Beyond the Job-Shop: Scheduling in the Real World

Mark Boddy, Ph. D.

mark.boddy@adventiumlabs.org

http://www.adventiumlabs.org/People/Boddy/index.html

Todd Carpenter

todd.carpenter@adventiumlabs.org

http://www.adventiumlabs.org/People/Carpenter/index.html

Tutorial Overview

Tutorial Objective: Attendees should come away understanding complexities inherent in several real-world scheduling domains

- Background & Terms
- Domain Abstractions and the Real World
  - Aircraft and Avionics
  - Space
  - Manufacturing & Processing Industries
  - Sports Leagues

Classic Job Shop Scheduling

- A set *J* of jobs to be run,
- each with a set of tasks to run in sequence,
- each on some unique element of
- a set *M* of machines

Common problem statements:

- Find a feasible schedule
- Minimize makespan
- Given a set of deadlines, minimize tardiness

Example Complications

- Choice of resource
- Other resource constraints
- (Global) capacity constraints
- State constraints
- Inter-activity constraints
- Consumable resources
- Complex temporal constraints (latency, preemption, hierarchical activity relationships)
- System dynamics (flow rates, chemical composition, mixing)

Schedule Usage

- Schedules are generated to satisfy existing plans
- Schedules are generated in a historical context
- Schedules are usually not constructed completely automatically
- Schedules frequently require input from multiple parties
- Schedules are used in operations:
  - Schedules require updating as events occur
  - Rescheduling can be required when conflicts are detected
  - Post mortem examinations
- Schedules are persistent artifacts
- Schedules are large
- Optimization is important, but hard to formalize properly

Commonly Desired Features

- Incremental modifications, as information comes back from plant operations
- Conflict detection and (local) rescheduling
- Feasibility determination and “culprit identification”
- Mixed-initiative operations
- Integration with plant/corporate data systems
- Things you need in any case:
  - Efficiency
  - Flexibility
  - Good user interface

Results Sell, Not the Math

- What part of the current manufacturing process would most benefit from improved scheduling?
  - Raw material receipts
  - Manufacturing process itself
  - Shipments and transportation
- What kinds of optimization results are measurable?
  - Minimize tardiness
  - Maximize throughput
  - Minimize makespan
  - Maximize yield
  - Minimize resource usage (e.g., labor, power, water, …)
- Where is the greatest potential payoff:
  - Improved predictability (e.g., on-time delivery)?
  - Improved responsiveness to disruptions?
  - Improved capacity utilization?
  - Reduced working capital?
  - Reduced raw material costs?

The customer's problem needs to be solved; scheduler implementation is irrelevant

Common Scheduling Approaches

- **Simulation:** Describe the system behavior in terms of how it changes over time. Crank the model forward to create a history and examine it.
- **Mathematical programming**: Encode the system behavior and the problem to be solved (tasks, resources, sequencing, objective function, etc.) in a mathematical model, and solve. Variable assignments in the solution specify system behavior.
- **Constraint-based scheduling**: Represent system behavior and problem specification in terms of a set of constraints. Add additional constraints to enforce scheduling “decisions” (e.g., activity start/end times, task ordering, or resource assignment).

Mathematical Programming

E.g., mixed integer linear programming:

- Strengths
  - Theoretical basis
  - Well-understood optimization algorithms
  - Performance guarantees
- Weaknesses
  - Modeling restrictions
  - Model explosion (20 recipes -- 30,000 variables)
  - Manual intervention is cumbersome
  - Explicit objective functions are required
  - Rescheduling is either inflexible, or too flexible

Constraint Envelope Scheduling

Constraint Satisfaction Problems (CSP) are specified as:

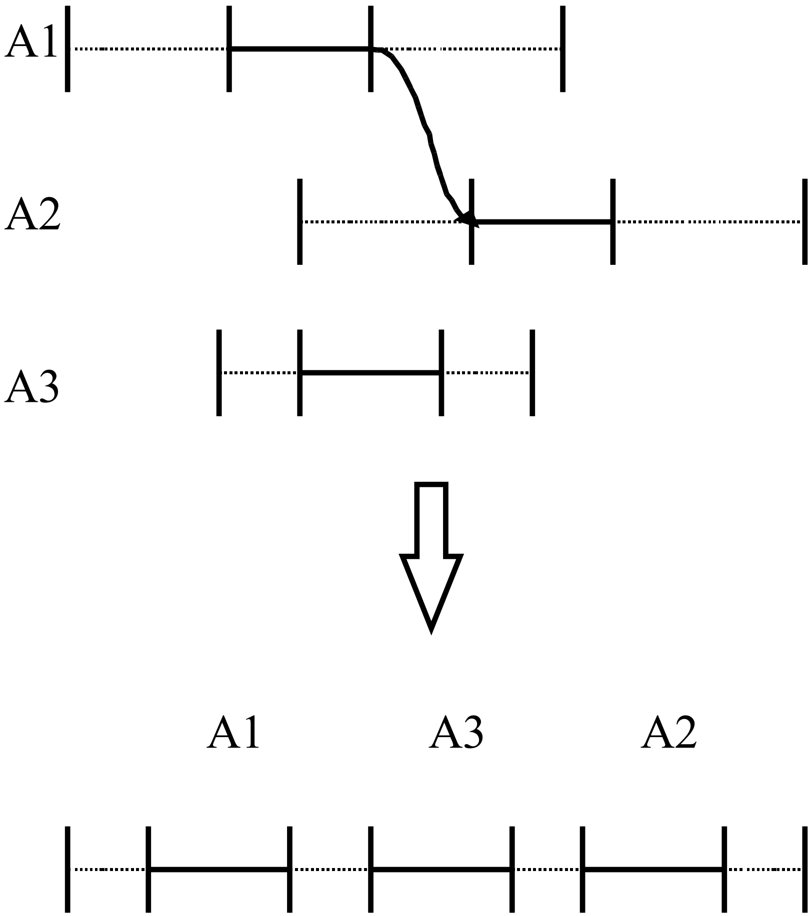
- A set *V* of *variables*
- A set *C* of *constraints,* each constraint a relation specifying tuples of permissible values for some subset of *V.*

