Genetic Algorithms for Selecting and Scheduling Waterway Projects

Shiaaulir Wang and Paul Schonfeld
University of Maryland, College Park

NETS SYMPOSIUM
Salt Lake City
January 13, 2006

Outline

- Optimization
- Simulation-Based Optimization
- Genetic Algorithms
- Project Scheduling Problems
- Scheduling Waterway Projects: Example with SIMOPT
- Conclusions

Optimization

- Optimization Process
  - A mathematical process that searches for the solution which best satisfies a stated objective
- Optimization Problems
  - Constrained vs. unconstrained
  - Continuous vs. discrete (combinatorial)
  - Deterministic vs. stochastic

- Optimization Methods
  - Exact Optimization
    - Calculus
    - Enumerative Search
    - Mathematical Programming
    - Branch and Bound algorithms
  - Heuristic optimization
    - Simulated Annealing
    - Tabu Search
    - Genetic Algorithms
    - Swarm Intelligence

- Decision Variables
  - Serve as the controllable variables determining the value of the objective function.
  - Discrete or continuous
    - Binary variables (Yes/No or 1/0)
      - Example: which projects are selected
    - Continuous variables
      - Example: when (if ever) is each additional project implemented

- Formulation of an Optimization Problem
  - Minimize or maximize objective function, e.g.,
    - Minimize cost, loss, delays, risk of failure, etc.
    - Maximize profit, net benefits, product quality, speed of service, etc.
  - Subject to constraints, e.g.,
    - On budgets, capacities, queue lengths, waiting times, construction times or facility closure times.

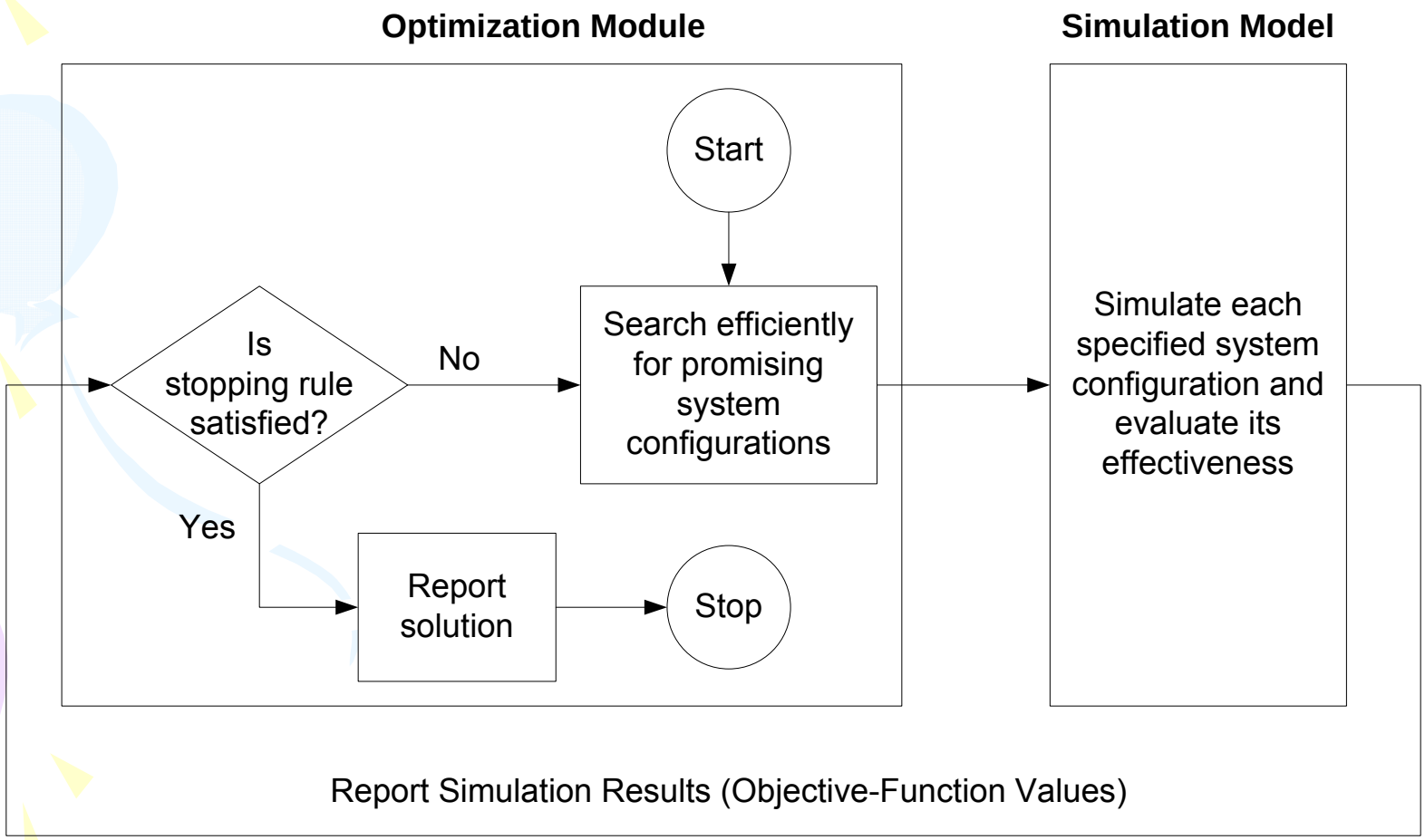
- Issues in Solving Complex Optimization Problems
  - The objective function must be repeatedly evaluated
  - The objective function might computed with a simple equation, a queuing model, or various other methods.
  - If the system analyzed is complex enough and subject to probabilistic variations, it is difficult to evaluate without a detailed simulation model.

