Constraint-Based Planning and Scheduling Models

Stephen F. Smith

The Robotics Institute
Carnegie Mellon University

CAPD EWO Seminar – November 27, 2012

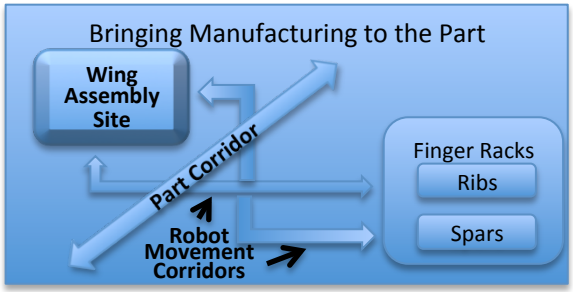
Intelligent Coordination and Logistics Laboratory

The Robotics Institute, Carnegie Mellon University
Stephen F. Smith, Research Professor and Director

Focus: *Scalable, execution-driven technologies for planning, scheduling and coordination*

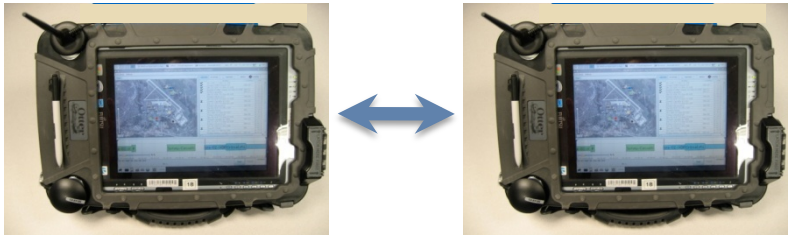
– Adaptive traffic signal control [Traffic21, Heinz]

– Dynamic, Real-Time Routing of Paratransit Vehicles [ATWIC]



– Multi-Robot Coordination for Aircraft Assembly [Boeing]

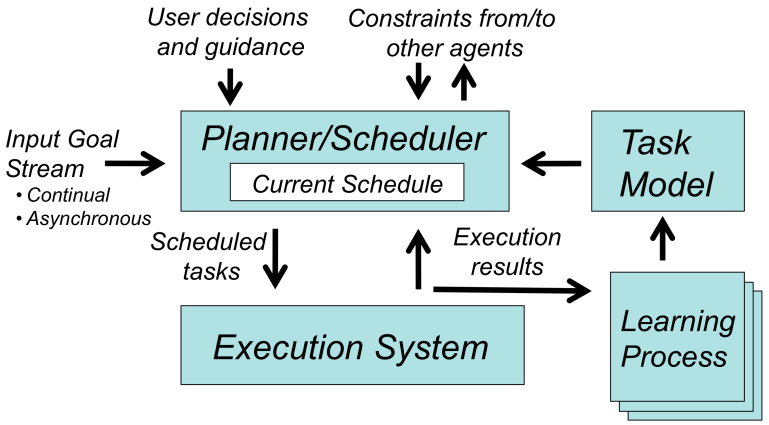
– Mixed-Initiative Transportation Planning [AFRL - USTRANSCOM]



– Distributed management of joint plans for disaster response [DARPA]

Execution-Driven Planning and Scheduling Technologies

- Organized around principle of "optimization in context"
  - A plan/schedule is always in place and executing
  - Continual integration of new demands, new information, new constraints, new results (coming from users, other agents, execution system)
  - Continual refinement of task/domain model
- Some key attributes:
  - Decision-making in pace with execution
  - Plans that account for and hedge against uncertainty
  - Incremental change and solution stability

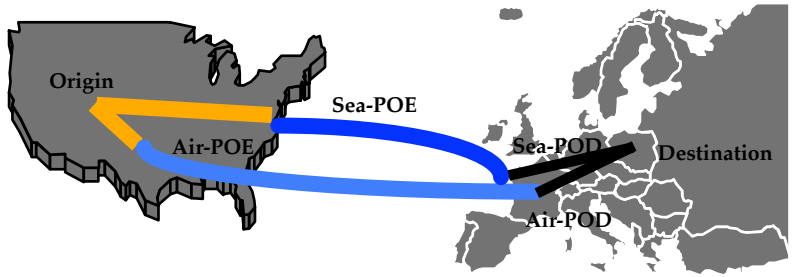


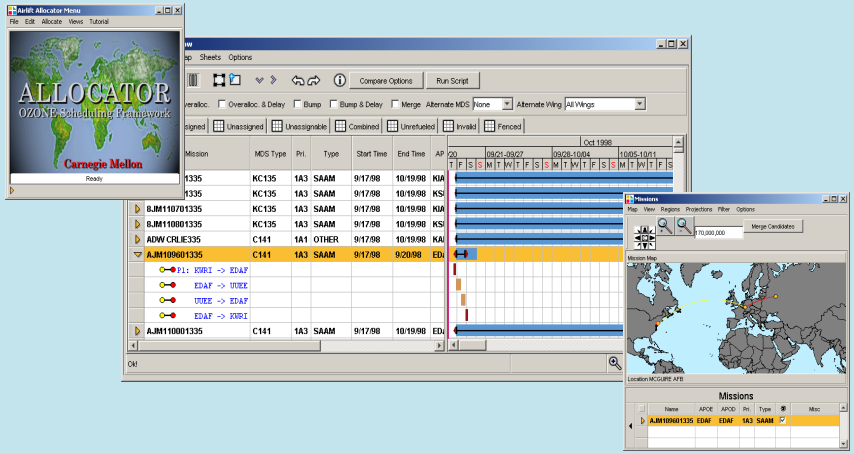
Carnegie Mellon

THE ROBOTICS INSTITUTE

The AMC Allocator:

Advanced Scheduling for the USAF Air Mobility Command





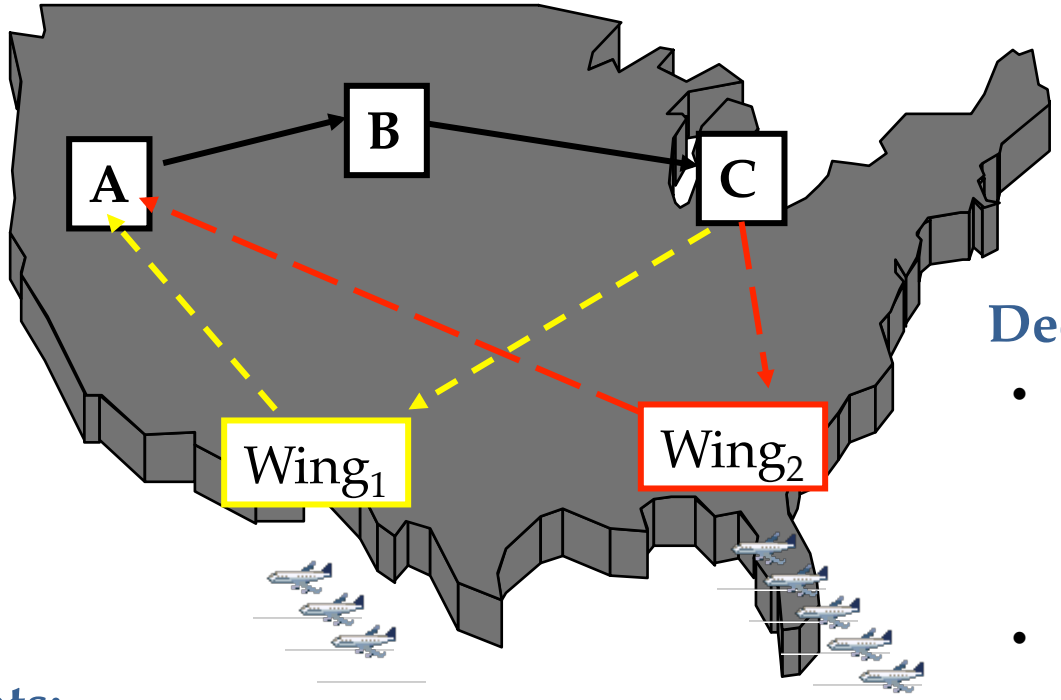
- **Problem:** Day-to-day allocation of aircraft & crews to airlift, tanker missions
- **Characteristics**
  - Large scale: 1,000s of missions; 100s of assets
  - Continuous, dynamic stream of mission requirements

- **Core Technology:** Incremental, constraint-based search
  - integration of new requirements into the schedule to minimize deadhead travel time
  - Controlled reallocation of existing missions to maximize resource usage
  - Extended iterative search to further optimize if time permits
- **Status:** Embedded in AMC's operational planning system

AMC Airlift Allocation Problem

Requests:

- **Mission$_1$**: pick up cargo at A, deliver to B, then C.
- **Mission$_2$**
  ...
- **Mission$_n$**



Decisions:

- Assign resources (aircraft, aircrews) from Wing$_1$ or Wing$_2$?
- Start at what time?