The objective is to find a feasible complete assignment for *V,* consistent with *C.*

Scheduling variables: Activity start and end times and resource assignments

Scheduling constraints: Temporal, resource, and system dynamics

Constraint Envelope Scheduling: Accumulate additional constraints (e.g., activity orderings), so as to narrow the space of possible schedules, until all schedules in the remaining set are feasible.



Domain Examples

Aircraft and Avionics

Avionics Scheduling

Objective: Schedule processing and communications

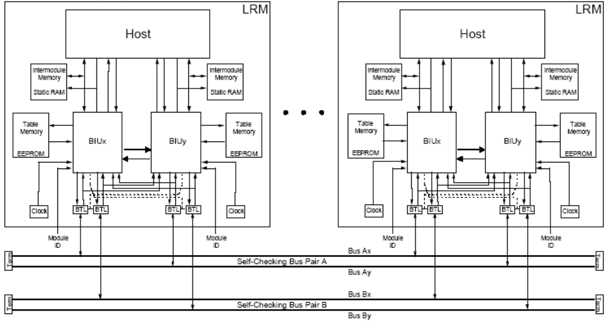
Simple

- Deterministic, hardware driven, cyclic scheduling on high integrity, high availability backplane
- Schedule software processes on processors and data transfers on shared backplane to satisfy rate, latency, and jitter requirements

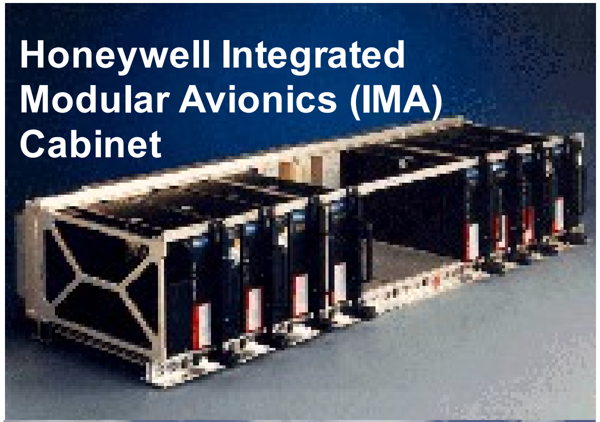
More Complex

- Precise definitions of latency and jitter
- Non-zero partition interrupts and process context switches
- Hard real-time system that supports inter-process communications with tight latencies
- Fault-tolerant applications and data communications
- Asynchronous inter-cabinet and external input/output communications
- Infrastructure synchronization communications
- 10s of diverse software applications hosted in cabinet
  - High rate, tight latency requirements, fairly low processing overhead
  - Low rate, medium latency requirements, significant processing overhead
- Growth and incremental change
- Scale: 30k objects subject to 100k metric constraints
- Utilization: 98% scheduled processors

Avionics (SAFEbus(R)): Honeywell International, Brendan Hall et al, "Ringing Out Fault Tolerance." (DSN'05) pp 298-307.



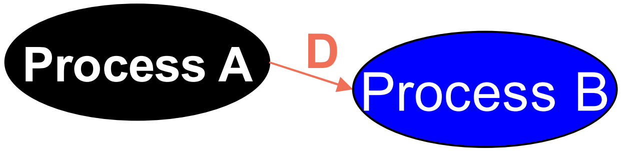
SAFEbus Hardware



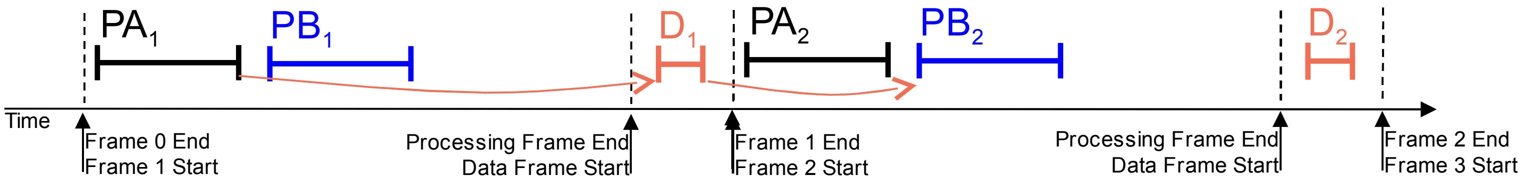
http://www.boeing.com/commercial/777family/gallery/k63667_05.html

Frame Based Communications

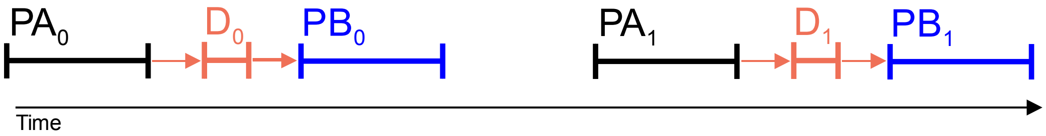
Consider simple case where Process A sends Data D to Process B: Move(A(D),B):



