefit multiple customer sets**

Our Approach

- Identify generic language for defining scheduling problems
- Develop evaluator to assess multi-objective performance of candidate schedules
  - Minimize cost
  - Minimize time to complete a set of tasks
  - Minimize schedule risk (e.g., critical activity timing)
  - Maximize impact of activities (e.g., requirements coverage)
- Develop innovative genetic algorithm (GA) for schedule optimization
  - Standard GAs struggle with highly-constrained problems (e.g., scheduling)
  - Build on latest research in scheduling GAs
- Build interface to analyze/filter/view results
  - Enables effective communication of insights, tradeoffs, and courses of action



Generalized Scheduling Language

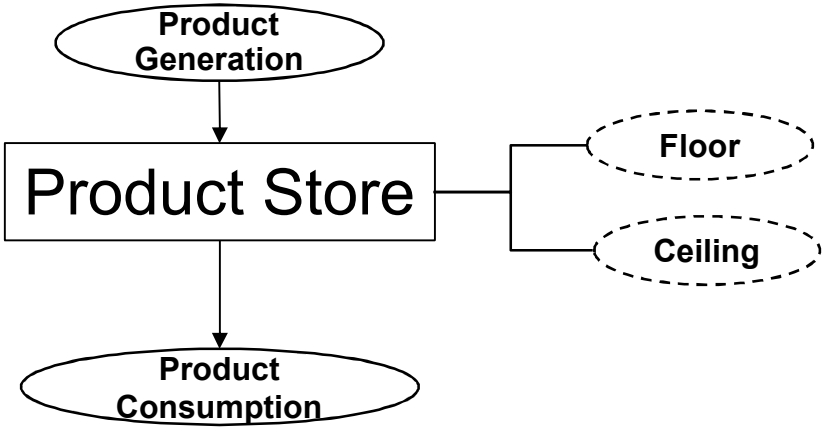


Project: Set of tasks that changes state of something
- Unproduced to produced product
- Untested to tested system
- Unconstructed to constructed facility

Task: Activity that needs to be done
- Pour foundation
- Assemble sub-system
- Apply coating

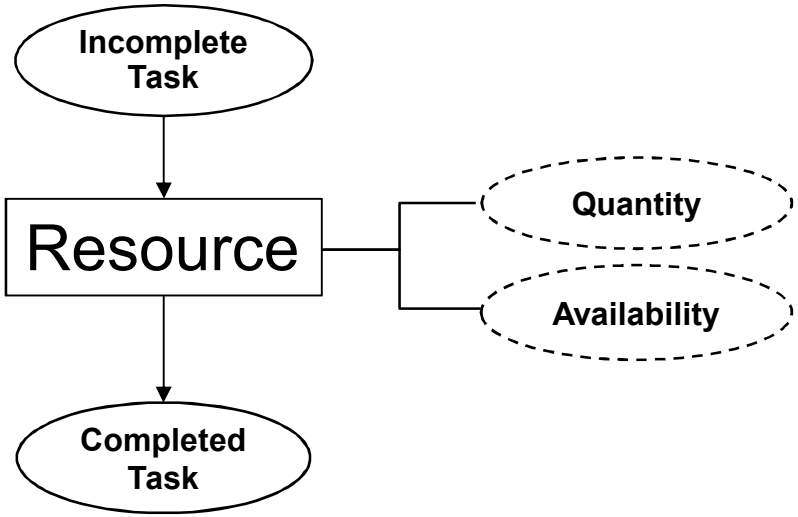
Modes: Different ways of performing tasks
- Perform task with {1, 2, 3} personnel
- Test using {machine 1, machine 2}

Additional Constructs



Product: Something that can be generated or consumed by a task

- Finished, yet-to-be inspected products
- Tank of unburned fuel



Resource: Something required to complete a task that becomes available for use by other tasks after task finishes

