

IAAI, July 2014

A Schedule Optimization Tool for Destructive and Non-Destructive Vehicle Tests

Jeremy Ludwig, Annaka Kalton, and
Robert Richards

Stottler Henke Associates, Inc.

Brian Bautsch, Craig Markusic, and J.
Schumacher

Honda R&D Americas, Inc.

Overview

- Inspirational Video
- Introduction
- Scheduling Framework
- Scheduling UI
- Domain Customization
 - UI
 - Scheduler
- Methods
- Results
- Conclusion

Inspirational Video

Testing Conducted by
Vehicle Research and Test Center at
Transportation Research Center Inc.

Improved Restraints in Rollovers
CONTRACT NO. DTNH22-07-D-00060

April 20, 2010

Introduction

- Create a schedule for testing new and refreshed vehicles
 - Test vehicles hand-built
 - Project end date defined externally
 - Limited personnel and facility resources

Aurora Scheduling Framework

Given a list of tasks (or jobs or activities) each with a set of required resources and constraints, assign resources to tasks (for specific time windows)

- Heuristic-based scheduling framework
- Customized for domain
- E.g. *Minimize* the number of vehicles required while still completing project on time

Boeing Airplane Assembly Scheduling



- Very large, complex models
 - Large numbers of resource contentions, constraints
- Widely distributed users working on different projects
- Part of integrated management system
 - Accepts inputs from modeling system, sends outputs to shop floor management system

Medical Resident Scheduling

- Allocating residents for hospital staffing and educational purposes

- 150+ residents must be scheduled for a full year

- Extensive rules provide flexible constraints for an acceptable schedule

Residents

Rotations

Annual Rotations Schedule							
	Vacations	July	August	September	October	November	December
Jason PGY-3		County	Elective	Rm 1322	Surgery	Peds	ER
Jason PGY-3	1	ER	County	Rm 1322	Rm 1322	County	Electiv
Jenn PGY-3	1 2	Rm 1322	VA	Peds	ER		
Jonathan PGY-3		Surgery	County	VA	County		
Anastice PGY-4		Peds	Elective	Elective	ER	County	Surgery
Murrieta PGY-4	1					Surgery	
Nicolas PGY-4	(none)		Surgery		Peds	Elective	Elective
Select Residents							
County		Jason	Jason	Elective	Anastice	James	Anastice
Elective		Jason	Anastice		Nicolas	Nicolas	Jason
ER		Jason		Anastice	Jenn		James
Peds		Anastice		Jenn	Sarah Nicolas	James	
Rm 1322		Jenn		James Jason	Jason		
Surgery		Jonathan	Nicolas		James	Anastice	

Blocks

Space Station Processing Facility Scheduling

A large, cylindrical space station component, possibly a module or airlock, is being processed in a clean room. The component is covered in white protective sheeting and has various cables and hoses attached. Three workers in white protective suits and masks are visible, working on the component. A yellow caution sign is visible on a railing in the foreground. The facility is a large, industrial building with high ceilings and metal railings.

At NASA's Kennedy Space Center, Aurora schedules the use of floor space and other resources at the Space Station Processing Facility, the world's largest low-particle clean room where Int'l Space Station components are prepared for flight.

Managed Intelligent Deconfliction And Scheduling (MIDAS)

Performs automated resource assignment, scheduling, and deconfliction for Defensive Space Control and Space Situational Awareness operations.