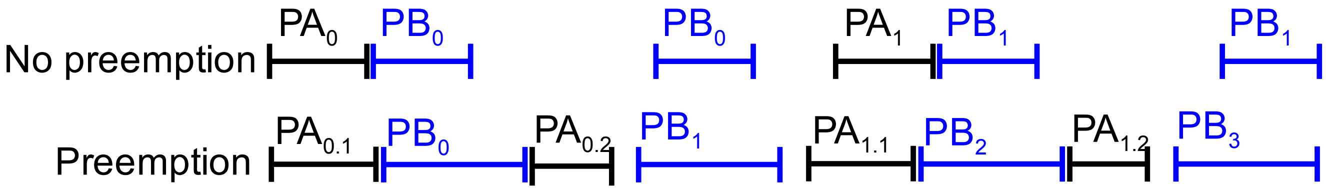
Frame-Based Communications – can add a period of latency to each data transfer, so $PA_1 \rightarrow PB_2$ is the effective transfer



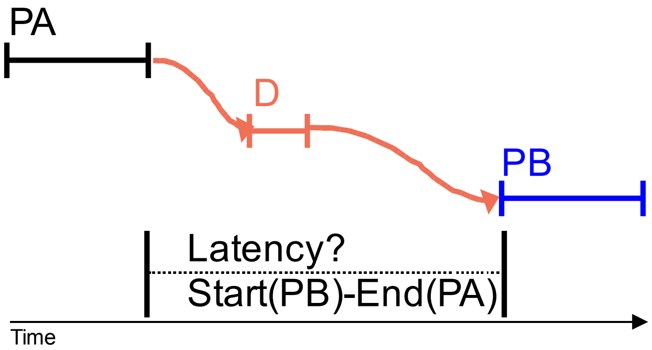
Intra-Frame Communications – reduces latency, but now need buffer consistency: so End(PA)<Start(D), End(D)<Start(PB), etc. $PA_1 \rightarrow PB_1$ is the effective transfer



It starts to get tricky when PA and PB do not run at the same rate:

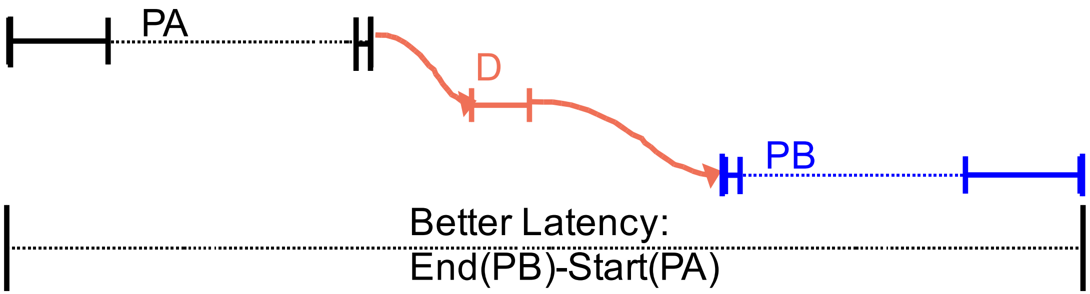


Real World Avionics: Latency Example



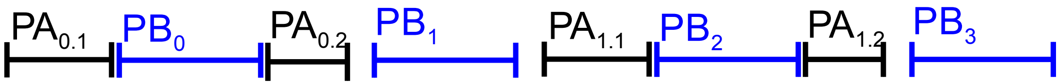
This might work in the presence of other assumptions, e.g., non-preemption, rate matching, or acceptable latencies greater than frame-times.

What happens with preemption? Is the above latency metric meaningful?



Real World Avionics: Context Switches

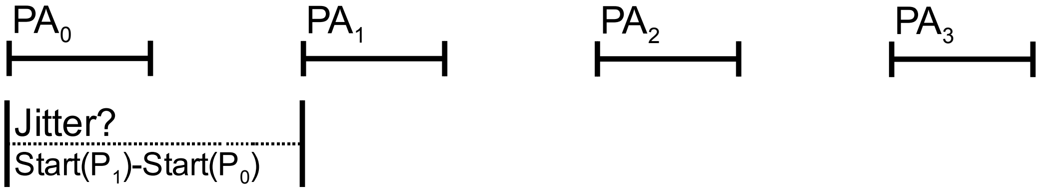
- Simple Processor Utilization: $\sum_{process(i=0)}^{all\_processes} ExecutionTime(i)$
- Consider preemption:
  - PB interrupting execution of PA
  - PA could be low rate, long execution time
  - PB could be high rate process



$PA_{0.1}$ PB_0 $PA_{0.2}$ PB_1 $PA_{1.1}$ PB_2 $PA_{1.2}$ PB_3

- Above processor utilization prediction is unrealistic, given nonzero, significant, and different “Start” and “Resume” context switches:
  - Inter-process switching overhead (processes within same partition may interrupt each other)
  - Inter-partition switching overhead (where a partition is a unit of protection)
    - Application performance can be affected if caches are flushed (rather than buffered and restored) between partition context switches. Therefore, execution time allocation might need to increase.
    - Which operating budget “pays” for this overhead? The interrupted partition, interrupting partition, or the runtime? The application teams might be from different companies

Real World Avionics: Jitter

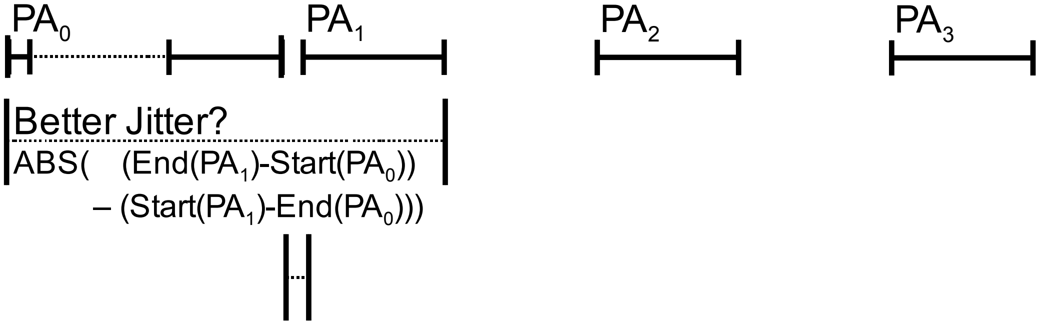


PA_0 PA_1 PA_2 PA_3

Jitter?