- Machines
- Facilities
- Personnel

Optimization Algorithm Selection

Sandia National Laboratories

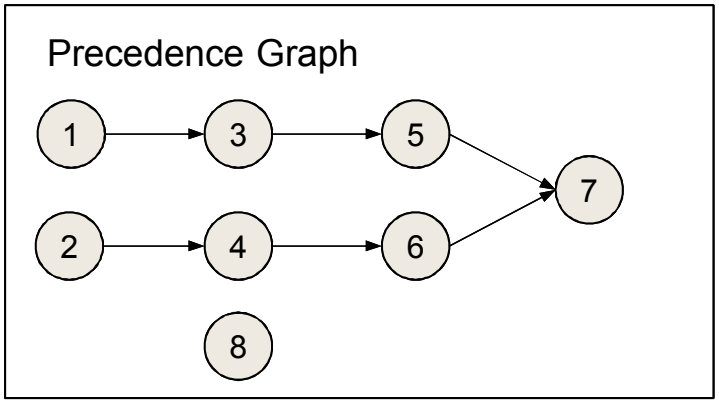
Consideration	Genetic Algorithm	Branch and Bound
Restrictions on functional form of objectives and constraints	No restrictions, can be highly non-linear	Must be linear (non-linear can sometimes be accommodated)
Produces results along multiple objectives	Up to ~8 objectives	~2 objectives (using weighted sum or ε-constraint methods)
Indicator of true optimality	None	Gap
Handles continuous variables	Best for discrete/categorical variables	Handles continuous and discrete quantitative variables
Parallel evaluations	Yes	Yes

Using a GA for Scheduling

Sandia National Laboratories

- Evaluator and optimizer work together to identify set of high-performing schedules in terms of defined objectives
- Genetic algorithm decides task priority ordering and mode selections (how)
- Task ordering from GA obeys precedence constraints
- Evaluator schedules tasks (when) as early as possible without violating resource availability and precedence based on ordering from GA
- Specialized initialization, crossover, and mutation operators are defined that modify task ordering without violating precedence

Initialization Operator (single-mode)



Traditional Initialization

- Choose a random element and add it to gene

C1: [1 2 5 3 4 6 7 8]

Some tasks may appear in an infeasible order

Schedule GA Initialization*

- Identify all remaining tasks in precedence graph with no predecessors
- Randomly select one and add it to end of ordered task list
- Remove selected task from precedence graph
- Repeat until all tasks have been placed

C1: [1 2 3 5 4 6 7 8]

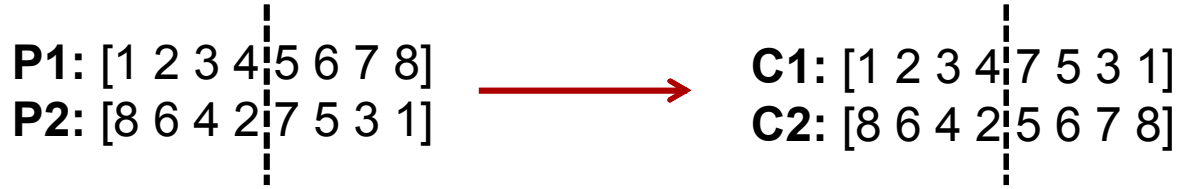
Each task appears in precedence feasible order

*Adapted from K.S. Hindi, H. Yang, K. Fleszar, "An Evolutionary Algorithm for Resource-Constrained Project Scheduling," *IEEE Trans. Evolutionary Comp.*, vol. 6, no. 5, pp. 512-518, 2002.

Crossover Operator (single-mode)

Traditional Crossover

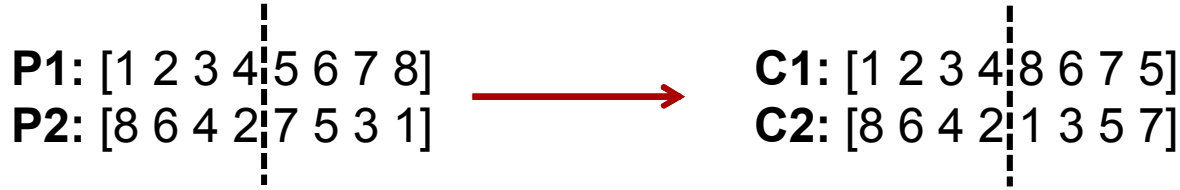
- Given random crossover point *q*, positions 1,…,*q* copied from parent 1
- Remaining positions copied from elements *q*+1,…,*n* of parent 2



Some tasks may appear twice, others not at all

Schedule GA Crossover*

- Given random crossover point *q*, positions 1,…,*q* copied from parent 1
- Remaining positions are filled from parent 2 in order, but tasks already taken from parent 1 may not be considered again



Each task appears exactly once, precedence is preserved

*Adapted from S. Hartmann, "Project Scheduling with Multiple Modes: A Genetic Algorithm," *Annals of Operations Research,* vol. 102, pp. 111-135, 2001.

