Scheduling Using Computational Intelligence

Russell C. Eberhart, Ph.D.
Associate Dean for Research
Purdue School of Engineering and Technology
Indiana University Purdue University Indianapolis
eberhart@engr.iupui.edu

Daniel Reiss CEO Automated Terminal Systems

ATS
Automated Terminal Systems

Purdue University
IUPUI

Outline

- Why computational intelligence?
- The components of computational intelligence and their applications
- Definitions of computational intelligence
- Example of computational intelligence system
- Schedule optimization using computational intelligence
- Particle swarm optimization
- Scheduling system for integrated automated container terminal (IACT)

Why Computational Intelligence?

- Solve problems previously intractable
- Rapid prototyping
- Low cost
- Use existing PC platforms
- CI systems are robust; no “mesa effect”

A T S
Automated Terminal Systems

purdue university
IUPUI

Components of Computational Intelligence

- Artificial neural networks
- Evolutionary computation algorithms
- Fuzzy logic
- Knowledge “tidbits”

A T S
Automated Terminal Systems

IUPUI

Artificial Neural Networks

- Analysis paradigms very roughly modeled after the massively parallel structure of the brain
- Simulate highly interconnected, parallel computational structures with numerous relatively simple individual *processing elements*

Neural Network Application Areas

- Classification
- Associative memory
- Clustering or compression
- Simulation or composition
- Control systems

Fuzzy Logic

- Non-statistical imprecision and vagueness in information and data
- *Fuzzy sets* model the properties of imprecision, approximation, or vagueness
- *Fuzzy membership values* reflect the membership grades in a set
- *Fuzzy logic* is the logic of approximate reasoning; it is a generalization of conventional logic

Fuzzy Logic Application Areas

- Control systems
  - vehicles
  - home appliances
- Fuzzy expert systems
  - industrial processes
  - diagnostics
  - finance
  - robotics and manufacturing

Evolutionary Computation

- Machine learning optimization and classification paradigms roughly based on mechanisms of evolution such as natural selection and biological genetics
- Major paradigms are genetic algorithms, evolutionary programming, evolution strategies, genetic programming, and particle swarm optimization

Evolutionary Computation Application Areas

- Optimization
  - design
  - scheduling
- Classification
  - diagnosis
- Discovering computer programs

IUPUI

Evolutionary Algorithm Process

1. Initialize population of potential solutions (usually randomly).
2. Evaluate fitness of each population member.
3. Reproduce new population.
4. Apply evolutionary algorithm operators (such as crossover, mutation, etc.).
5. Terminate process if some condition is met.
6. Go to step 2.

Definition of Intelligence

The capability of a system to adapt its behavior* to meet its goals in a range of environments. It is a property of all purpose-driven decision makers.

- David Fogel

* implement decisions

Computational Intelligence: Definition I

A methodology involving computing that exhibits an ability to learn and/or deal with new situations, such that the system is perceived to possess one or more attributes of *reason* , such as generalization, discovery, association, and abstraction.

Silicon-based computational intelligence systems usually comprise hybrids of paradigms such as artificial neural networks, fuzzy systems, and evolutionary algorithms, augmented with knowledge elements, and are often designed to mimic one or more aspects of carbon-based biological intelligence.

Computational Intelligence: Definition II

Computational intelligence comprises practical **adaptation** concepts, paradigms, algorithms, and implementations that enable or facilitate appropriate actions (intelligent behavior) in complex and changing environments.

CI Example: Evolutionary Fuzzy Expert Systems

- Evolve fuzzy membership functions and fuzzy rules
- Simultaneously adapt parameters in evolutionary algorithm using fuzzy logic
- Can also include artificial neural network models, etc.