Start(P_1)-Start(P_0)

Preemption might void semantic intent:



PA_0 PA_1 PA_2 PA_3

Better Jitter?

ABS((End(PA_1)-Start(PA_0))
– (Start(PA_1)-End(PA_0)))

Real World Avionics: Incremental Change

Testing to level necessary for FAA certification is phenomenally expensive.

- If something needs to change to accommodate additional functionality, the impact of changes must be precisely *identified* so retesting can be performed
- *Identification of impact* is insufficient – it must be managed, so the critical partitions (most expensive to recertify) have minimal change
- Given this, then how do you accommodate future growth?
  - Do you schedule open slots at a high rate?
  - Or low rate, long processes?
  - Once you've done it, you're stuck, since changes imply recertification.

Real-World Avionics: Hot Spots

Failure of any sort is unacceptable.

- Meeting “90% of the constraints” does not make sense
- Proof of constraint satisfaction is required

Applications sell avionics. Scheduling is merely a support function. Therefore, if a schedule satisfying the constraints cannot be found, the application design teams typically require proof of unschedulability.

How do you do that when such a proof is, in the general case, computationally equivalent to the scheduling problem?

It was fairly easy when the processors were 130% loaded. After that it got tricky.

Airplane Crew Scheduling

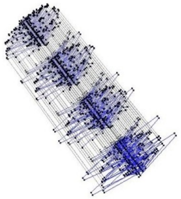
Objective: Match cockpit and cabin crews to airline routes

Simple

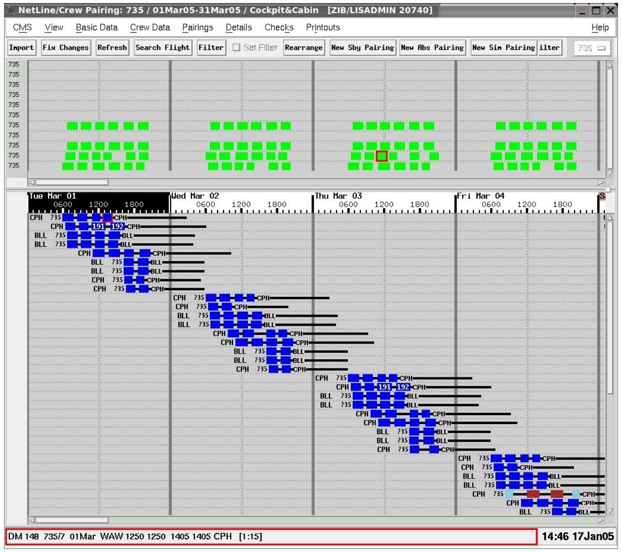
- Satisfy FAA operations constraints (flight crew aircraft qualifications, number of consecutive hours on duty, total hours per week, down time)
- Satisfy airline operations (crews for flights)
- Manage operations overhead
  - Irregular schedules
  - Large number of flights
  - Minimize flight delays for crew transfers
  - Minimize deadhead flights (relationship to the traveling salesman problem)
  - Maximize down time at home, rather than out

More Complex

- Crew "teams" for safety and working efficiency
- Disruptions
  - Weather
  - Maintenance
- Union seniority: (***HUGE Usability Impact***)
  - Preferred routes and destinations
  - Distributed, time varying input
  - Minimize downgrading (pilot flying as co-pilot)



http://www.zib.de/Optimization/Projects/Traffic/CS/CSlong.en.html



NetLine/Crew Pairing: 735 / 01Mar05-31Mar05 / Cockpit&Cabin [ZIB/LISADMIN 20740]

DM 148 735/7 01Mar WAW 1250 1250 1405 1405 CPH [1:15]

14:46 17Jan05

Stratospheric Observatory for Infrared Astronomy

Objective: Generate flight plans (takeoff, duration, tracks) which satisfy requested observations, designated flight days, weather predictions and aircraft constraints.

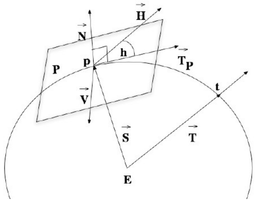
“Simple” (none of it is simple)

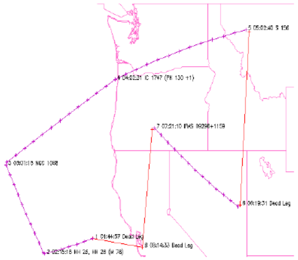
- Aircraft constraints (elevation, horizontal telescope motion, flight duration, altitude profile, takeoff / landing sites)*
- Astronomical constraints (object visibility, object tracking)
- Optimize observing runs (minimize dead legs)