Mutation Operator (single-mode)

Sandia National Laboratories

Traditional Mutation

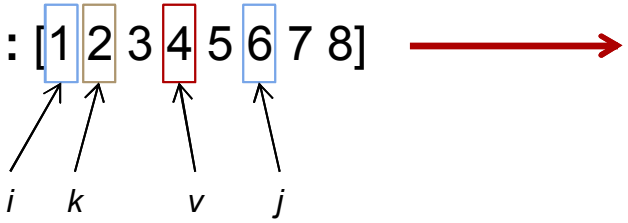
- Choose a random element and replace it with different value

C1: [1 2 3 4 5 6 7 8] → **C1:** [1 2 3 4 5 3 7 8]

Some tasks may appear twice, or not at all

Schedule GA Mutation*

- Choose a random element *v* and identify its latest predecessor *i* and earliest successor *j*
- Choose a random point *k* between *i* and *j* and move *v* to that position (slide other elements toward *v*'s previous position)



- **4** moves to the position previously occupied by **2**
- **2** and **3** slide towards position previously occupied by **4**

Each task appears exactly once, precedence is obeyed

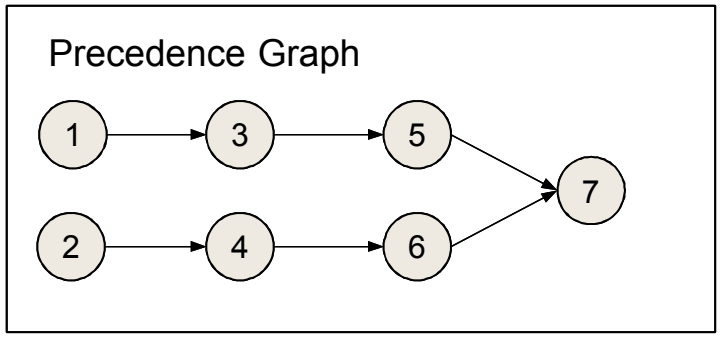
*Adapted from K.S. Hindi, H. Yang, K. Fleszar, "An Evolutionary Algorithm for Resource-Constrained Project Scheduling," *IEEE Trans. Evolutionary Comp.*, vol. 6, no. 5, pp. 512-518, 2002.

Simple Example

7 tasks (single mode), 1 resource type:

Task ID	Resource Requirement	Duration	Predecessor(s)
1	2	2	-
2	2	3	-
3	3	2	1
4	3	2	2
5	2	4	3
6	1	5	4
7	2	3	5, 6