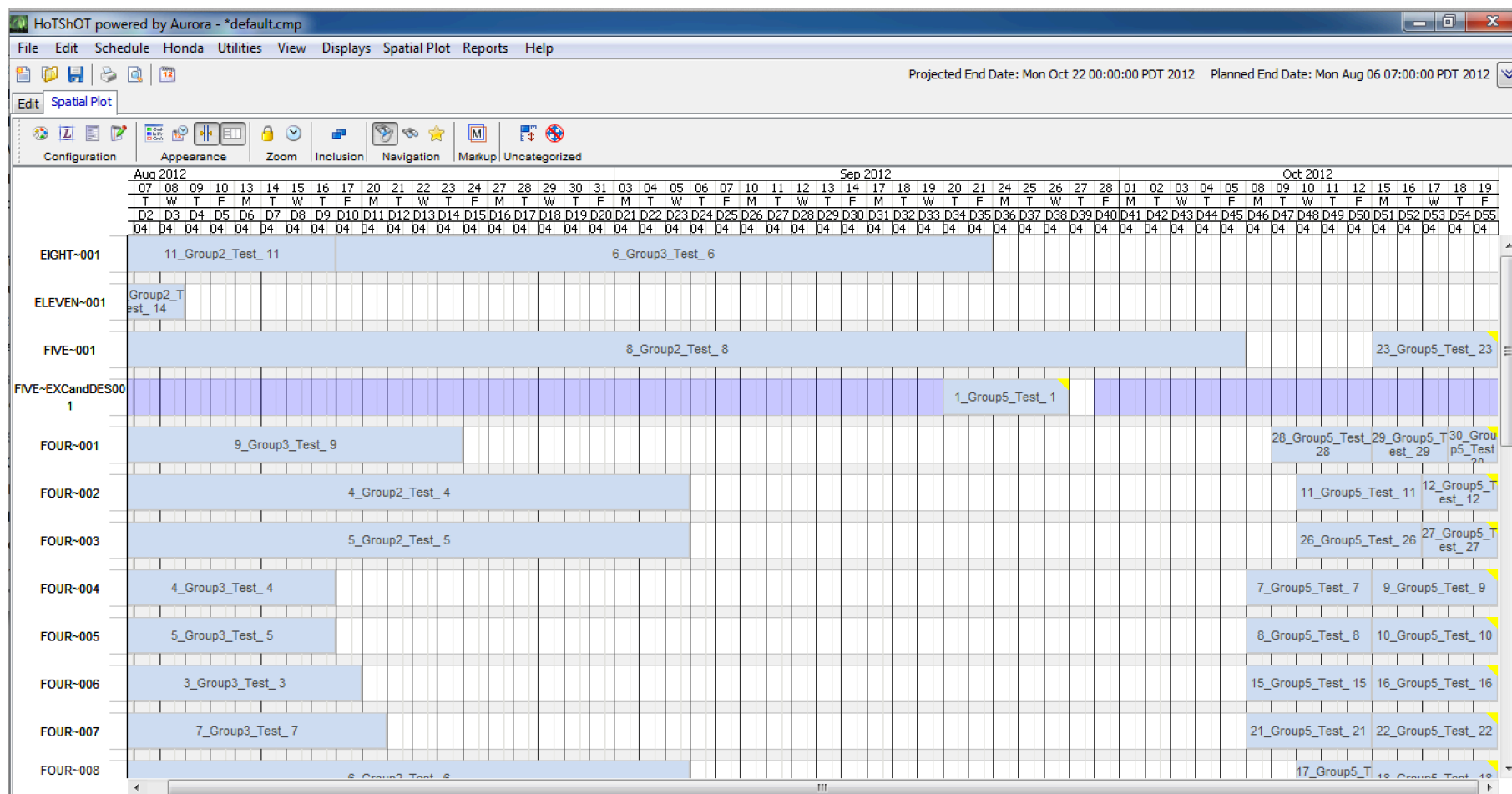
Scheduling UI

The screenshot displays the HoTShOT software interface, titled "HoTShOT powered by Aurora - *default.cmp". The menu bar includes File, Edit, Schedule, Honda, Utilities, View, Displays, PERT Chart, Reports, and Help. The toolbar contains icons for file operations and view toggles.

The interface is divided into three main sections:

- Left Panel (Project Tree):** Shows a hierarchical view of the project. The "Test" folder is expanded, listing various test groups (e.g., 1_Group1_Test_1, 2_Group1_Test_2, etc.).
- Center Panel (Property Window):** Displays the properties for the selected item, "ID: 1_Group1_Test_1". It includes tabs for "Actuals", "Flags", "Constraints", "Properties", "Schedule Attributes", "Analysis", "Schedule Results", and "Requirements". The "Schedule Results" tab is active, showing fields for "scheduled order", "explanation", "scheduled duration", "early start date", "start date", "end date", "late end date", "flow start", and "flow end".
- Right Panel (Network Diagram):** Shows a PERT chart or network diagram. Nodes represent tasks, and arrows indicate dependencies. The diagram is organized into columns, with tasks like "2_Group5_Test_2", "3_Group5_Test_3", "4_Group5_Test_4", etc., connected by red and blue arrows.