More Complex

- Integration of planning and scheduling – observatory is oversubscribed, as they all are
- Airspace restrictions (traffic, population)
- Fuel consumption modeling
- Time varying weather predictions – including winds (affects aircraft performance) atmospheric water and temperature (affects observing quality and performance)
- Handle uncertainty in weather predictions and airspace restrictions

http://www.usra.edu/sofia/photos.shtml

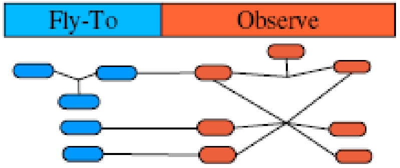




Solving for target elevation and ground track

http://www.stsci.edu/institute/conference/iwpss/poster-a-frank.pdf

http://is.arc.nasa.gov/AR/slides/images02/SOFIASchd02.pdf



*Flight Planning for Sofia, Jeremy Frank, Elif Kurklu, NASA Ames http://is.arc.nasa.gov/AR/slides/images02/SOFIASchd02.pdf

Space, Earth Observing and Science

Earth Observing System (EOS)

Objective: Schedule observing and data dissemination

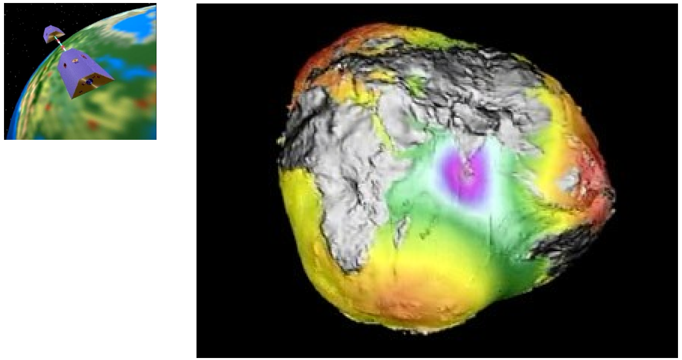
http://eospso.gsfc.nasa.gov/

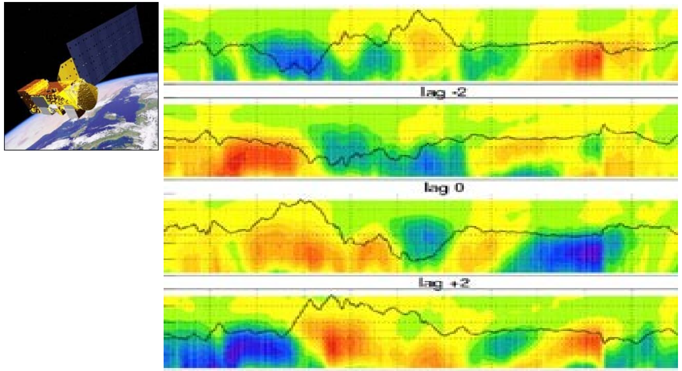
Simple

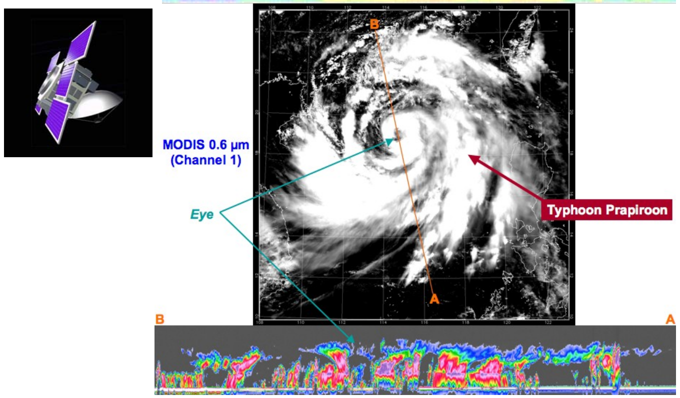
- Observing targets with different science values and ALWAYS more targets than observing time
- Orbital mechanics
- Coordination of simultaneous observations

More Complex

- Conservation of momentum and use of expendables
- Communications scheduling with ground stations and the Tracking and Data Relay Satellite System (TDRSS)
- Model drift (e.g., equipment wear due to extended mission lifetimes)
- EOSDIS Core System science data processing and data management







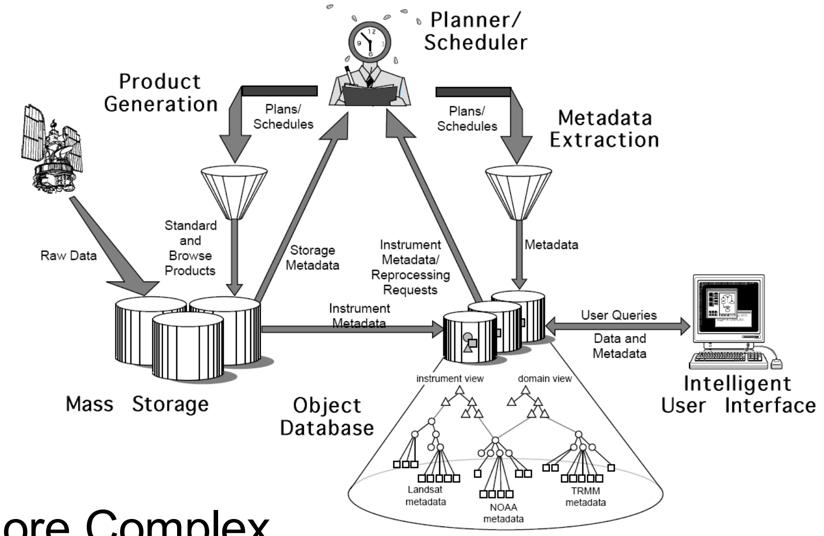
EOS Science Data

Objective: Efficient distribution, processing, and management of petabytes of data

- Petabytes of raw data stored on media with varying access bandwidths and lags available at Distributed Active Archive Centers (DAACs)
- The data is not necessarily “on-line.”