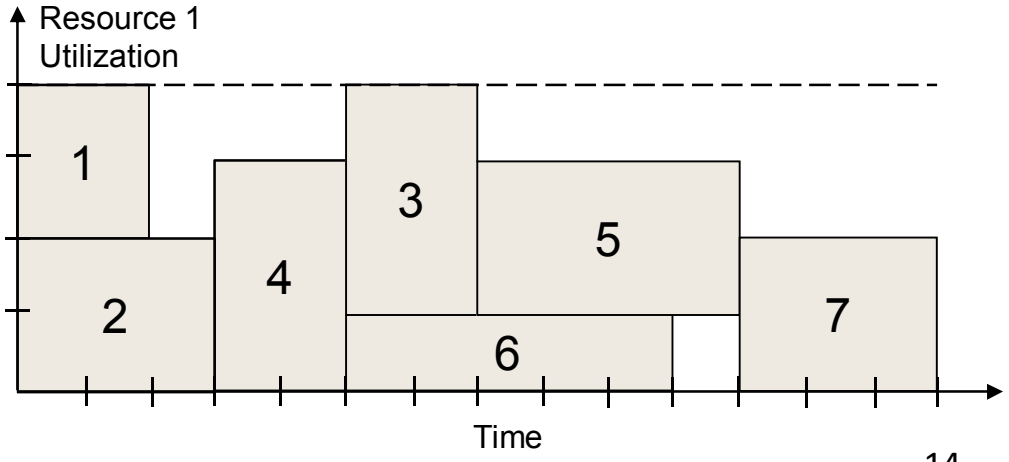
Resource ID	# Units
1	4



Given precedence-feasible ordering from GA…

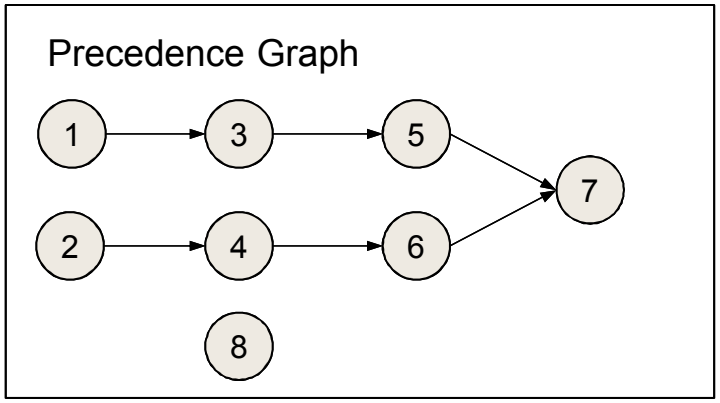
[2 4 1 6 3 5 7]

…evaluator builds…

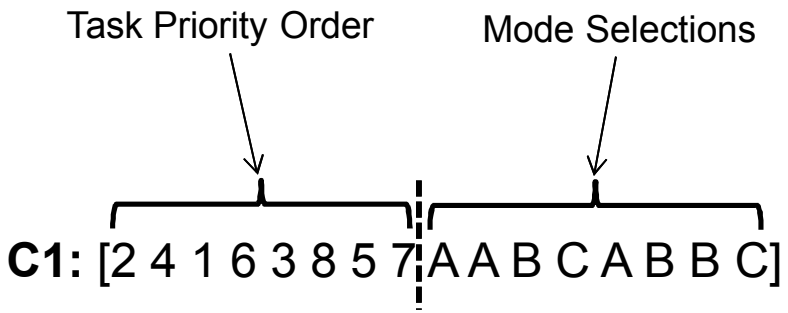


Extension to Accommodate Multiple Modes

- 1st half of solution's encoding represents task priority order
- 2nd half of solution's encoding represents mode selections
- Precedence-aware initialization, crossover, and mutation used for task priority order
- Traditional initialization, crossover, and mutation used for mode selections



Example: 8 tasks, 3 mode options each



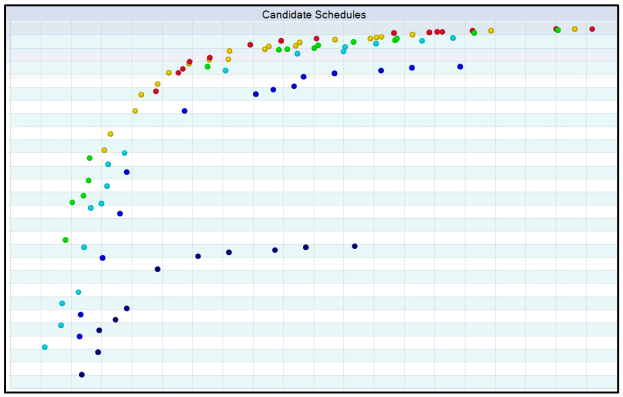
Schedule Management Optimization (SMO)