Constraints:

- Aircraft, aircrew capacity at different locations
- positioning/de-positioning time
- crew duty day/crew rest
- cargo pickup and drop off windows
- mission priority
- port throughout

Objective:

- Maximize number of missions flown(oversubscribed form)
- Minimize total late days

THE ROBOTICS INSTITUTE

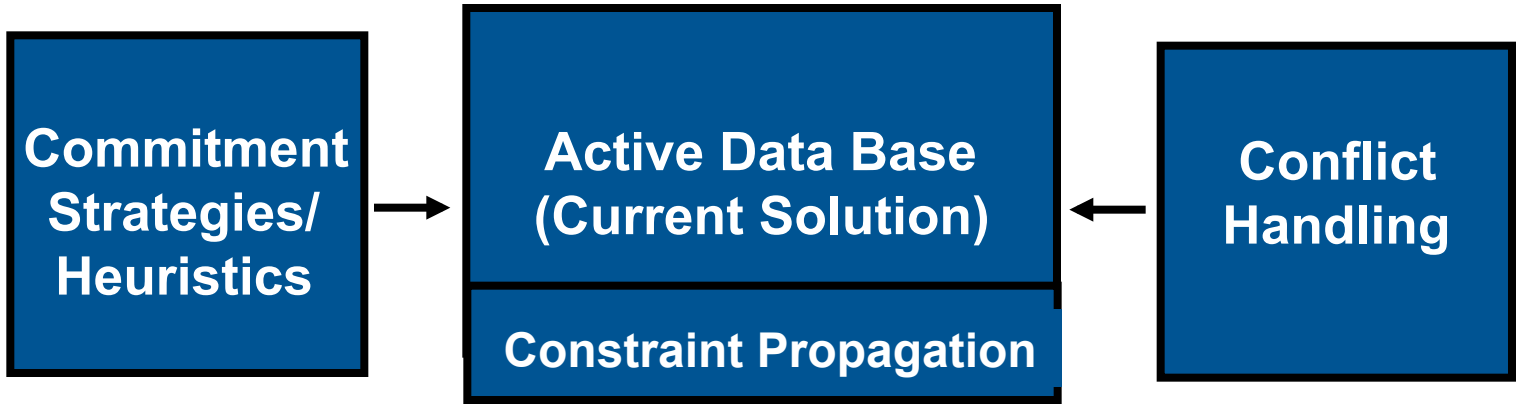
Carnegie Mellon

Outline of Talk

- Incremental Constraint-based Solution Generators
- Adding Optimizing Search Procedures
  - Heuristic-Biased Stochastic Sampling
  - Task Swap
- Generating Resilient Schedules
  - Hedging Against Temporal Uncertainty
  - Probabilistic Analysis of Deterministic Schedules
- Current Focus

Constraint-Based Search Models

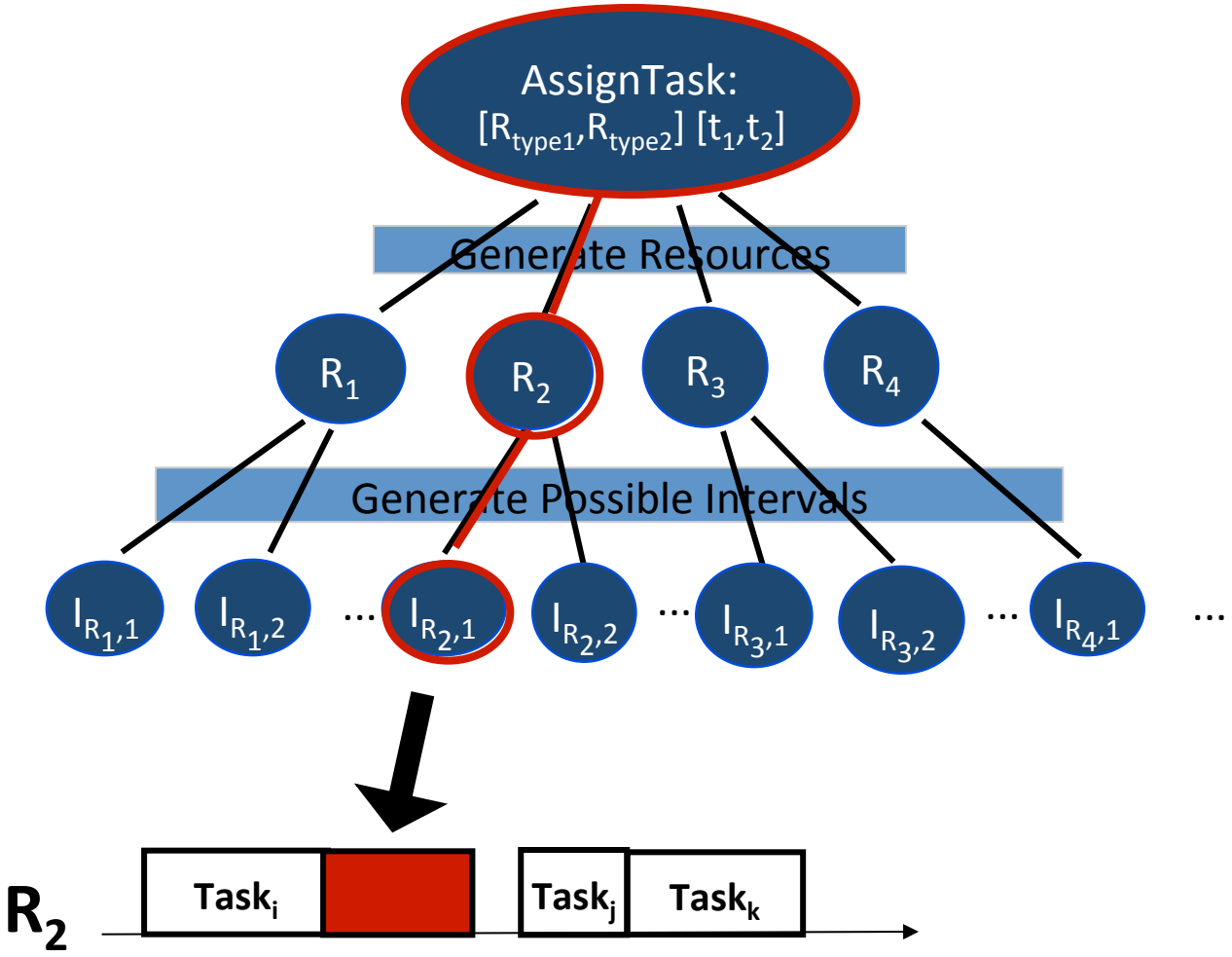
Components:



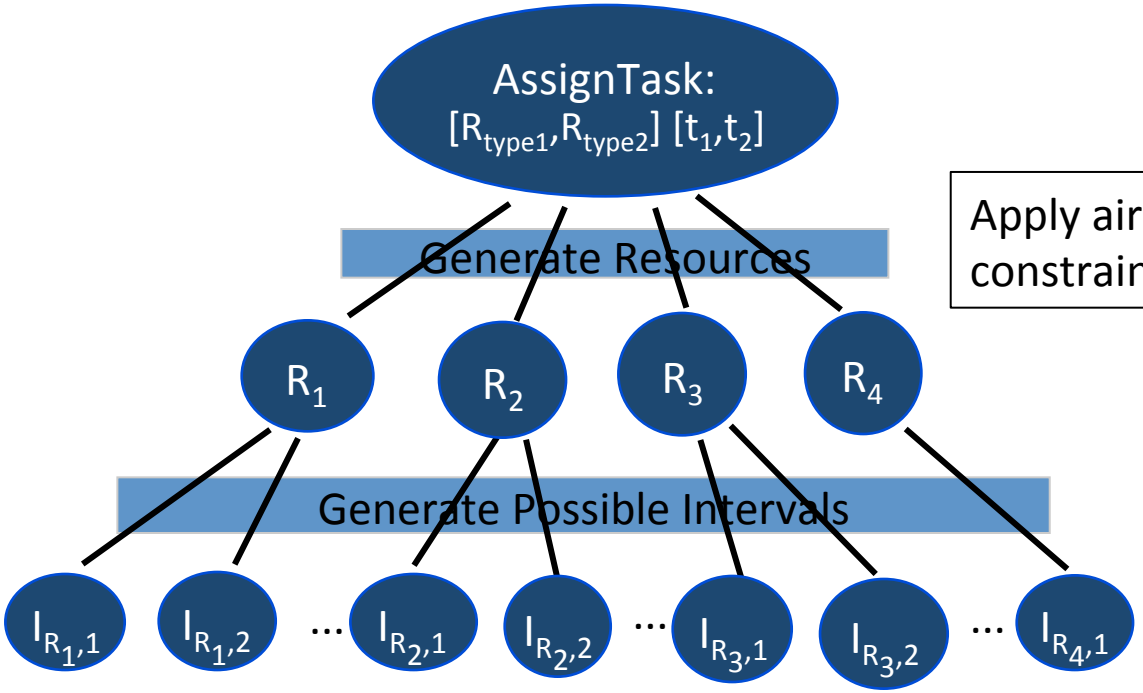
Properties:

- Modeling Generality/Expressiveness
- Incremental
- Compositional

Scheduling a New Task



Scheduling a New Task



Apply aircraft/cargo compatibility constraints to filter possible air wings

Apply aircraft, crew capacity constraints, mission time bounds, mission priority, port capacity and crew duty day constraints to filter possible intervals

Evaluate options and select best:

$O_{best} = \underset{j}{\text{argmin}}(w1*PDDist(O_j) + w2* st(O_j))$,

where

- PDDist is total pos/depos distance and
- st is the start time

THE ROBOTICS INSTITUTE

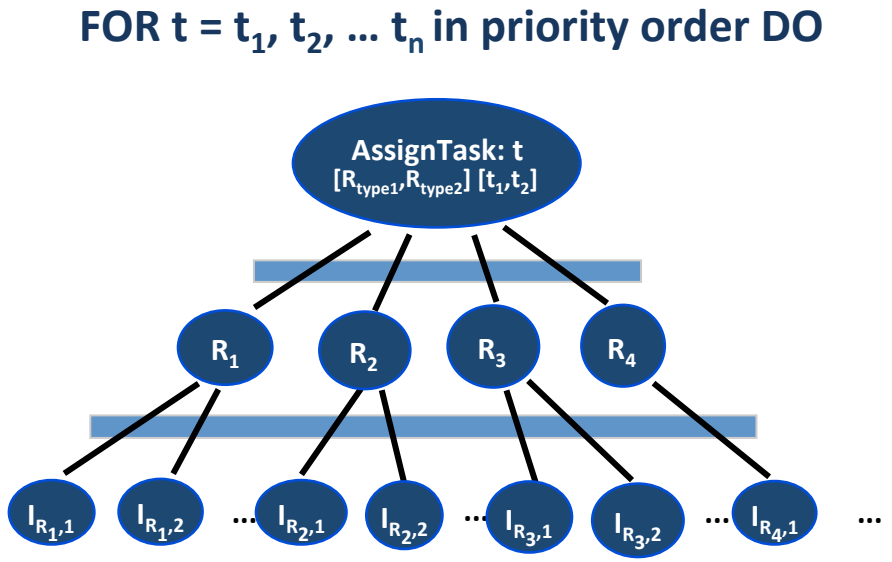
Carnegie Mellon

Option Generation

- Core search procedure is *configurable* to support generation of options when all constraints cannot be met
  - Treat time windows as relaxable and minimize delay
  - Relax capacity constraints and minimize number of extra tails/crews needed
  - Bump one or more lower-priority tasks to insert new task and minimize disruption