Simple - Distributed science teams need the data

- Shared objectives
- Uncoordinated schedules
- Activities:
  - Generation of content-based meta-data for indexing
  - Search
  - Distribution
  - Archival
  - Retrieval
  - Analysis
- Near real-time responsiveness
- Fault tolerance – intermediate results and rollback



More Complex

- Integration of planning, or deciding what activities to perform, compared to scheduling, which decides how and when those activities are performed
- Dealing with uncertainty:
  - Actual durations vary from predictions
  - Requests arrive unexpectedly
  - Resources will be unavailable or fail during use
- Dynamic, distributed scheduling
  - Each DAAC can receive requests
  - Request satisfaction might involve coordination between multiple DAACs, based on data, computing, or communications availability

Manufacturing and Process Industries

Manufacturing Abstraction

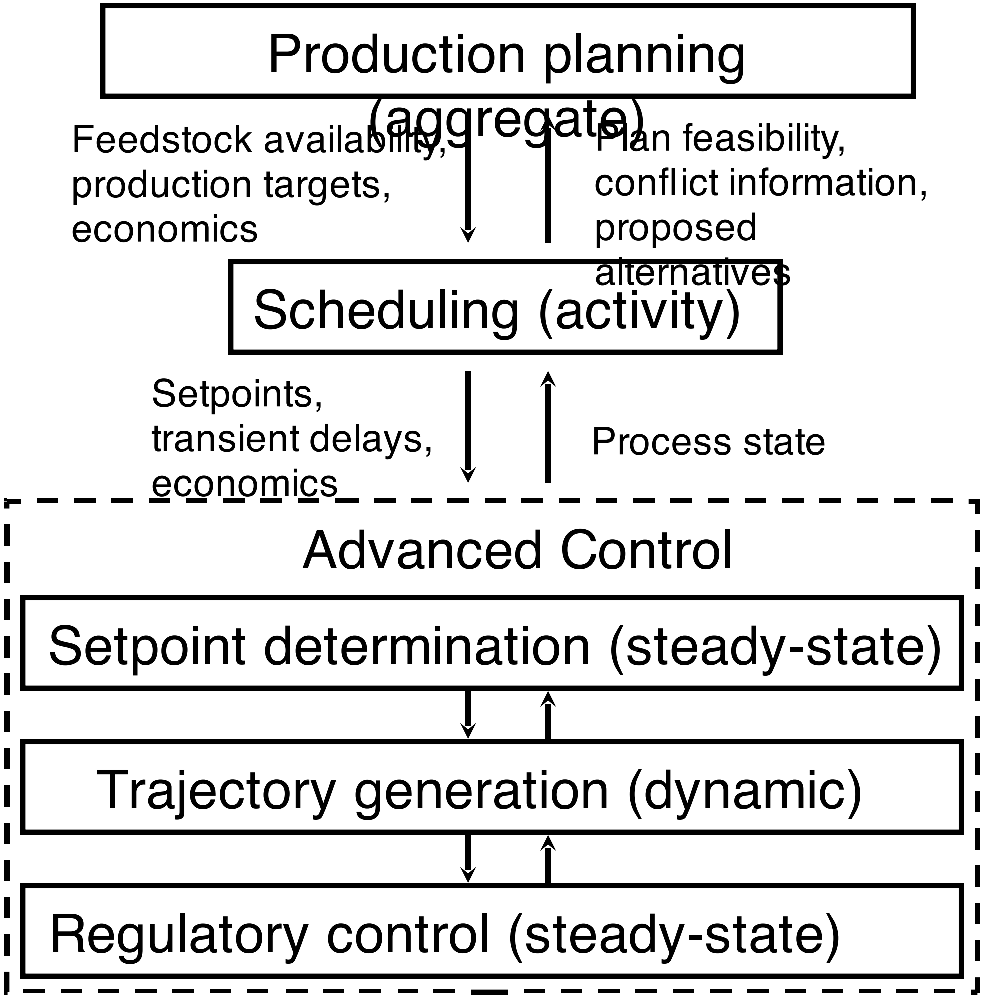
Multiple, asynchronous operations

Resource constraints drive everything else

Control over events is limited

Flexibility is important

System dynamics is (sometimes) crucial



Production planning (aggregate)

Feedstock availability, production targets, economics

Plan feasibility, conflict information, proposed alternatives

Scheduling (activity)

Setpoints, transient delays, economics

Process state

Advanced Control

Setpoint determination (steady-state)

Trajectory generation (dynamic)

Regulatory control (steady-state)

Interesting Manufacturing Quotes

“All we really want is to be able to model polymer production in units of less than a day.”

“If your system can’t trade weight for volume based on price, it’s useless to us.”

“Our main problem is hot lots. On one occasion, we shut down the entire plant to get a product out for a high priority customer.”

“What we really need is lot tracking.”

Example Manufacturing Scales

Semiconductors:

- 100s of different processes, dozens of steps
- Conditional and “reentrant” processes

Batch:

- 10s of products, multiple possible processes
- Different equipment
- Different parameter settings

Pulp and Paper:

- Multi-step processes, batch sizing/partitioning

Petroleum:

- Flexible plants, multiple product paths
- Complex intra- and inter-process dependencies

Batch

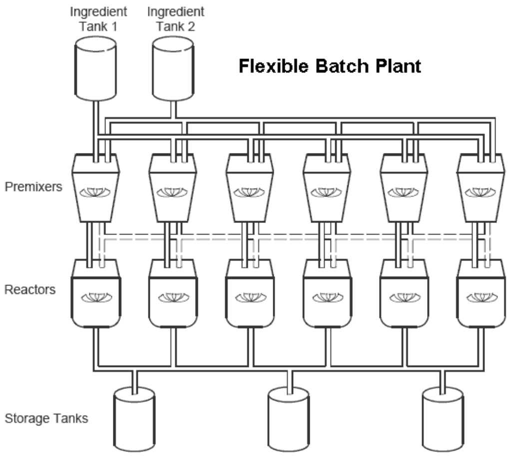
Objective: Schedule units and ingredients to product products

Batch Process: discontinuous process involving the bulk movement of material through sequential manufacturing steps.