The bottom of the interface features a toolbar with icons for "Configuration", "Appearance", "Arrangement", "Zoom", "Inclusion", and "Navigation".



Scheduling Framework

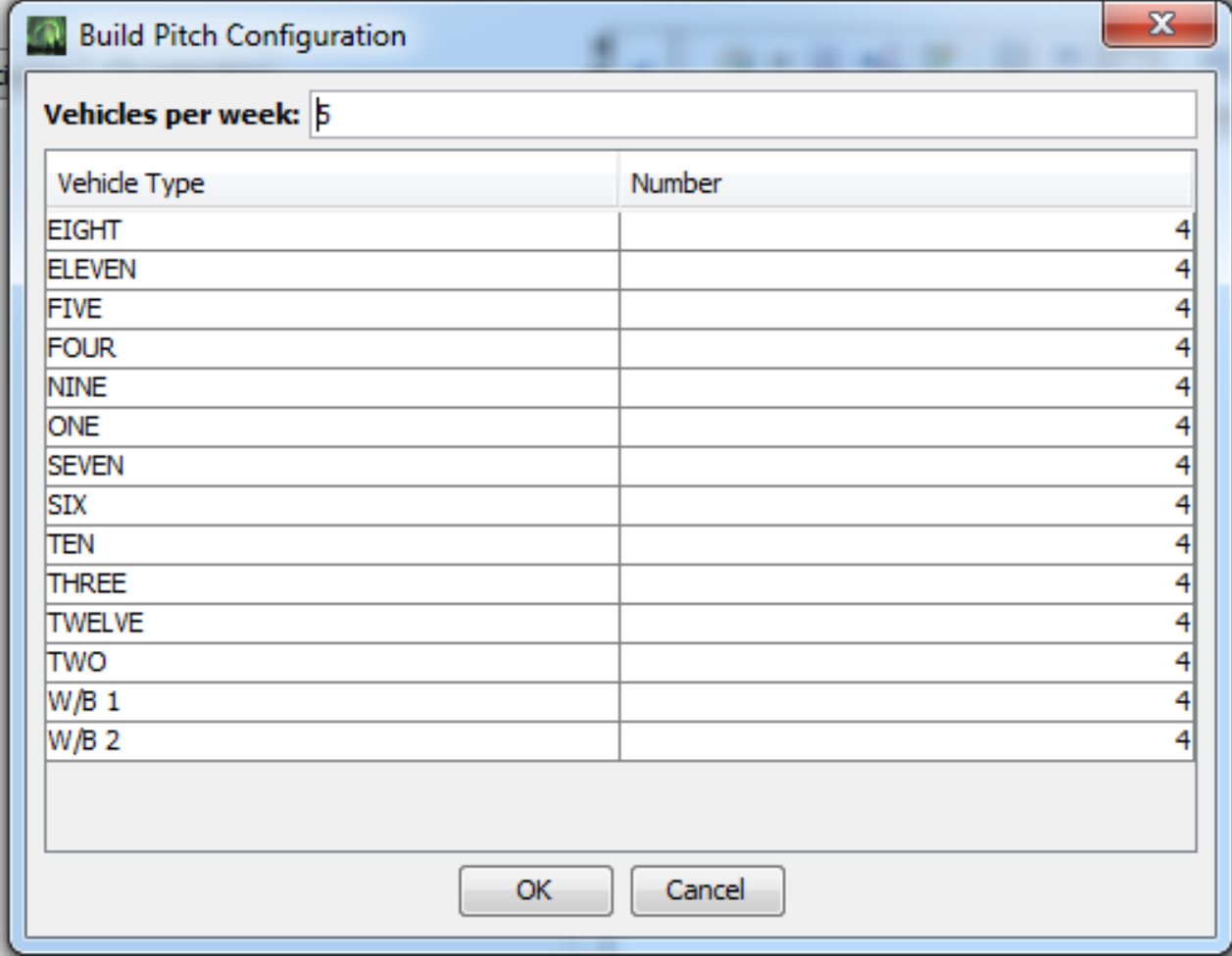
- Schedule Initialization
 - Preprocessor
 - Queue Initializer
 - Prioritizer
- Scheduling Loop
 - Scheduler
 - Quality Criterion
 - Conflict Manager
- Schedule Finalization
 - Postprocessor



Domain Specific Customization

- User Interface
 - Build Pitch
 - Manage Vehicles
 - Optimization Dashboard
- Scheduling Components

Build Pitch



The image shows a software dialog box titled "Build Pitch Configuration". At the top, there is a text field labeled "Vehicles per week:" with the value "5" entered. Below this is a table with two columns: "Vehicle Type" and "Number". The table lists 15 vehicle types, each with a corresponding number in the "Number" column. At the bottom of the dialog, there are "OK" and "Cancel" buttons.