Configuring A Basic Scheduling Search Model

- Couple core search procedure with a *prioritization heuristic* to produce greedy solution generator
- Embed in larger optimizing search by



1. Dynamically changing the randomizing the heuristic and iteratively re-invoking the core procedure
2. Using the core procedure to seed a local search

THE ROBOTICS INSTITUTE

Carnegie Mellon

Outline of Talk

- Incremental Constraint-based Solution Generators
- Adding Optimizing Search Procedures
  - Heuristic-Biased Stochastic Sampling
  - Task Swap
- Generating Resilient Schedules
  - Hedging Against Temporal Uncertainty
  - Probabilistic Analysis of Deterministic Schedules
- Current Focus

Amplifying the Performance of Heuristics via Stochastic Search

Starting assumption: We have a good search heuristic, but its discriminatory power varies from context to context

Issue: How to balance adherence to heuristic against possibility of missing better solutions

Basic Approach:

- Randomize the heuristic to enable search in the neighborhood of the heuristic's trajectory
- Calibrate the degree of randomness to the level of uncertainty in a given decision context

Heuristic-Biased Stochastic Sampling

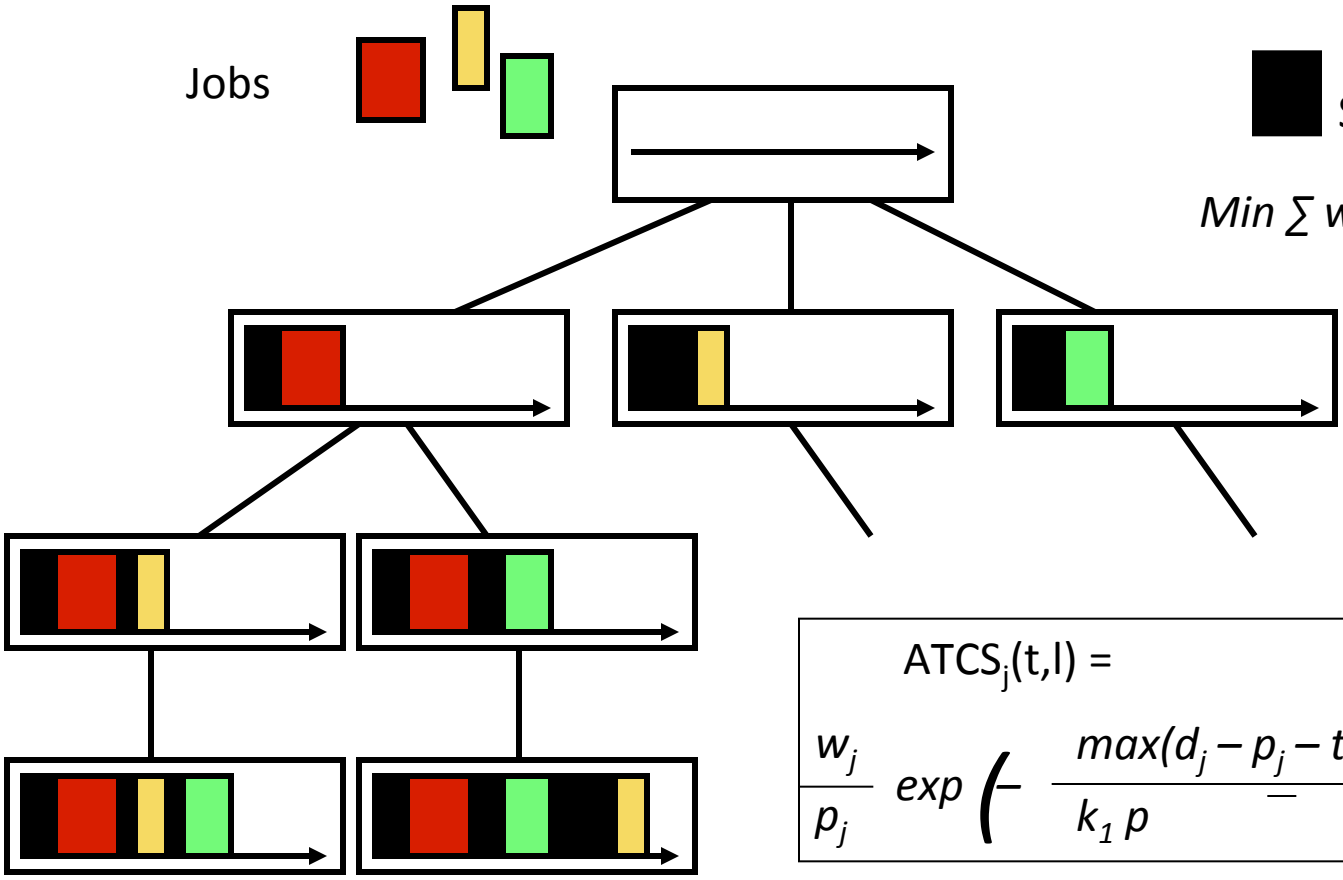
- At each decision point, order the possible choices according to a search heuristic.
- Choose branch of search space randomly but biased according to a function of this ordering.
- Rank-based bias (Bresina 1996): Use rank order and choose branch b_i with probability:

$$\frac{bias(rank(b_i))}{\sum_j bias(rank(b_j))}$$

- Value-based bias (Cicirello & Smith 2004): Use heuristic value assigned and choose branch b_i with probability:

$$\frac{bias(value(b_i))}{\sum_j bias(value(b_j))}$$

Application to Single Machine Weighted Tardiness Problem: The Search Space



$$ATCS_j(t,l) = \frac{w_j}{p_j} \exp\left(-\frac{\max(d_j - p_j - t, 0)}{k_1 \bar{p}} - \frac{s_{lj}}{k_2 \bar{s}}\right)$$

Percentage Improvement over Deterministic ATCS Rule

	Value	Rank	Value	Rank	Value	Rank
# Restarts	1	1	10	10	100	100
Loose due-dates	**20.29**	14.86	**45.14**	38.98	**55.35**	52.38
Medium due-dates	**2.13**	1.47	**8.38**	6.40	**13.73**	10.73
Tight due-dates	0.04	**0.21**	**0.91**	0.88	1.71	**1.83**
Severe setups	**8.12**	4.34	**20.94**	17.21	**27.03**	24.37
Moderate setups	**6.86**	6.69	**15.35**	13.63	**20.16**	18.93

- Significant cost performance advantage also shown wrt *Limited Discrepancy Search*, a systematic procedure designed to exploit a good heuristic

Task Swapping

[Kramer and Smith 2003,2004,2005]

Controlled schedule revision to accommodate additional tasks without relaxing constraints

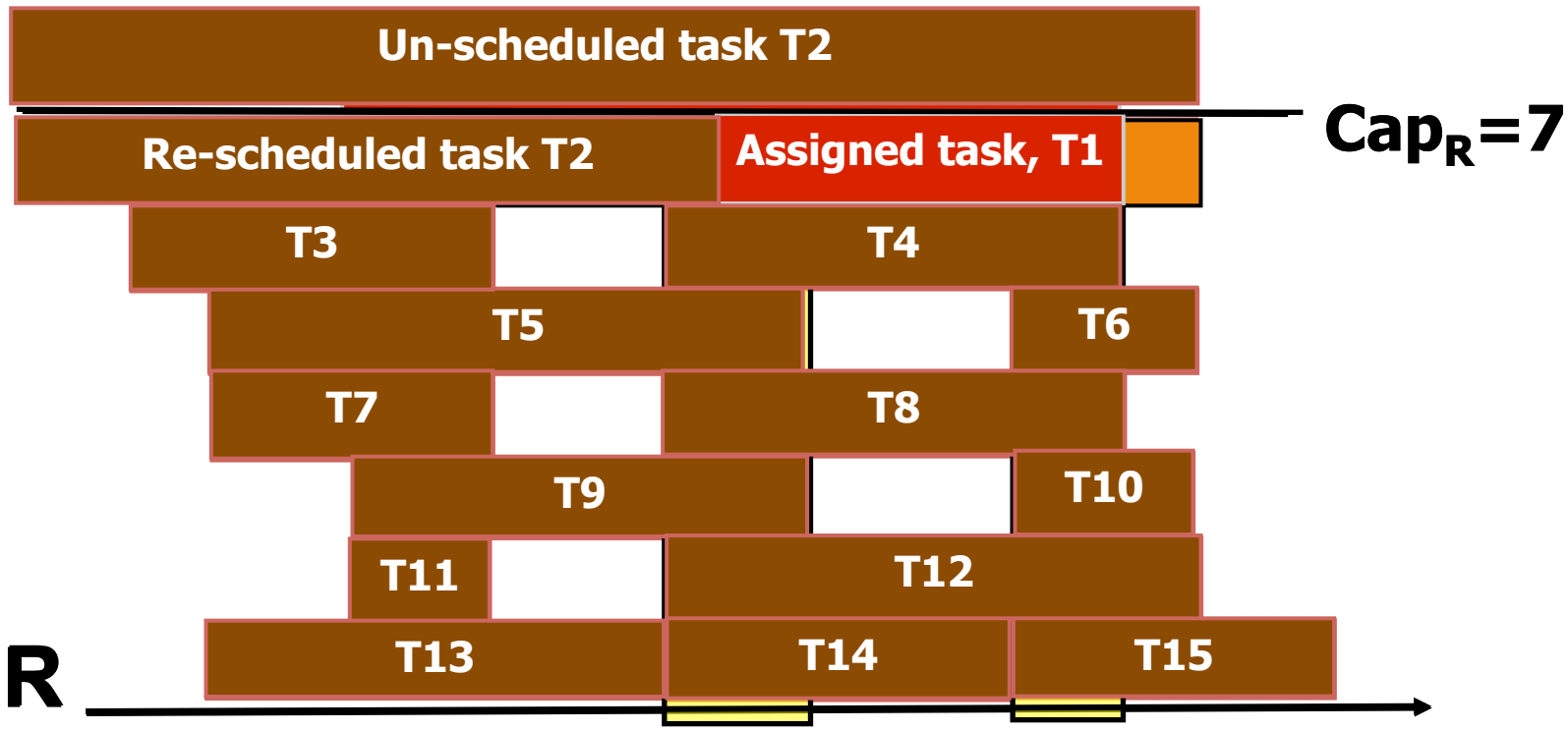
Motivation:

- Incremental, priority-based scheduling can result in fragmented resource usage, but
- Re-allocation is disruptive and should be minimized

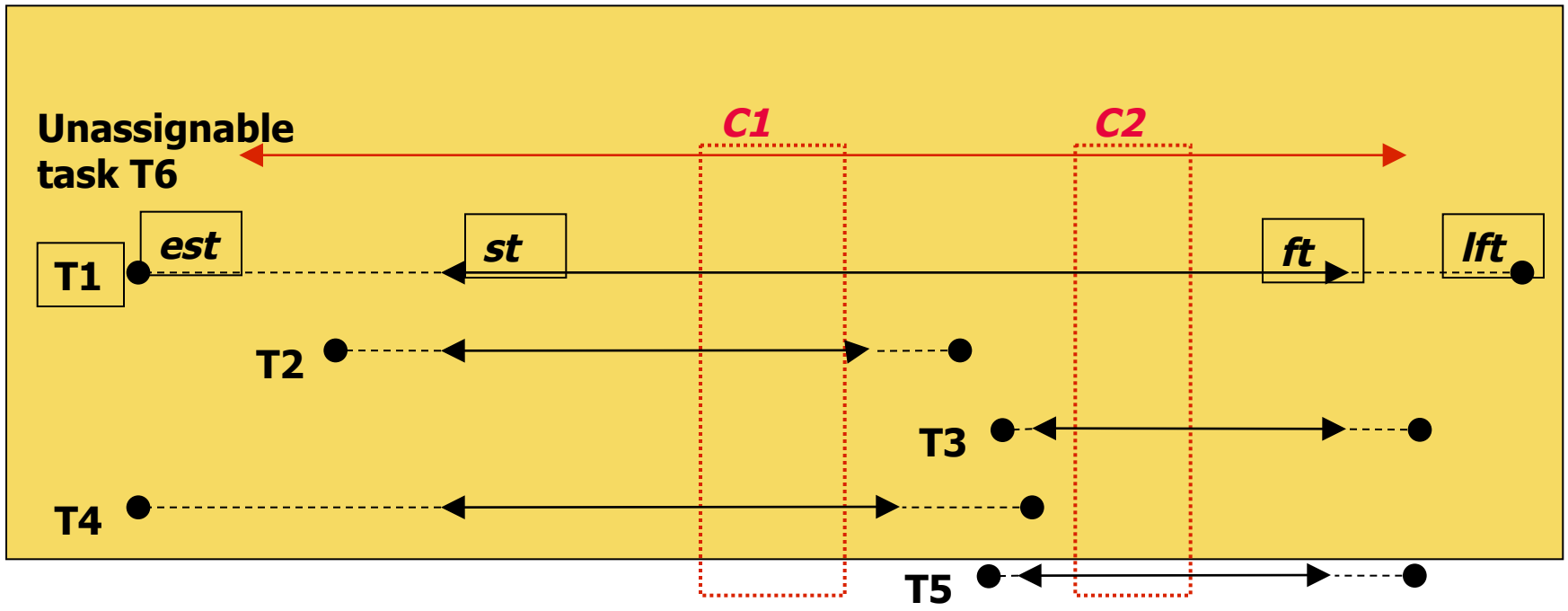
Basic Approach

- Temporarily relax priority constraint
- Conduct repair-based search around the "footprint" of an unassignable task *t*'s feasible execution window
- If all tasks displaced to accommodate *t* cannot be feasibly reinserted, Undo

An Example


Un-scheduled task T2
Re-scheduled task T2
Assigned task, T1
Cap$_R$=7
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
T15
R

Retraction Heuristics



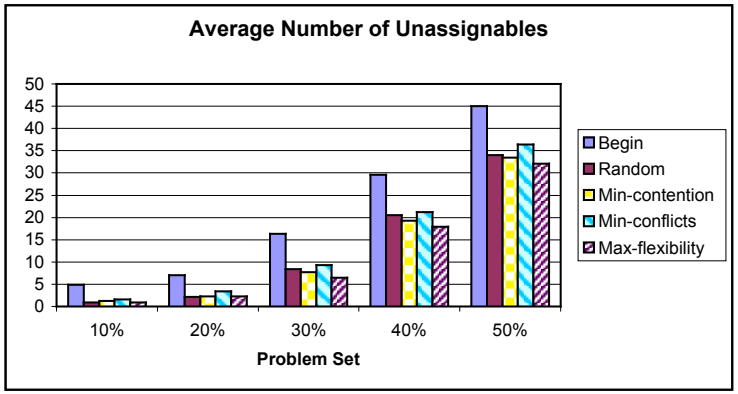
- Max-Flexibility = task-duration/feasible-window-size
- Min-Conflicts = count intervals that are at-capacity which conflict with a task's feasible window.
- Min-Contention = conflict-count * conflict duration

General Task Swap Procedure

- Given an *unassignable* task *t* to insert into the schedule,
  1. Identify where conflicts with that task exist.
  2. Retract 1 or more tasks in the conflicted areas to free up capacity
  3. Schedule task *t*, and mark it as “seen.”
  4. Re-schedule the retracted tasks.
  5. If all retracted tasks cannot be re-scheduled, recurse on them, most constrained first.
  6. If all tasks have been tested (seen) and some remain unassignable, backtrack to original state.

Application to AMC Domain

- **Original Results:** Real-world data sets from AMC Airlift mission scheduling domain (982 missions, 3251 tasks, 12 AirWings)
  - overall 42% of initially unassignable missions successfully integrated



- **More Recent Work:**
  - order of magnitude speedup resulting from incorporation of better pruning techniques
  - additional 20% improvement via use of iterated stochastic search

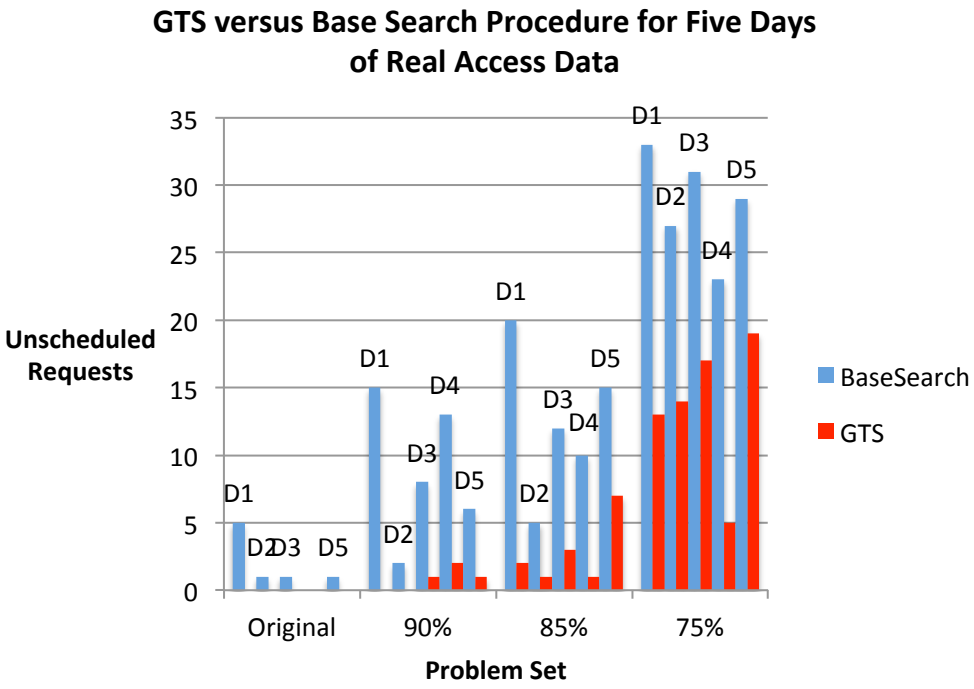
THE ROBOTICS INSTITUTE

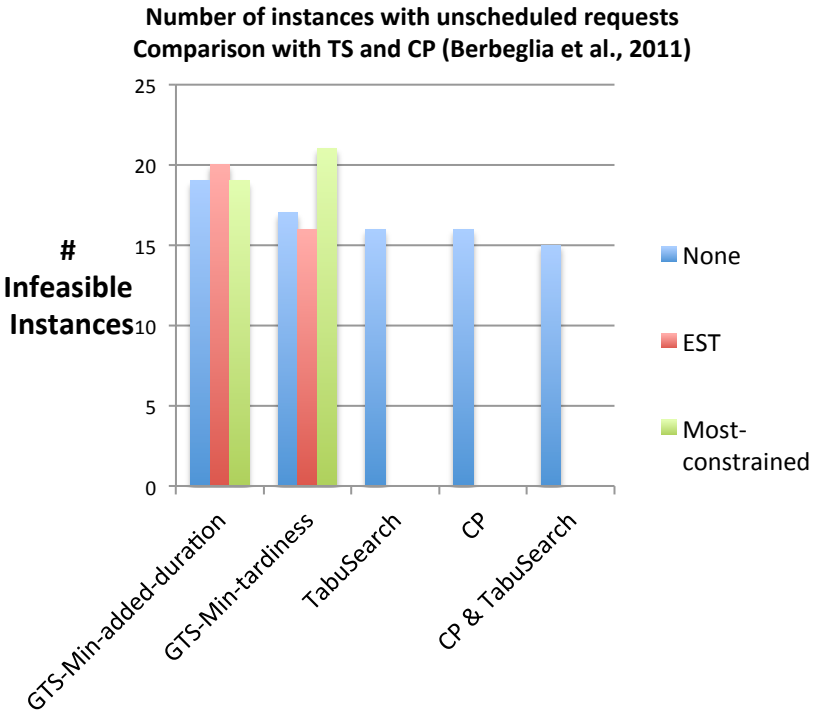
Carnegie Mellon

Generalized Task Swap

[Rubinstein, Smith, Barbulescu 2012]