Sandia National Laboratories

- **What is SMO?**
  - *General-purpose scheduling optimization capability* developed by Sandia
  - Identifies actionable solutions to resource-constrained scheduling problems
- **Benefits of using SMO**
  - Allocates resources and assigns activity start times to minimize tardiness, maximize robustness, and use resources efficiently
  - Enables users to define complex scheduling problems in terms of simple, generic language
- **Different from commercial project management software**
  - Scheduling software typically requires users to set activity start times and resource allocations manually, SMO does this automatically
  - Can model and optimize complex scheduling issues commercial packages handle poorly or not at all
- **Different from other optimization approaches**
  - SMO can be *rapidly applied to a broad range of multi-objective scheduling problems* with little to no software or formulation changes

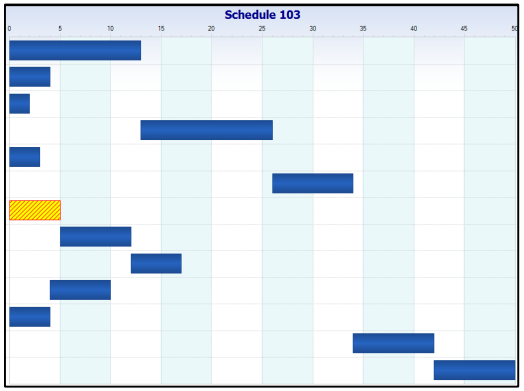
Results and Insights

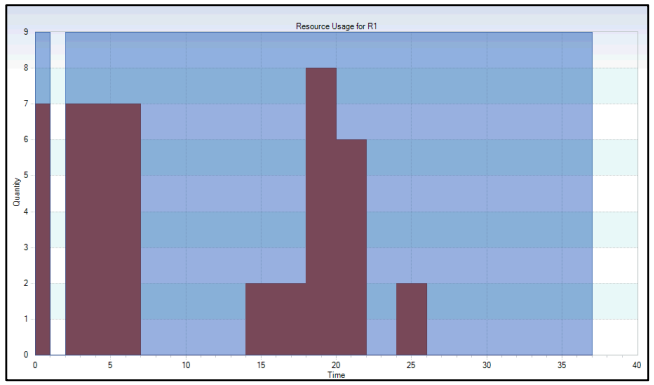
- Set of Pareto-optimal schedules enables comparison of courses of action and understanding of tradeoffs



- Gantt charts display detailed information about what gets done when and how

- Resource utilization plots inform decisions about investment in additional personnel, equipment, or facilities





Summary

- Challenge
  - Creating schedules that meet delivery dates and minimize project duration with limited resources is a significant challenge
- Original Goal
  - Develop user-friendly, general-purpose scheduling optimization capability
  - Identify actionable solutions to resource-constrained scheduling problems
- Impact of SMO
  - Solves a general class of *multi-objective scheduling problems*
  - Allows for *rapid adaptation and deployment* to multiple application domains
  - Generates optimal candidate plans using a *GA with specialized operators*
  - Enables *exploration of potential schedule solutions to understand tradeoffs* between cost, makespan, robustness, and resource utilization

Sandia National Laboratories

QUESTIONS?

Sandia National Laboratories

BACKUP

Why not use commercial project management tools?

Sandia National Laboratories

Scheduling Challenge	MS Project/Primavera P6	SMO
Resource constraint resolution	Limited; uses heuristics to resolve resource conflicts (resulting schedules are feasible, but not optimal)	Optimizes task start times such that resources are used as effectively as possible
Choose from multiple task modes	Unable	Optimizes mode selections
Optimize multiple objectives	Unable; can only minimize span time	Identifies tradeoff between competing objectives (span time, lateness, resource load level, robustness, cost)
Account for stored entities	Limited; can compute total expenditures using resource costs, but can't model rising/falling inventories with upper and lower limits	Models changing inventories of any desired entity, such as funds or materials, along with constraints on upper and lower inventory limits
Account for time-varying resource requirements of a task	Limited; user can specify percentage of a resource required to complete a task, but can't specify exactly when that resource is needed during task execution	User can say exactly when a resource is needed throughout the duration of a task (e.g., needed at beginning/end for setup and cleanup, but not in middle)
Determine resource allocations as percentage of FTE	Not optimized, however user can manually specify resource allocation to tasks as a percentage	Optimizes resource assignment to tasks; project percent allocations are calculated in post-processing if desired

SMO Framework