Vehicle Type	Number
EIGHT	4
ELEVEN	4
FIVE	4
FOUR	4
NINE	4
ONE	4
SEVEN	4
SIX	4
TEN	4
THREE	4
TWELVE	4
TWO	4
W/B 1	4
W/B 2	4

Manage Vehicles

Manage Vehicles

Flexible Start Vehicles

Fixed Start Vehicles

Vehicle	Build Order	Build Date	End Date
FIVE~001	1	08/06/2012 07:00	01/19/2058 00:00
SEVEN~001	2	08/07/2012 07:00	01/19/2058 00:00
SIX~001	3	08/08/2012 07:00	01/19/2058 00:00
THREE~001	4	08/09/2012 07:00	01/19/2058 00:00
EIGHT~001	5	08/10/2012 07:00	01/19/2058 00:00
TWO~001	6	08/13/2012 07:00	01/19/2058 00:00
FOUR~007	7	08/14/2012 07:00	01/19/2058 00:00
ONE~001	8	08/15/2012 07:00	01/19/2058 00:00
FOUR~001	9	08/16/2012 07:00	01/19/2058 00:00
FOUR~008	10	08/17/2012 07:00	01/19/2058 00:00
FOUR~010	11	08/20/2012 07:00	01/19/2058 00:00
FOUR~003	12	08/21/2012 07:00	01/19/2058 00:00
TEN~001	13	08/22/2012 07:00	01/19/2058 00:00
W/B 2~001	14	08/23/2012 07:00	01/19/2058 00:00
FOUR~002	15	08/24/2012 07:00	01/19/2058 00:00
NINE~001	16	08/27/2012 07:00	01/19/2058 00:00
FOUR~004	17	08/28/2012 07:00	01/19/2058 00:00
W/B 2~002	18	08/29/2012 07:00	01/19/2058 00:00
FOUR~009	19	08/30/2012 07:00	01/19/2058 00:00
W/B 1~001	21	08/31/2012 07:00	01/19/2058 00:00
W/B 1~002	22	09/03/2012 07:00	01/19/2058 00:00
ELEVEN~001	23	09/04/2012 07:00	01/19/2058 00:00