- Extension to operate with arbitrary task networks
- Application to dynamic dial-a-ride problems





Results:

- Comparable performance to state of the art solvers
- > 50% reduction in unschedulable requests over baseline scheduler at ACCESS Transportation Systems

Outline of Talk

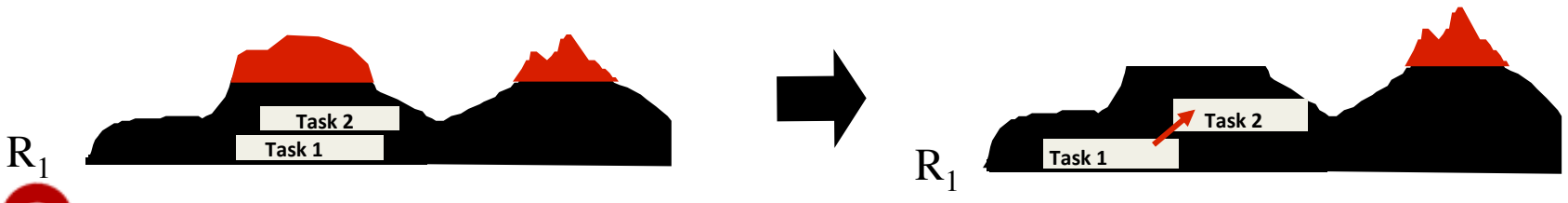
- Incremental Constraint-based Solution Generators
- Adding Optimizing Search Procedures
  - Heuristic-Biased Stochastic Sampling
  - Task Swap
- Generating Resilient Schedules
  - Hedging Against Temporal Uncertainty
  - Probabilistic Analysis of Deterministic Schedules
- Current Focus

Hedging Against the Possibility of Unexpected Task Behavior

Complementary Perspective: Build schedules that retain flexibility and can absorb some amount of unpredictability in execution

- task execution windows instead of precise times
- resource options instead of precise assignments
- process redundancy to increase likelihood of success

Basic Approach: *Partial-order scheduling procedures*

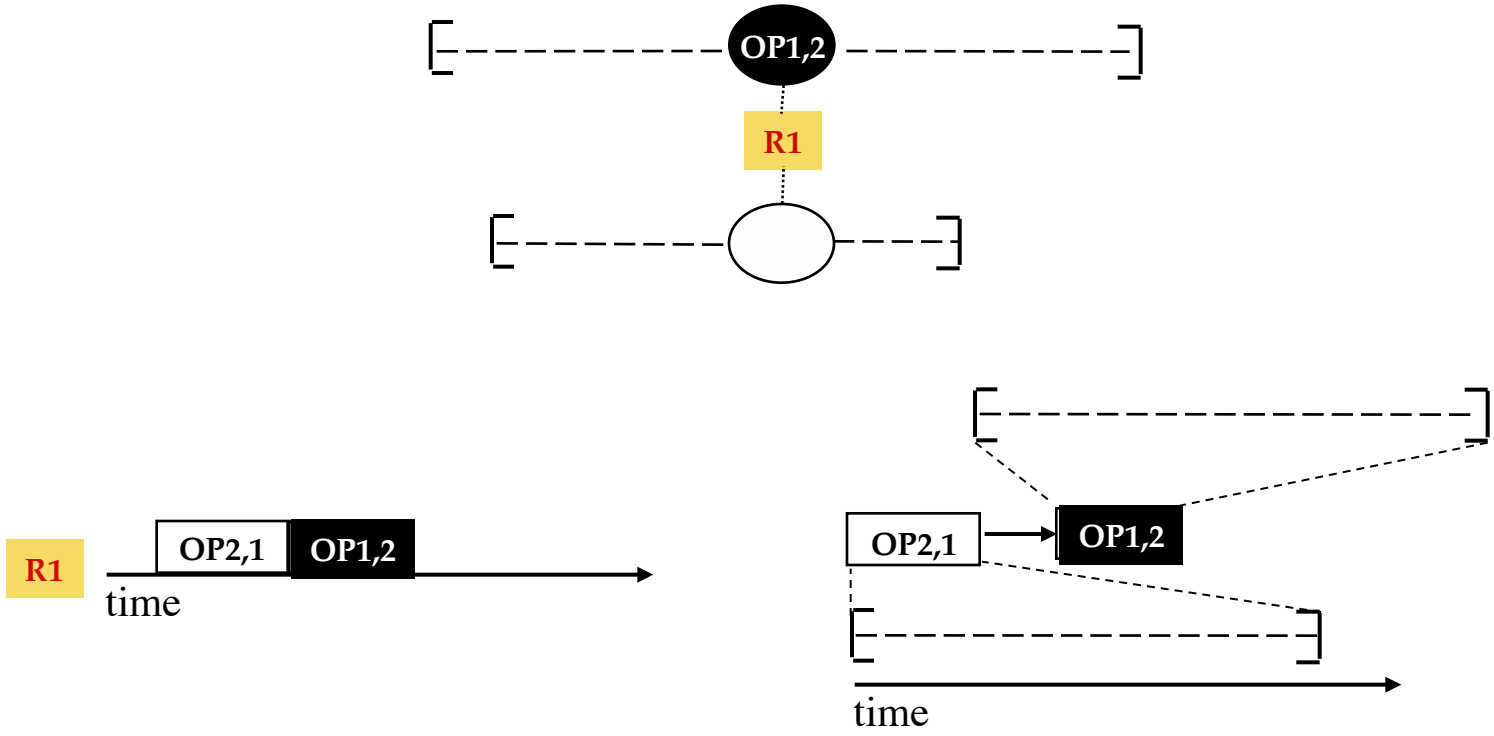


THE ROBOTICS INSTITUTE

Carnegie Mellon

Partial Order Schedules

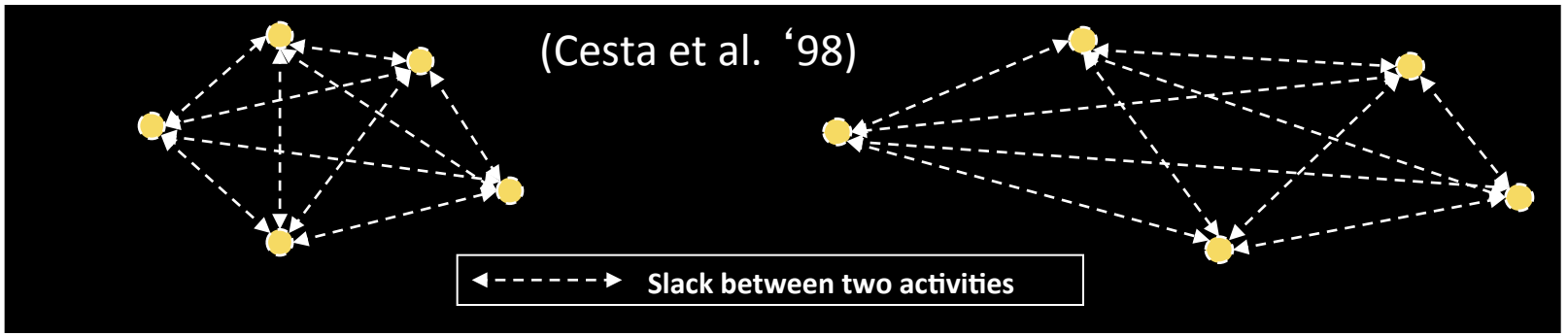
- Sequence activities that are competing for the same resources and let start and end times float



Characterizing Schedule Robustness

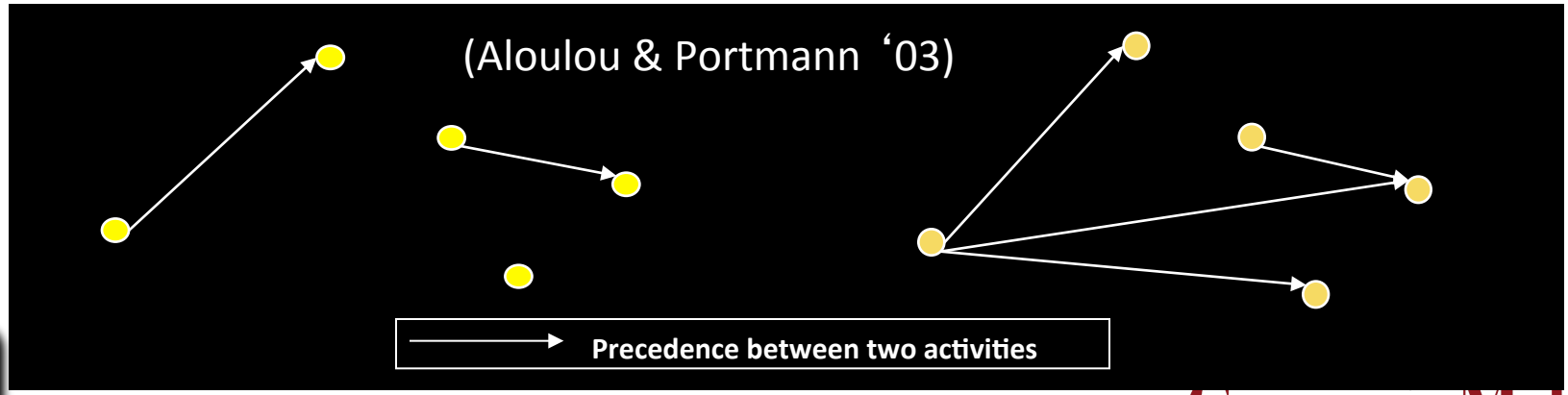
- **Fluidity:** *average slack in the schedule*

$$rb = \sum_{h \neq l} \frac{|d(e_{ah}, s_{al}) - d(s_{al}, e_{ah})|}{H \times N \times (N-1)} \times 100$$