- Each step can be abstracted to forming material when a measured quantity of reactants is added to a reaction vessel, the reaction is carried out, and the products, with a goal of identical properties throughout, are removed from the reaction vessel.
- Batch manufacturing is somewhere between continuous and discrete manufacturing

Examples: candy, cookie dough, beer, plastics, metal smelting

Key components: material handling and reactive processes, and plant topology



Batch Recipe

Simple

- Ingredient materials and reactions
- General, site, and unit level descriptions
- Physical resource allocation

More Complex

- Multiple recipes to satisfy product property requirements
- Batch sizing
- Changeover constraints (e.g., chocolate follows vanilla, not the other way around)
- Resource capacities
  - Power
  - Heating or cooling
  - Labor
- Schedule Upsets
  - Unexpected events
  - Process step duration variability
  - Business issues
  - Manual adjustments

Example Recipe Hierarchy:

General Recipe

- Amounts: ingredient 1: 500 gal; ingredient 2: 1000 gal; catalyst: 10 lb; produces 1200 gal
- Mix ingredients 1 and 2
- Add catalyst
- Heat for 2 hr, agitating

Site Recipe

- Mix ingredients 1 and 2 in Premixer to make slurry
- Transfer slurry to Reactor
- Add catalyst to Reactor
- Heat Reactor while agitating

Unit Recipe

- Charge from Premixer
- Charge from catalyst conveyor
- Start agitator
- Heat to 500F
- Hold at 500F for 2 hr
- Stop agitator
- Dump

Refineries & Process Industries

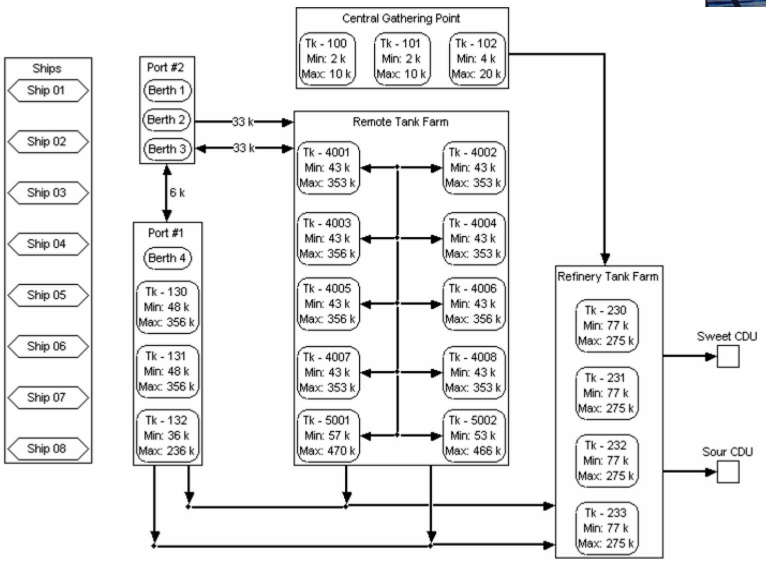
Objective: Schedule movements, blends, and processing of stock to create products

Both continuous and batch processes

Example Unit Resources:

- Movements
  - Shipments
  - Rundowns
  - Blends
- Tanks
- Paths
- Ships
- Properties
- Pipelines
- Pumps, Berths, Transfer lines
- Reactors

Movements



Simple

- Source tank
- Destination tank
- Volume
- Rate bounds

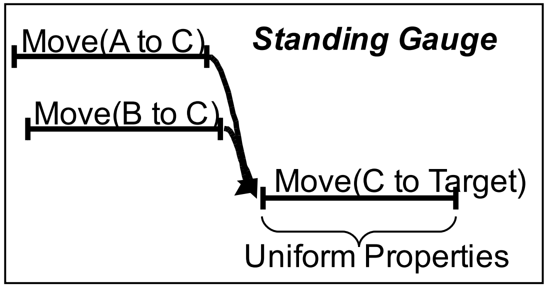
More Complex

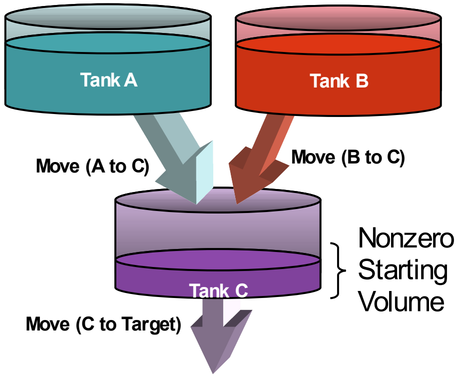
- Multiple *concurrent* sources and destinations
- Tank swings specifying volume to be moved
- Changeover can have cost; akin to “context switch”
- Blend recipes
  - Tree of multiple sources
  - Source may be a tank, a transfer line, a shipment, or some combination
  - More than one recipe might satisfy a product specification (e.g., there is more than one way to make an 87 octane gasoline blend)

Tanks

Simple

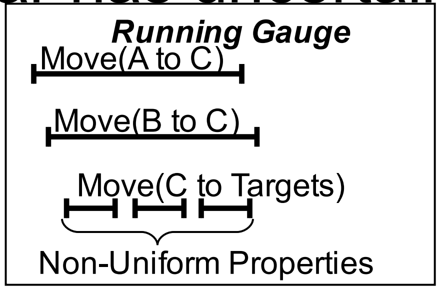
- Consumable resource with min/max
- Mass conservation
- Standing Gauge