↑

↓

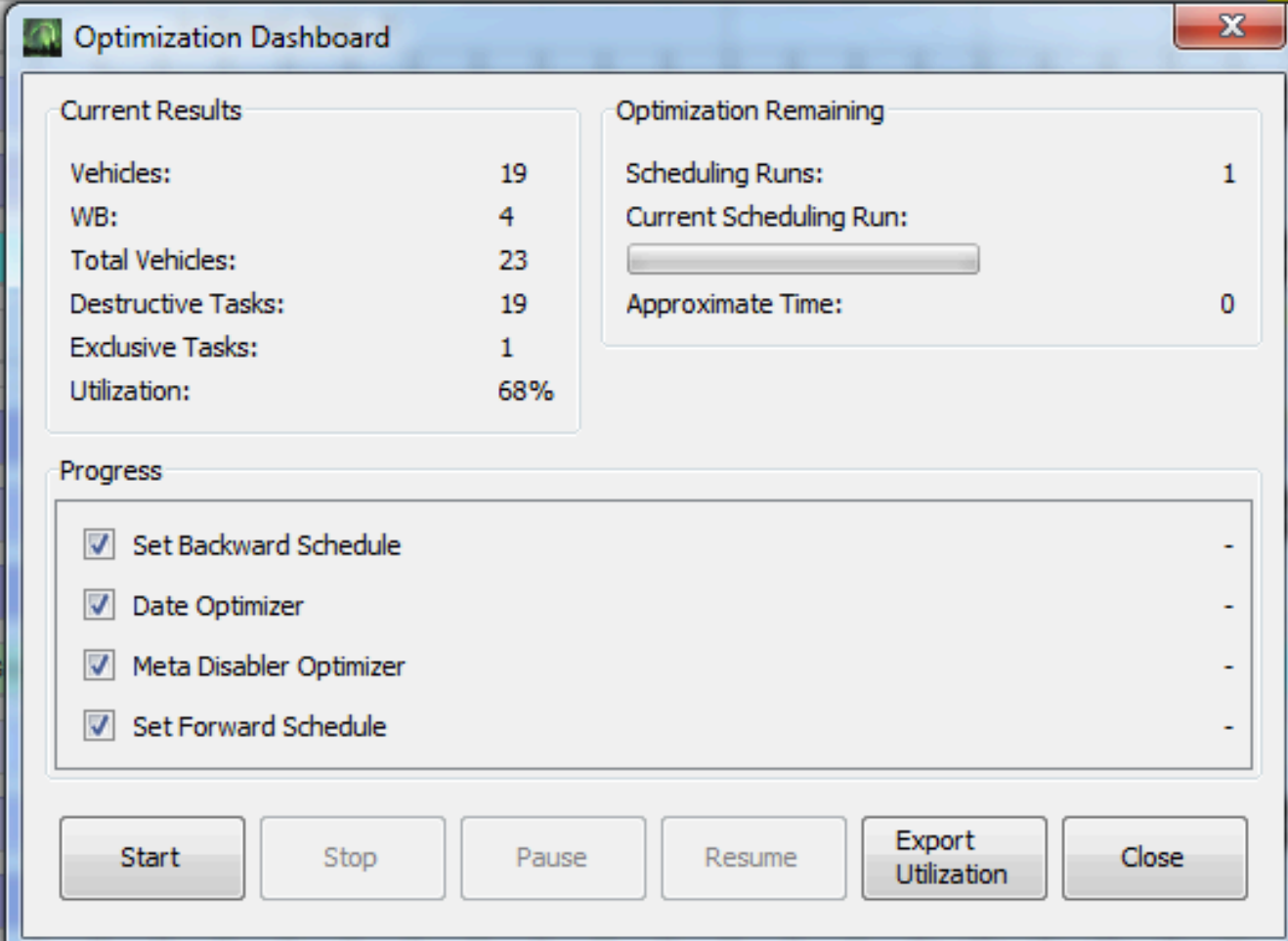
Recalculate Build Dates

Import Build Dates

+ Add Vehicle

Remove Vehicle

Optimization Dashboard



The Optimization Dashboard window displays current results, optimization remaining, progress, and control buttons.

Current Results

Vehicles:	19
WB:	4
Total Vehicles:	23
Destructive Tasks:	19
Exclusive Tasks:	1
Utilization:	68%

Optimization Remaining

Scheduling Runs:	1
Current Scheduling Run:	<div></div>
Approximate Time:	0

Progress

<input checked="" type="checkbox"/> Set Backward Schedule	-
<input checked="" type="checkbox"/> Date Optimizer	-
<input checked="" type="checkbox"/> Meta Disabler Optimizer	-
<input checked="" type="checkbox"/> Set Forward Schedule	-

Buttons: Start, Stop, Pause, Resume, Export Utilization, Close

Scheduling Component Customization

- Schedule direction management
 - Preprocessor, Postprocessor
- Support for exclusive tasks, destructive tasks, and task series
 - Preprocessor, Prioritizer, Scheduler, Postprocessor
- Heuristic Tuning
 - Preprocessor, Prioritizer, Quality Criterion, Postprocessor