Schedule Optimization Using Computational Intelligence

- One of the most common applications of evolutionary algorithm optimization
  - quick to prototype
  - runs fast
- Schedule optimization is an *NP-Complete* problem
- NP-Complete: Sufficiently complex such that any deterministic search technique that completely searches the problem domain will almost certainly not find an acceptable answer in an acceptable time

A T S
Automated Terminal Systems

Purdue University
IUPUI

The Scheduling Problem

- Schedulers usually schedule tasks
- A task is a time and resource requirement
- A basic decision for any scheduling project is the selection of time resolution
- More time is spent on *rescheduling* than on scheduling

A T S
Automated Terminal Systems

Purdue University
IUPUI

Constraints

- Two major types of constraints exist: hard (strong) and soft (weak)
- Hard constraints *cannot* be violated by the scheduler
  - example: scheduled maintenance
  - can sometimes be over-ridden manually
- Soft constraints can be violated with a penalty assigned to the fitness function
  - examples: time and resource preferences

IUPUI

Major Components of Scheduling System

- Schedule builder
  - major job is to build “legal” schedules
  - may take (some) soft constraints into account
- Schedule evaluator
  - calculates fitness value for each schedule
  - fitness value assigned according to algorithm
- Schedule optimizer
  - constructs a task order list
  - can be domain independent or domain dependent

IUPUI

Introduction to Particle Swarm Optimization

- A “swarm” is an apparently disorganized collection (population) of moving individuals that tend to cluster together while each individual seems to be moving in a random direction
- We also use “swarm” to describe a certain family of social processes

Introduction to Particle Swarm Optimization (PSO), Continued

- A concept for optimizing nonlinear functions
- Has roots in artificial life and evolutionary computation
- Developed by Kennedy and Eberhart (1995)
- Simple in concept
- Easy to implement
- Computationally efficient
- Effective on a variety of problems

A T S
Automated Terminal Systems

purdue university
IUPUI

Features of Particle Swarm Optmization

- Population initialized by assigning random positions *and* velocities; potential solutions are then *flown* through hyperspace.
- Each particle keeps track of its "best" (highest fitness) position in hyperspace.
  - This is called "pbest" for an individual particle
  - It is called "gbest" for the best in the population
  - It is called "lbest" for the best in a defined neighborhood
- At each time step, each particle stochastically accelerates toward its pbest and gbest (or lbest).

Particle Swarm Optimization Process

1. Initialize population in hyperspace.
2. Evaluate fitness of individual particles.
3. Modify velocities based on previous best and global (or neighborhood) best.
4. Terminate on some condition.
5. Go to step 2.

PSO Velocity Update Equations

- Global version:

$$v_{id} = w_i v_{id} + c_1 rand()\left(p_{id} - x_{id}\right) + c_2 Rand()\left(p_{gd} - x_{id}\right)$$

$$x_{id} = x_{id} + v_{id}$$

Where ***d*** is the dimension, $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are positive constants, ***rand*** and ***Rand*** are random functions, and ***w*** is the inertia weight.

For neighborhood version, change $\boldsymbol{p_{gd}}$ to $\boldsymbol{p_{ld}}$.

A T S

Automated Terminal Systems

IUPUI

Further details of PSO

- Performance of each particle measured according to a predefined fitness function.
- Inertia weight influences tradeoff between global and local exploration.
- Good approach is to reduce inertia weight during run (i.e., from 0.9 to 0.4 over 1000 generations)
- Usually set c_1 and c_2 to 2
- Usually set maximum velocity to dynamic range of variable