More Complex

- Lower bound in particular has uncertainty (water)
- Running Gauge
- Contents are mixtures
  - Mixing takes time
  - Can be constraints on allowable pre/post mixtures
  - Volume might not be conserved
  - Tank might be a ship that has limited window of availability



Pumps

Simple

- Rate
- Mass conservation

More Complex

- Type constraints (some pumps might not be able to pump all materials)
- Rate dependence on
  - Viscosity (which can be related to environmental temperature)
  - Length of pipe
- Volume might not be conserved

Pipes

Simple

- Transfer Delay
- Point-to-point
- Headers (including mixing)

More Complex

- Pipes have volume
- Pipes are networks (can be modeled as point-to-point with headers at nodes)
- Material doesn’t come out the end unless material is being pumped into the other end, and
- You can’t pump in one end unless material is coming out the other end (or an intermediate point)
- Mixing occurs at material interfaces
  - Certain materials are constrained from being adjacent
  - In other cases, order is constrained
  - Some constraints are strong desires (undesirable mixtures are dumped into a tank and eventually sent back for re-refining, or sold at a loss)

Complex Assets

- Crude oil is a mixture of chemicals, and assayed characteristics can have wide error bounds
- Pipes, tanks, blenders, etc., are “simple” storage, movement, and combining resources
- Other units, possibly operated as a continuous process, manipulate the component properties:
  - Distillation
  - Cracking
  - Recombination
  - Coking
  - Sulfur extraction
  - Metal extraction
- The resource model can be time varying
  - Catalyst recovery
  - Unit maintenance
  - Seasonal swing

Need for Agility

Examples of events that can require rescheduling:

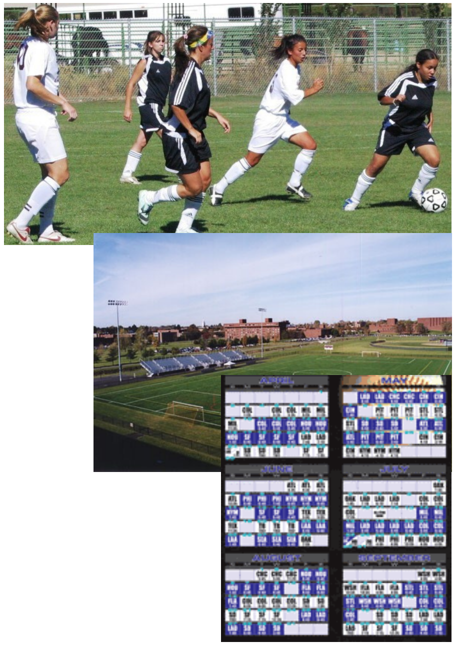
- Cancel a single pipeline shipment (24 hours notice)
- Triple a pipeline shipment (96 hours notice)
- Shift all shipments on one pipeline later by 24 hours (no notice)
- Increase one shipment, cancel the next.
- Tank swings out of a segregation (24 hours notice)
- Blender down for six hours, no notice
- Reduced rundown for blendstock across entire period

Refinery Optimization

- Many decisions to make
- Many options are available
- Many schedules provide realizable solutions (e.g., hard constraints are satisfied)
- United States refineries operate in the 90%+ utilization – profits seem to be accumulated when high 90s are achieved
- Revenue (not profit) can be measured in $M/hr throughput. Implications of down time are significant
- Optimal behavior may be at an enterprise level, rather than refinery or unit level; implication is seemingly arbitrary imposed constraints

Other Interesting Domains

Sports League Scheduling & Rostering



Objective: schedule sports teams to play each other.

Simple

- Satisfy all necessary combinations
- Satisfy home / away number of games
- Minimum number of rounds with fewest byes

More Complex

- Consecutive home or away games
- Consecutive “strong” opponents
- Field and stadium conflicts
- Minimize travel distances, travel overhead

Summary and Conclusions

Additional Issues

- Scheduling and probability (e.g., predicted orders, growth, arrivals)
- Risk vs robustness
- Planning
- Supply chain management
- Optimization
- Distributed vs localized scheduling
- Closed-loop scheduling

To Read More

- Boddington, C.E., Planning, Scheduling, and Control Integration in the Process Industries, McGraw-Hill, 1995.
- Boddy, M., White, J., Goldman, R., and Short, N.: Integrated Planning and Scheduling for Earth Science Data Processing, Hostetter, Carl F., (Ed.), Proceedings of the 1995 Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies, NASA Conference Publication 3296, 1995, 91-101
- T. G. Fisher, Batch Control Systems: Design, Application, and Implementation Instrument Society of America, 1990.
- Goldman, R., and Boddy, M.: Constraint-Based Scheduling for Batch Manufacturing, IEEE Expert, 1997
- Martin Henz, "Scheduling a major college basketball conference, Operations Research, 2000.
- L. Khatib, J. Frank, D. Smith, R. Morris, and J. Dungan, “Interleaved Observation Execution and Rescheduling on Earth Observing Systems,” in Proceedings of the ICAPS-03 Workshop on Plan Execution, 2003.
- E. Kirklu and Jeremy Frank, “SOFIA‘s Choice: Scheduling Observations for an Airborne Observatory,” in Proceedings ot the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS) 2003.
- Sherwood, R., A. Mishkin, S. Chien, T. Estlin, P. Backes, B. Cooper, G. Rabideau, and B. Engelhardt, “An Integrated Planning and Scheduling Prototype for Automated Mars Rover Command Generation,” in Proceedings of the 2001 European Conference on Planning (ECP2001), 2001.