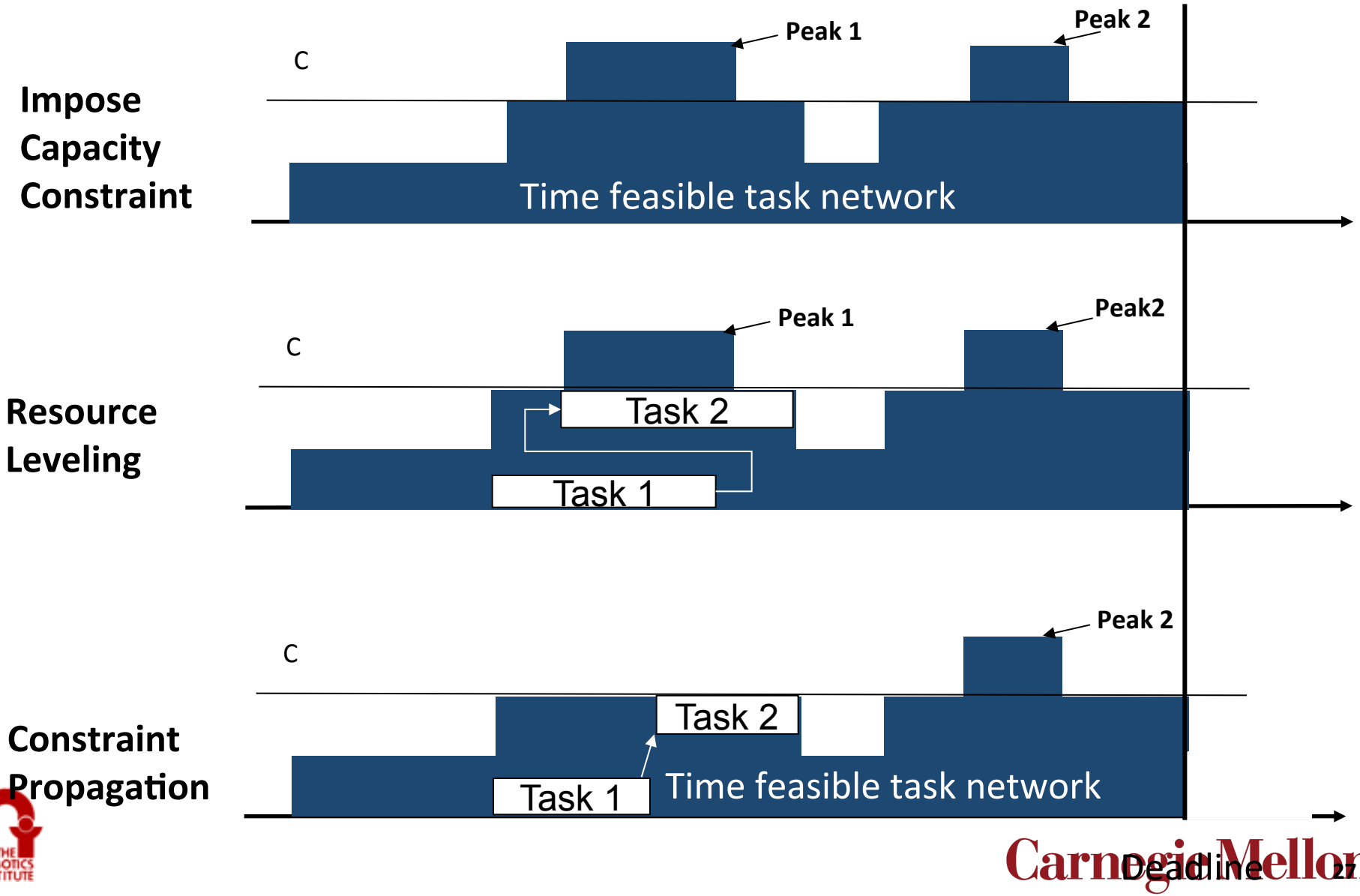
- **Flexibility:** *average nbr. of ordered activities*

$$flex = \sum_{i=1}^{N} \sum_{\substack{j=1: \\ dij \times dji < 0}}^{N} \frac{1}{N(N-1)}$$



Carnegie Mellon

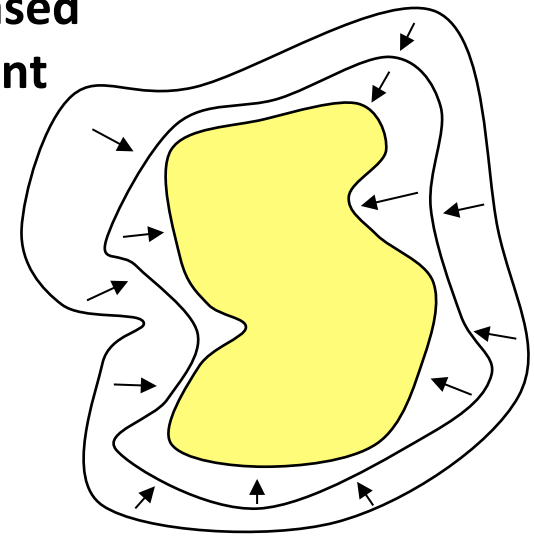
Basic Constraint-Posting Cycle

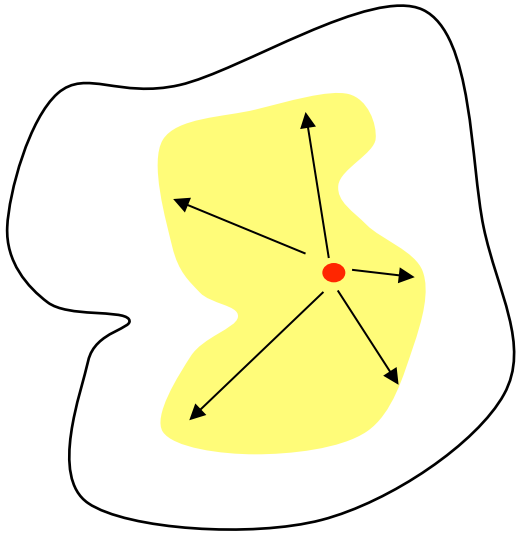


Building Robust Schedules: A Counter-Intuitive Result

[Policella, Smith, Cesta, Oddi - 2004]

Envelope-based commitment (EBA)





EST solution + generalization ($ESTA^C$)

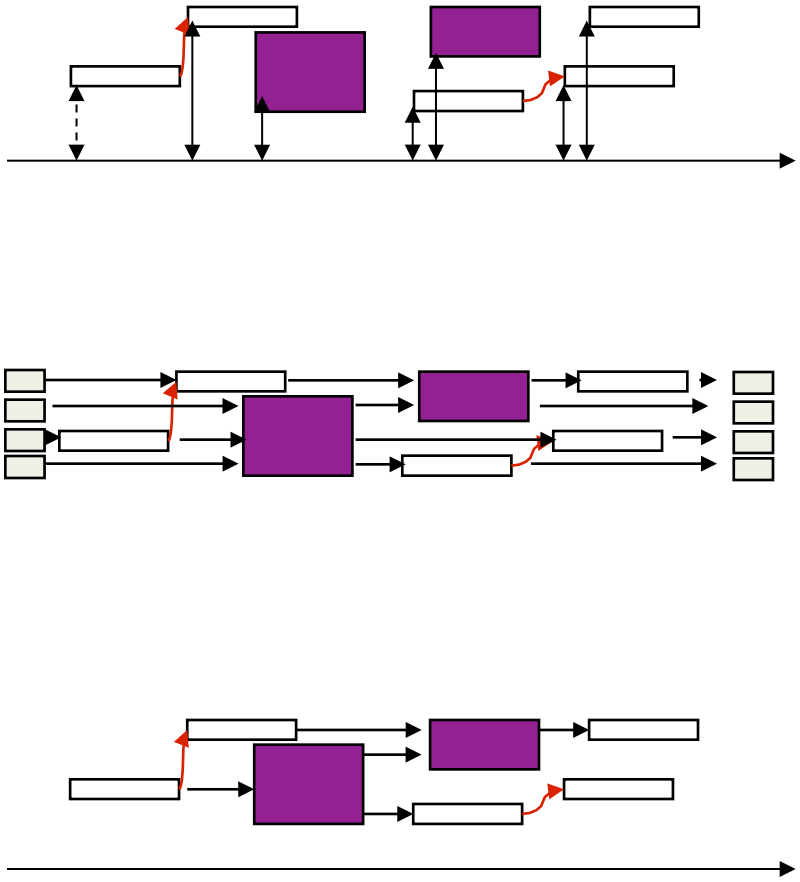
- Envelope-based approach can produce more *fluid* solutions but inefficient and generally ineffective
- Introduction of heuristic bias improves ability to find solutions but degrades fluidity
- $ESTA_C$ solution procedure generally dominates

Carnegie Mellon

ESTA^{C}: Earliest Start Time Analysis with Chaining

Two Step process

1. Generate a fixed-time solution using earliest start time profile
2. Transform solution into a partial order schedule (POS) through *chaining*



THE ROBOTICS INSTITUTE

Carnegie Mellon

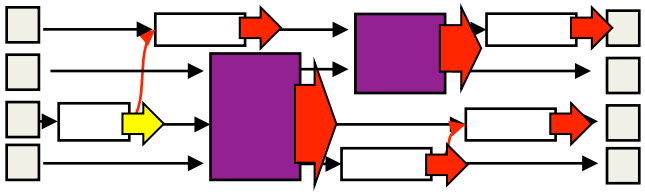
Some Theoretical Properties

- Chaining step is makespan, tardiness preserving
- No loss in generality in focusing on Chaining Form POSs