Methods

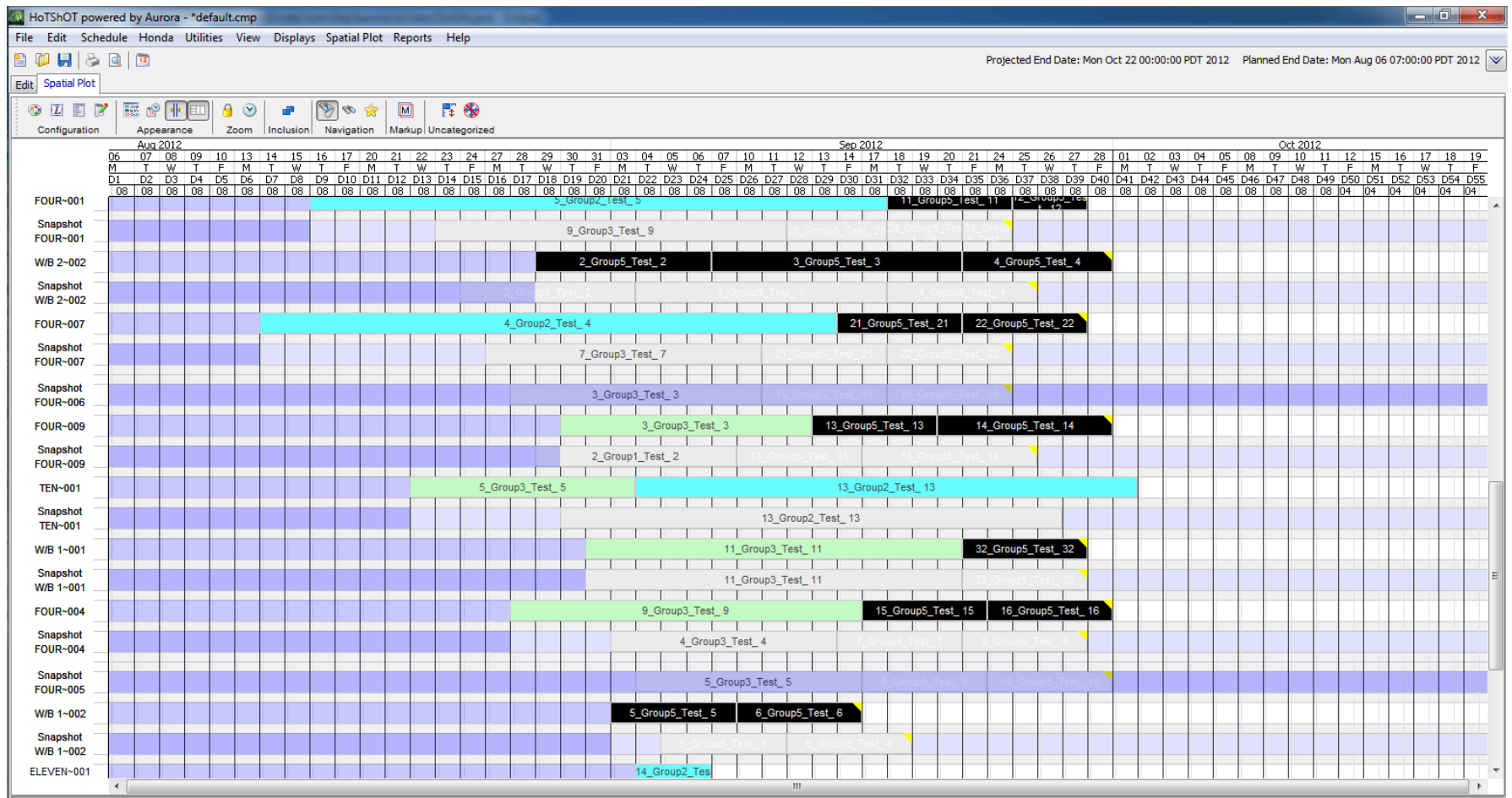
- Test Model
 - 60 tasks
 - 18 destructive, 1 destructive and exclusive
 - 680 days work over 55 calendar days
 - Fixed build pitch with 1105 possible work days
- Lower Bound
 - 22 vehicles
 - 19 destructive tasks, 3 specific vehicles required that do not match destructive tasks
- Manual Solution
 - 25 vehicles

Results

- Aurora Solution
 - Round 1: 22 Vehicles
 - Too good!
 - Round 2: 23 Vehicles
 - 8% reduction in vehicles
 - Withstood scrutiny
- Schedule created in 2 minutes from model vs. days of labor
 - Spend this time using 'What-if' capability to try and further improve the schedule

[illegible]

Comparing Schedule Snapshots



Conclusion

- Complex, real-world, scheduling problem
- Added domain-specific heuristics to a general intelligent scheduling framework
- Generated schedule for vehicle testing
 - with a significant reduction in the number of vehicles required
 - that still completed in the given timeframe

Ongoing Work

- Testing on more complex models that require over 100 vehicles
- Utilizing facility and personnel constraints when creating a schedule
- Supporting the transition of the software into the hands of the actual planners