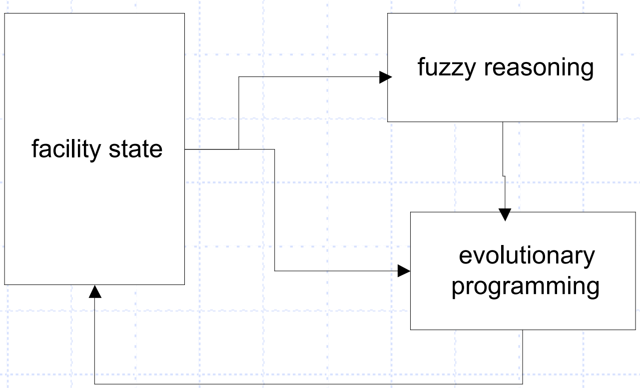
Summary

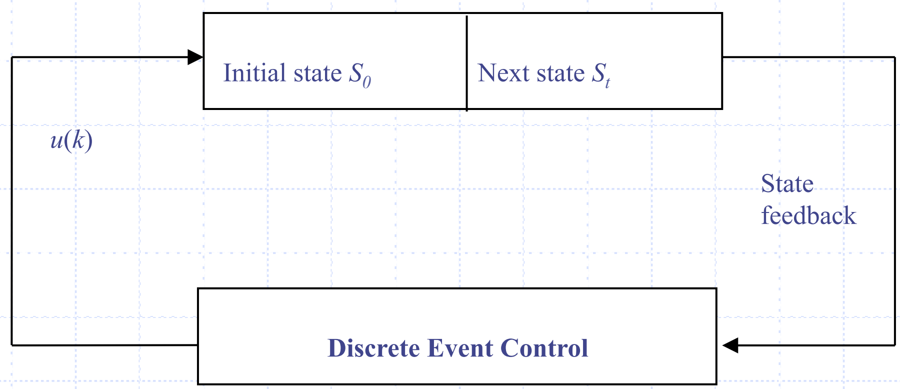
- Computational intelligence is a powerful method for building scheduling systems
  - quick to prototype
  - outperforms other approaches
  - fast to run
- Example: use particle swarm optimization at the core of the process, with fuzzy expert system shell

IUPUI

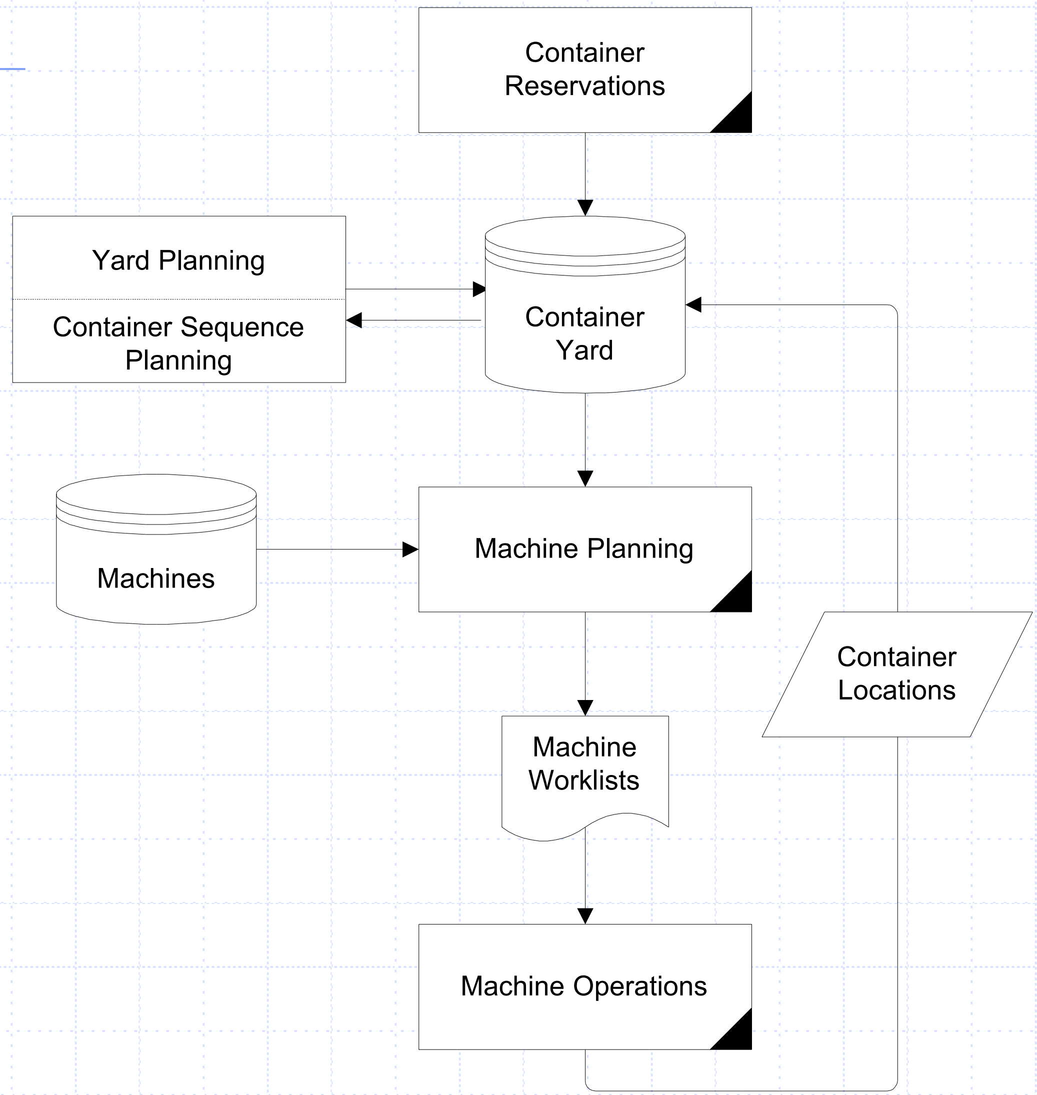
Scheduling System for Integrated Automated Container Terminal