Searching for Robust Schedules

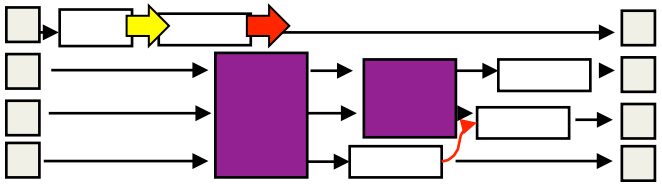
[Policella, Cesta, Oddi, Smith - 2004]

A Chaining Form POS



C=4

A Better Chaining Form POS



C=4

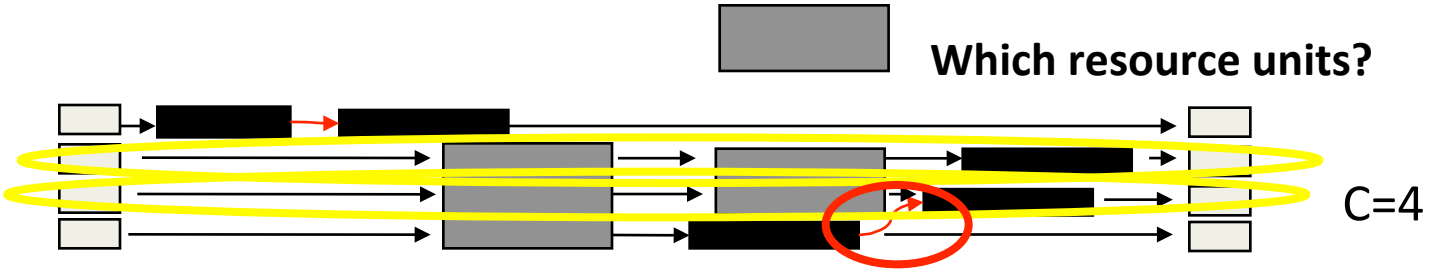
- Search in the space of *Chaining Form* solutions
  - Iterated application of stochastic chaining procedure
  - Robustness metric as objective criterion
  - Incorporation of chaining heuristics

THE ROBOTICS INSTITUTE

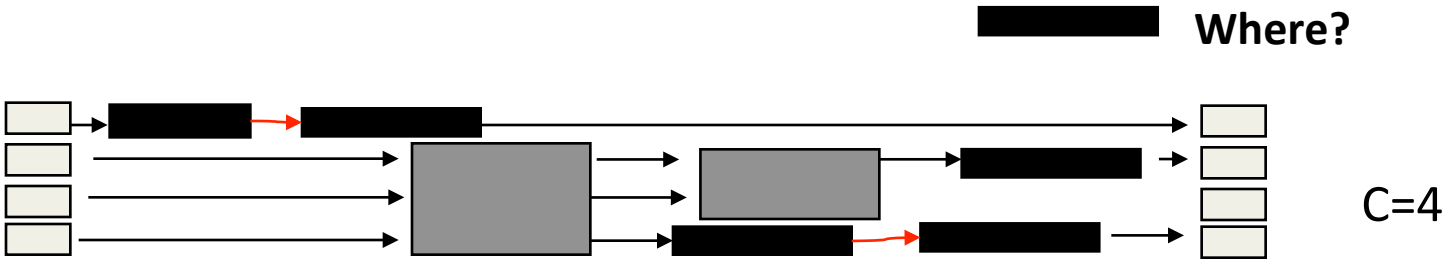
Carnegie Mellon

Chaining Heuristics

- Avoid synchronization points: *Put activities requiring multiple resource units on common chains*



- Allocate synergistically with problem constraints: *Put activities that are already ordered on the same chain*



Results

- Experimentation on Resource Constrained Project Scheduling Problem (RCPSP) Benchmarks previously solved by basic chaining procedure
- Improvement in robustness on both metrics:
  - 90% for Flexibility
  - 18% for Fluidity (where basic procedure already produces good results)
- Incorporation of chaining heuristics speeds up search process

Schedule Strengthening through Uncertainty Analysis

[Hiatt, Zimmerman, Smith, Simmons 2009]

- Issue: How to take advantage of knowledge about uncertainty
  - Reasoning with explicit models of uncertainty enables (expected) optimal solutions, but do not scale well
  - Deterministic (flexible times) scheduling is much more scalable but associated controllability models are overly conservative
- Basic Approach:
  - Construct partial order (flexible) schedule that maximizes quality under deterministic modeling assumptions
  - Use explicit analysis of uncertainty to identify weak points and take schedule strengthening actions

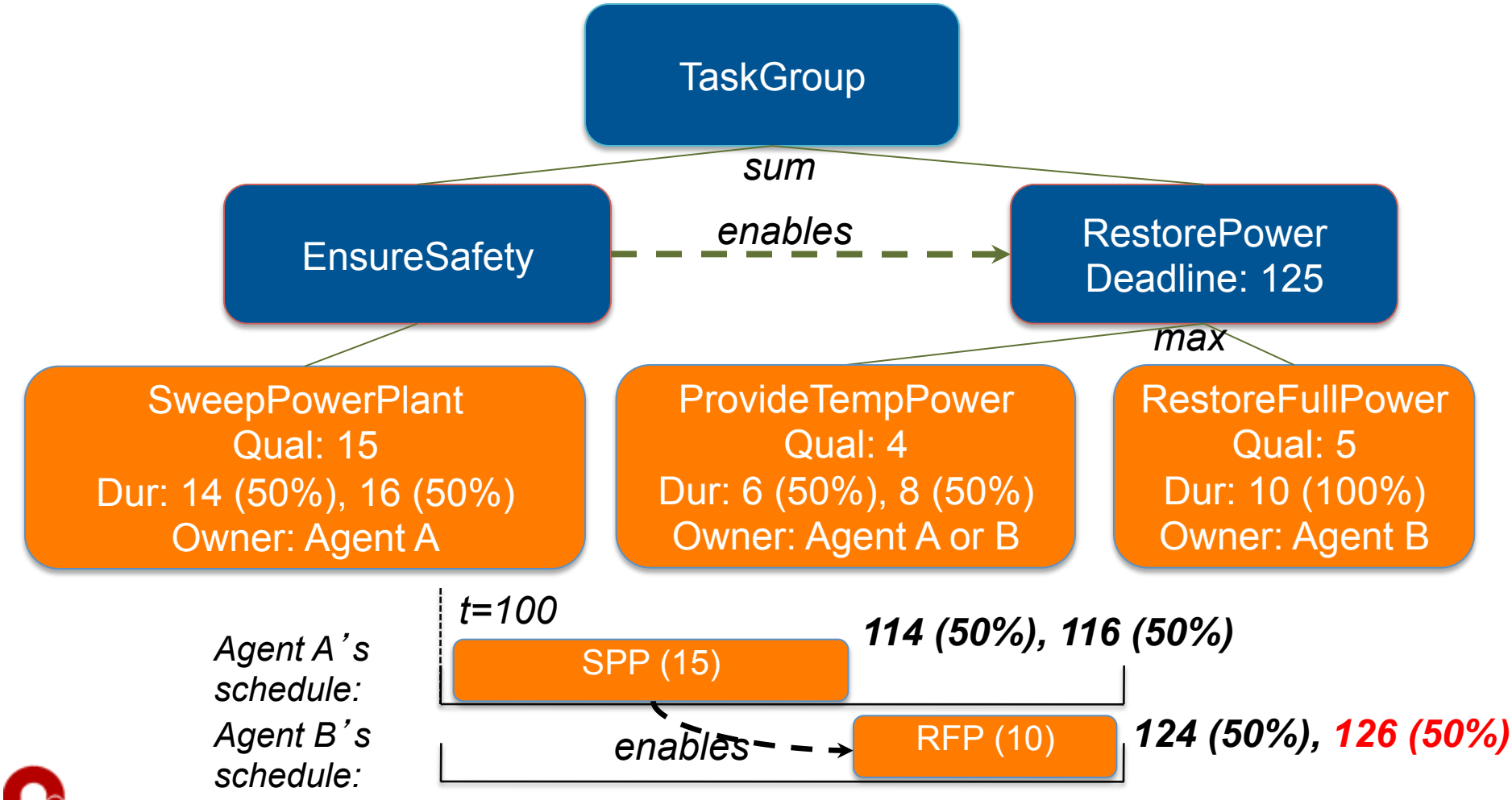
Strengthening Deterministic Schedules

- **Assumptions:**
  - Probability distributions associated with task durations and outcomes (payoff or "quality" accrued)
  - Tasks not completed by their deadline fail
- **Possible strengthening actions:**
  - Substitute tasks with lower failure likelihoods and/or shorter durations
  - Schedule redundant back-up tasks to increase probability of some level of success
- **Goal:** Boost likelihood that the quality projected by the deterministic schedule will be accrued during execution

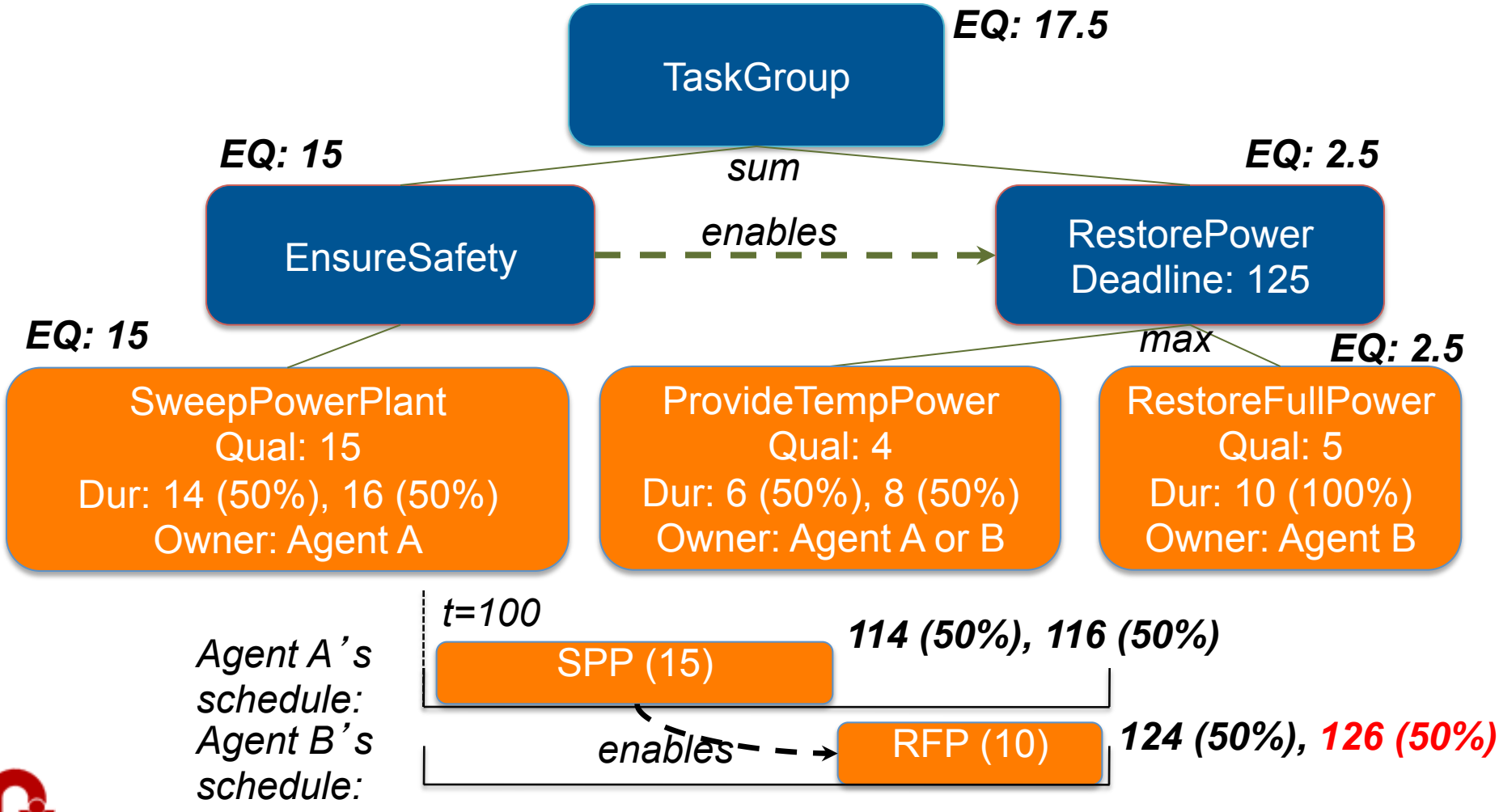
Probabilistic Analysis of Deterministic Schedules (PADS)

- Compute for each task, the probability of failing (achieving zero quality)
- For each task with non-zero probability, compute *expected quality loss* if it fails
- In order of expected quality loss, attempt to strengthen tasks
- Only keep strengthening actions that increase the schedule's *expected quality*

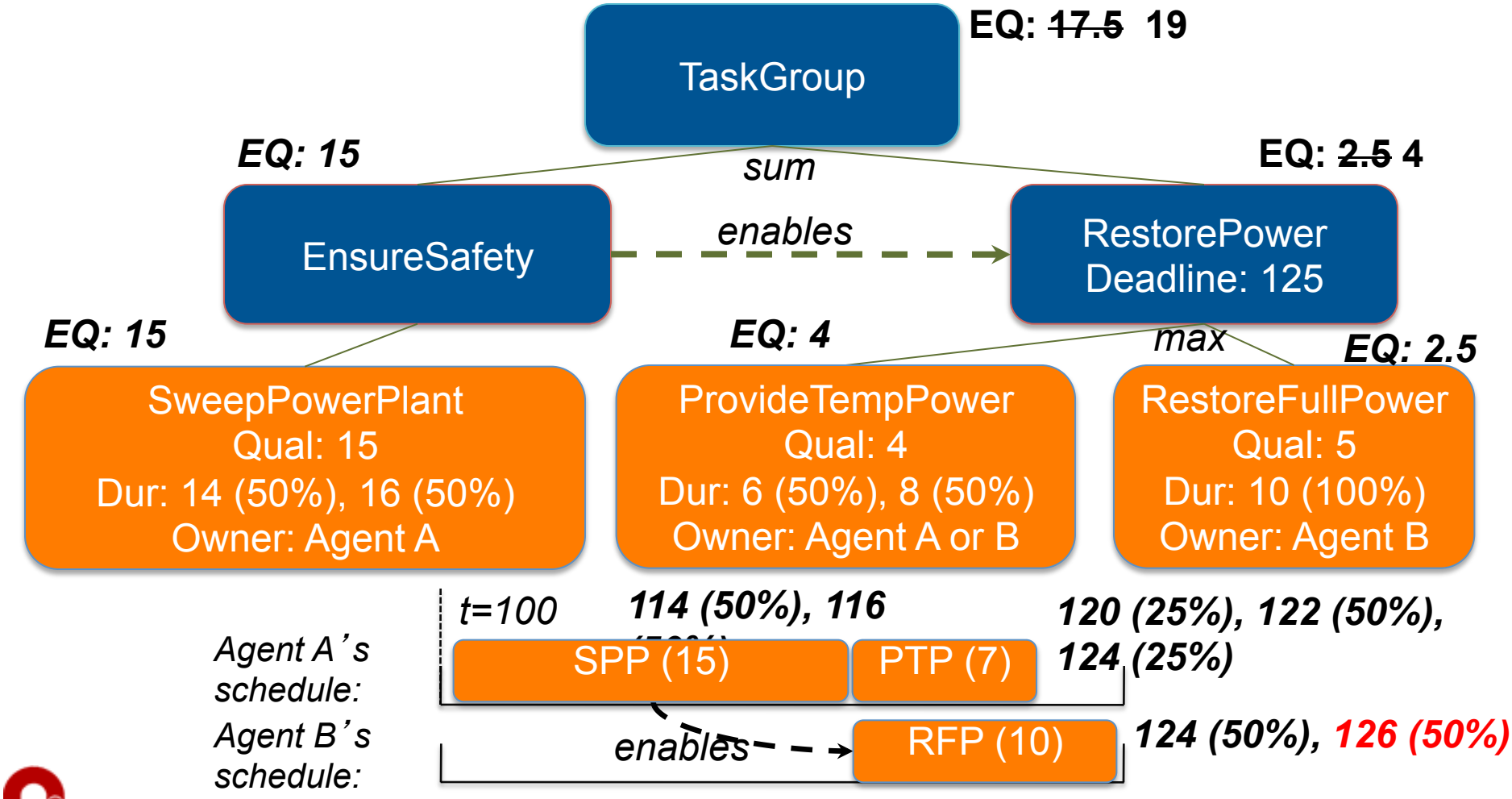
Probabilistic Analysis (PADS) Example



Probabilistic Analysis (PADS) Example



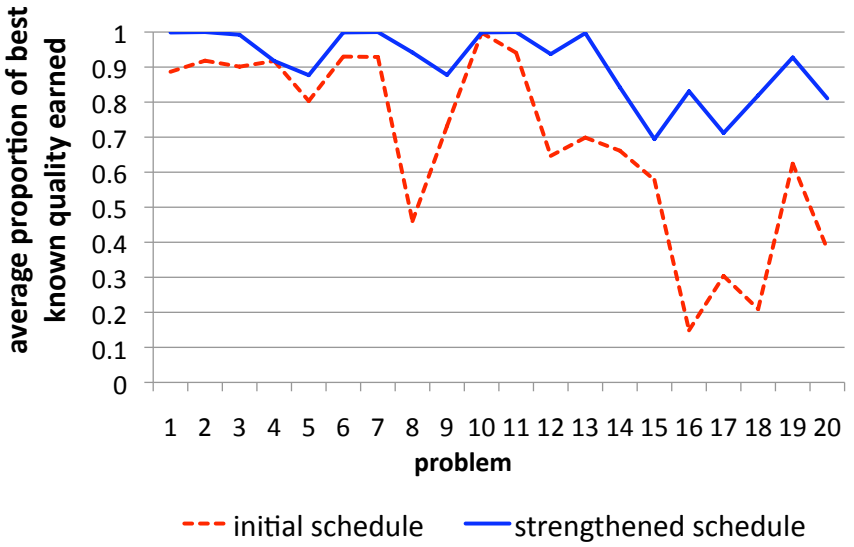
Schedule Strengthening Example



Executing Strengthened Schedules

- Simulated schedule execution under 2 starting conditions:
  - Initial deterministic schedule
  - Strengthened schedule
- No rescheduling allowed during execution (with basic conflict resolution on)
- Results:
  - Strengthened schedules earned on average 36% more quality *(p < 0.01)*

		# agents		
		10	20	25
# executable methods	< 200	1	4	0
	200-300	2	6	0
	301-400	0	5	0
	> 400	0	0	2



THE ROBOTICS INSTITUTE

Carnegie Mellon

Summary

- Scheduling in dynamic and uncertain domains requires a balance of pro-active and reactive decision-making
  - In multi-actor settings, the need for incremental change and solution stability has implications for the approach taken to optimization
  - Pro-active consideration of uncertainty can reduce the burden of execution-time schedule management
- Constraint-based planning and scheduling models support both of these complementary perspectives

Current Focus

- Distributed coordination and optimization Frameworks
  - Adaptive Traffic Signal Control
  - Automated Manufacturing
  - Task Allocation in Distributed Sensor Networks
- Learning task models and constraints
- Scheduling to maximize resilience to constraint violations

Thanks to My Collaborators in this Work

Laura Barbulescu, Marcel Becker, Amedeo Cesta, Vince Cicirello, Laura Hiatt, David Hildum, Larry Kramer, Angelo Oddi, Nicola Policella, Ricardo Rasconi, Zack Rubinstein, Reid Simmons, Terry Zimmerman

END