- Objective - develop planning and scheduling algorithm for fully integrated automated container terminals
- Approach - Fuzzy system and evolutionary programming

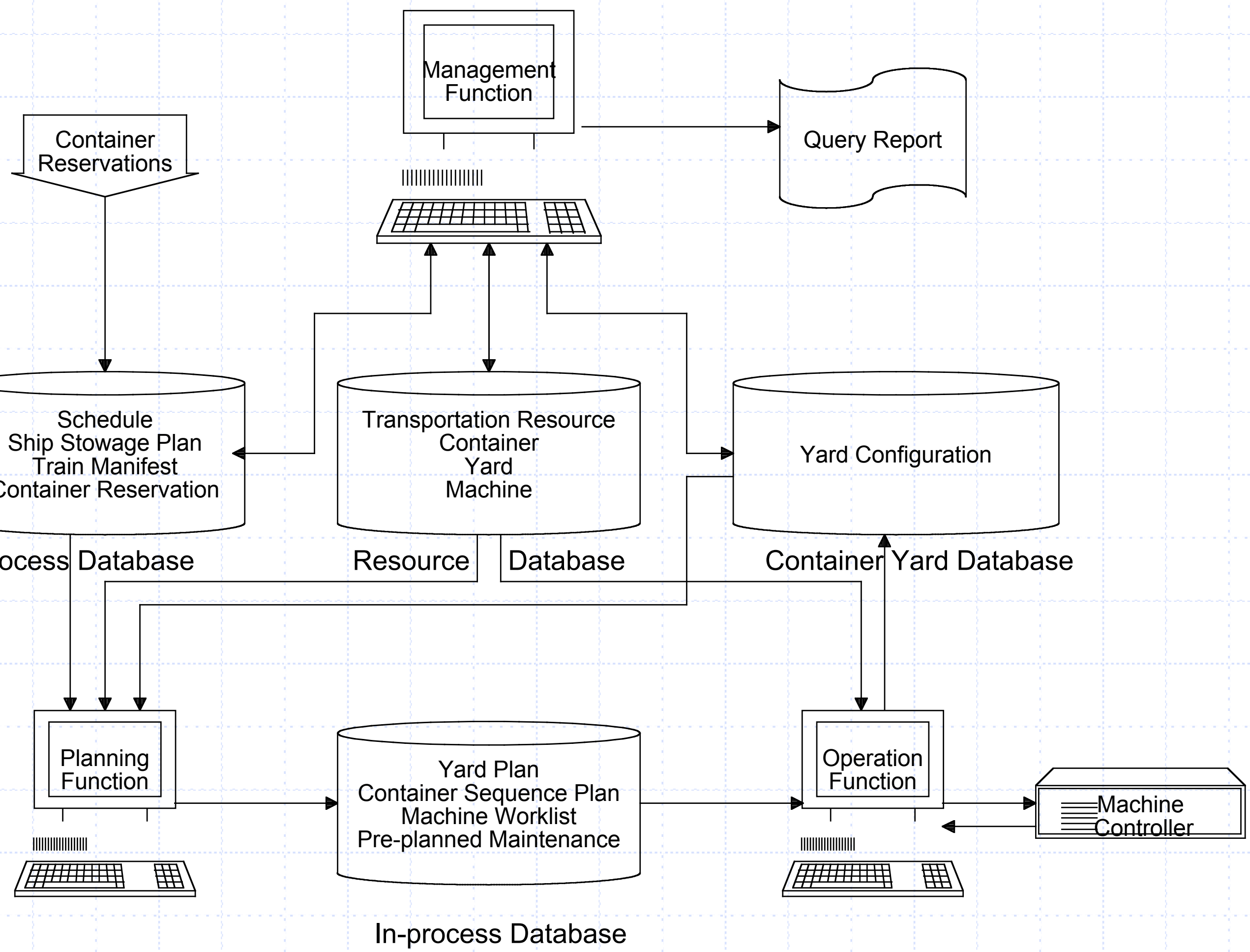




Scheduling System for IACT – Workflow

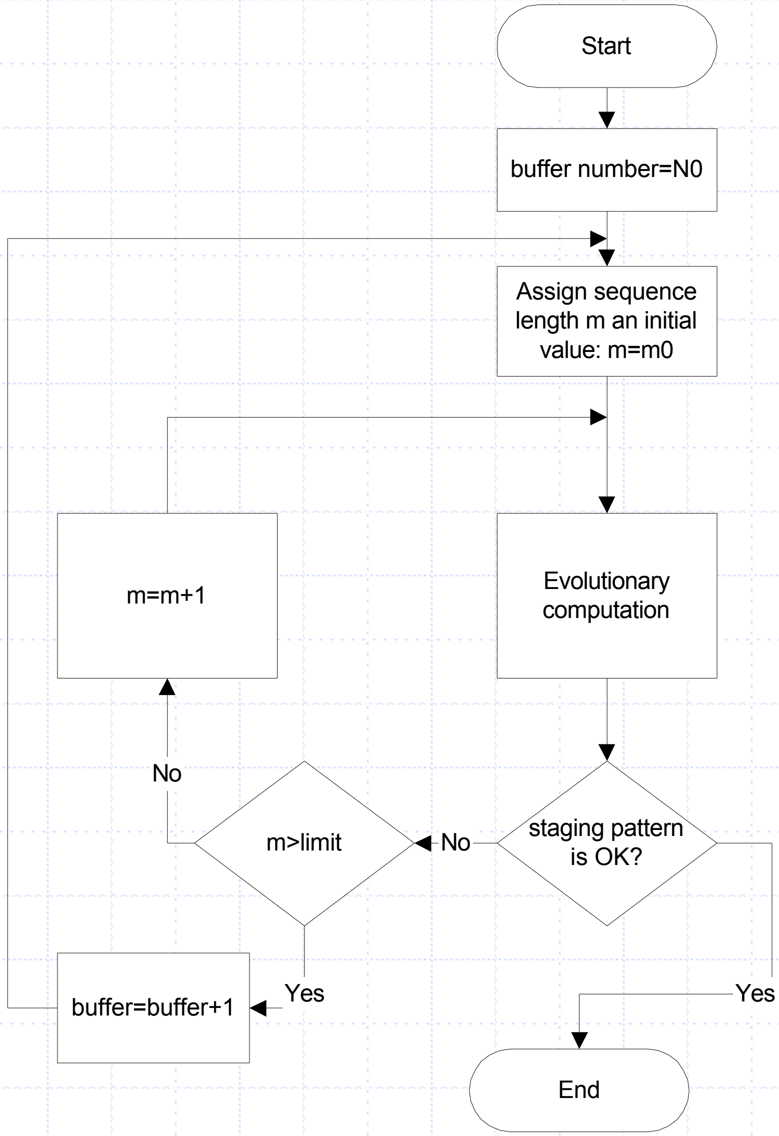


Scheduling System for IACT – System Overview



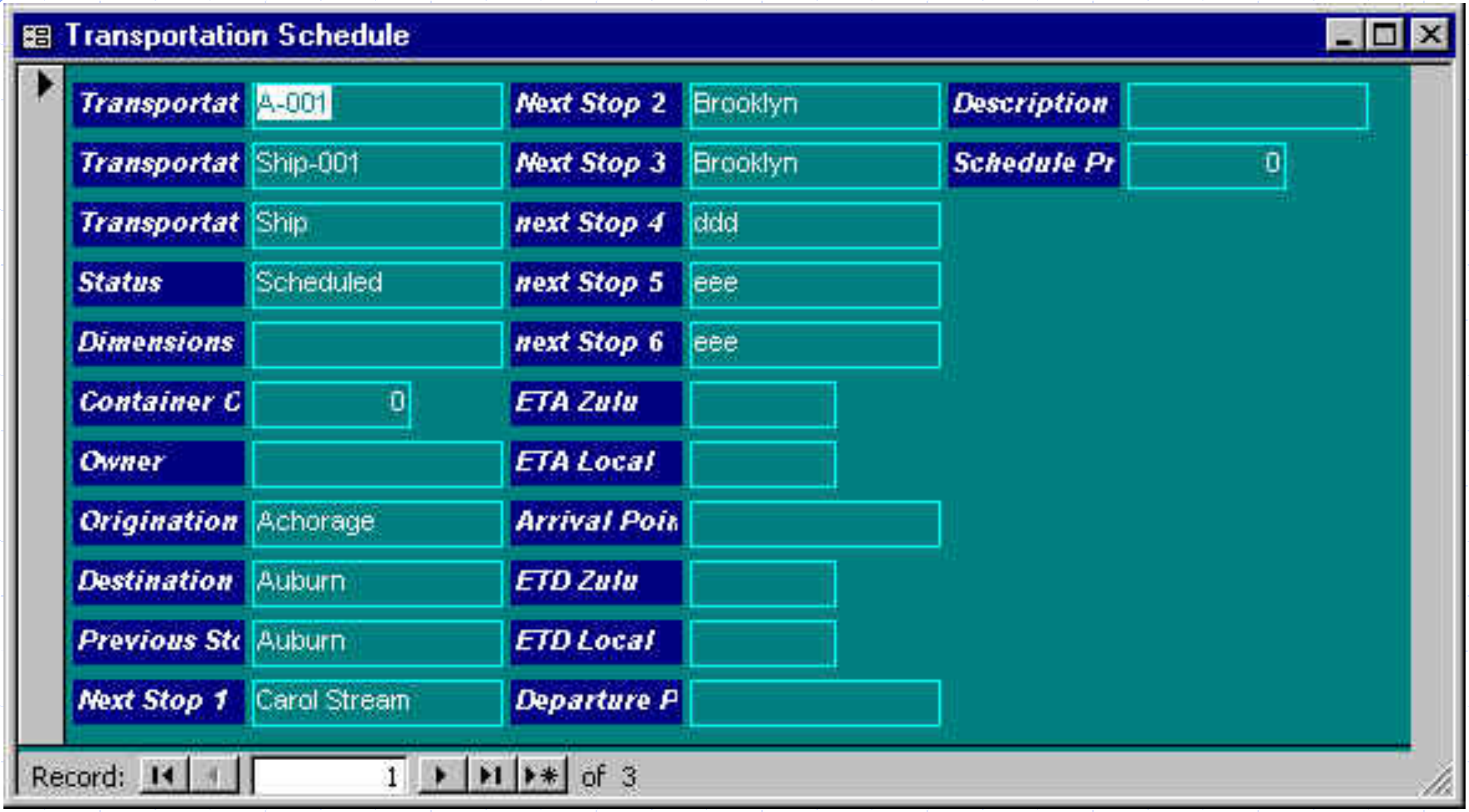
IUPUI

Scheduling System for IACT – Algorithm



O-Port: Yard Planning and Scheduling System

Transportation Resource Information



A T S

Container Schedule Information

Container Schedule

Booking No	Container ID	Status	Yard Location 1	Yard Location 2
A-01	IACT-0101-0101	On the Yard	D-01-01-01	

Yard Location 3	Chassis ID	Container Type	Cargo Type	L/UL Requirements	Length
		Dry	Dry		20

Height	Width	Dimension Exception	Owner	Gross Weight	Handling	Handling Priorit
8	8					0

Origination	Destination	Previous Stop	Next Stop	Arrival Transportation Resource Typ
Carol Stream	Achorage	Binghampton	Calgary	Ship

Arrival Transportation Schedule No	Location on Arrival Transportaiton Resource	Departure Transportation Resource Type
		Ship

Departure Transportation Schedule No	Location on Departure Transportaiton Resource	Description	Schedule Priority
			0

Booking No	Container ID	Status	Yard Location 1	Yard Location 2
A-01	IACT-0101-0201	On the Yard	D-01-01-02	

Yard Location 3	Chassis ID	Container Type	Cargo Type	L/UL Requirements	Length
		Dry	Dry		20

Height	Width	Dimension Exception	Owner	Gross Weight	Handling	Handling Priorit
8	8					0

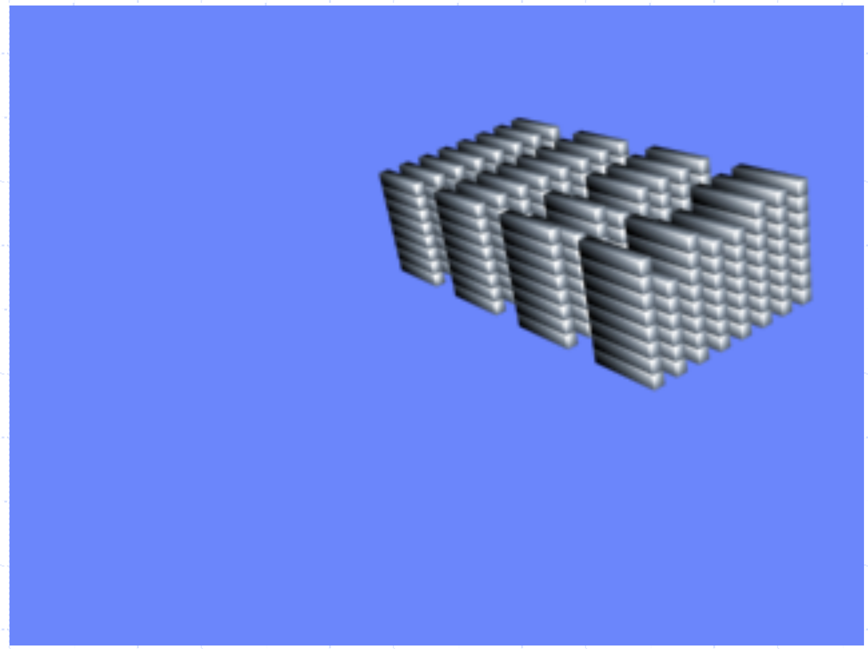
Origination	Destination	Previous Stop	Next Stop	Arrival Transportation Resource Typ

Arrival Transportation Schedule No	Location on Arrival Transportaiton Resource	Departure Transportation Resource Type

Departure Transportation Schedule No	Location on Departure Transportaiton Resource	Description	Schedule Priority
			0

Container Planning Sequences - Animation

- **250** Containers
- Move from yard to staging area along the berth
- Planning results
- Number of movements:



256

IUPUI