Simulation-Based Optimization

- Why Use Simulation?
  - Complex probabilistic systems are difficult to evaluate analytically
  - For complex probabilistic systems, the objective function may not be fully specifiable
  - System performance can be estimated under numerous conditions specified in detail

- Operations with alternative design and control characteristics can be compared
- Experimental scenarios can be carefully controlled
- Systems undergoing many changes over time can be studied

- Interaction between Simulation and Optimization



- Combining Simulation and Optimization
  - The simulation model evaluates and computes the objective function for each analyzed combination of decision variables.
  - Based on the above results, the optimization model selects new combinations of decision variables to be simulated until further improvements become insignificant.
  - The optimization model also insures that all constraints are satisfied.

- Optimization process
  - Step 1: Create initial solution(s)
  - Step 2: Evaluate those solution(s)
  - Step 3: Apply search algorithm to generate new solution(s) by evaluating previous solution(s) and determining new search directions
  - Step 4: Check the termination rule
    - Enough iterations?
    - Enough evaluated solutions?
    - Unchanged search results over n iterations?
  - Step 5: If the termination rule is satisfied, STOP. Otherwise, return to step 3

- Issues
  - Heuristic search algorithms applied in large-scale combinatorial optimization usually do not guarantee an absolute global optimum, but only a near-optimal solution.
  - The difference between global optimum and near optimal solution is usually insignificant, considering the uncertainties in inputs and in functional relations.

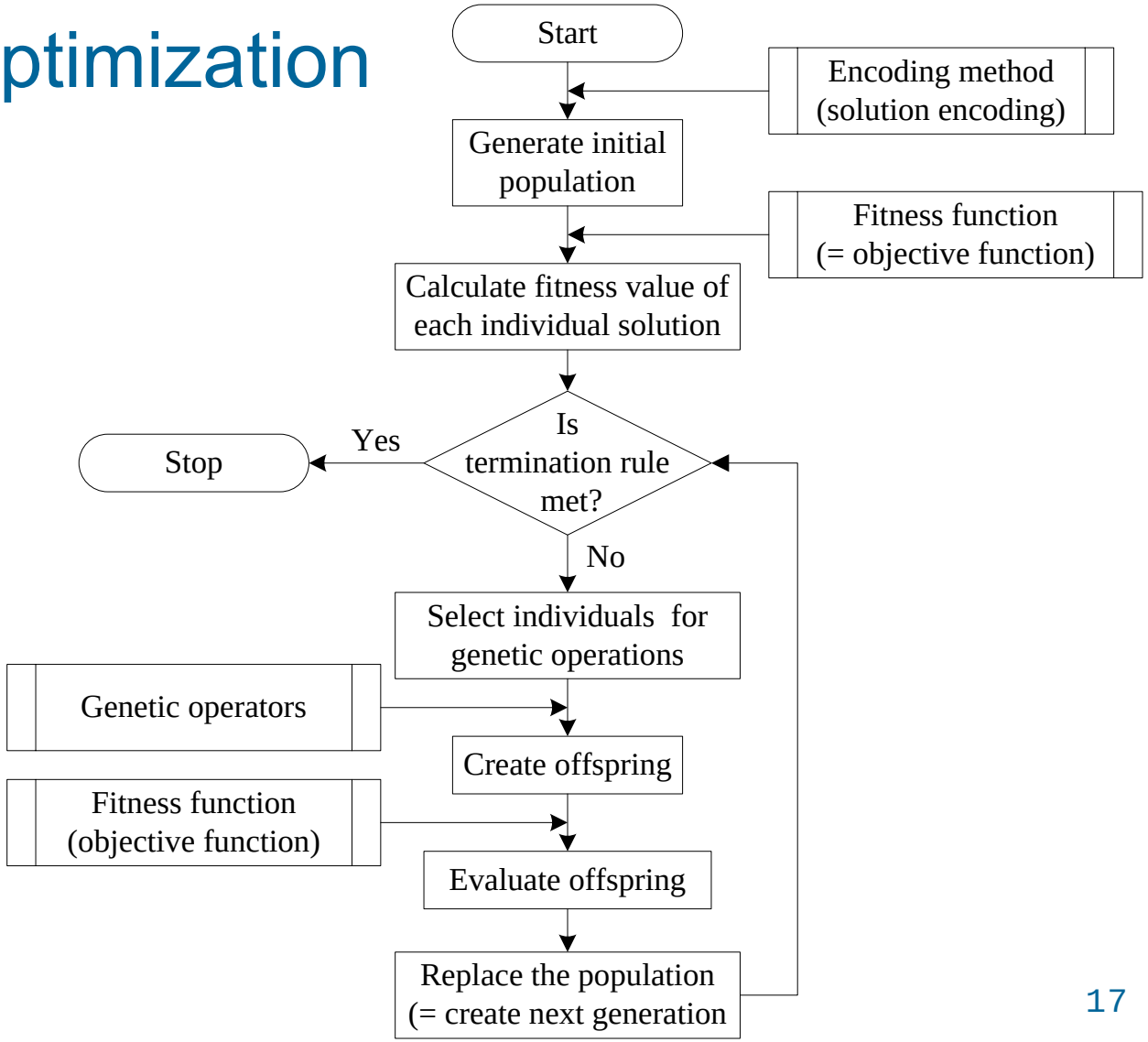
Genetic Algorithms

- Background
  - Rooted in natural genetics and computer science
  - Treat problem as the environment, and consider set of solutions as the population
  - Mimic natural evolution process
  - Encode each solution as chromosome
  - Better solutions are more likely to survive and reproduce

- Characteristics
  - GAs search through an entire population of solution at each iteration (generation)
  - GAs are especially effective in avoiding numerous local optima
  - GAs can handle any kind of objective functions (including very noisy ones)

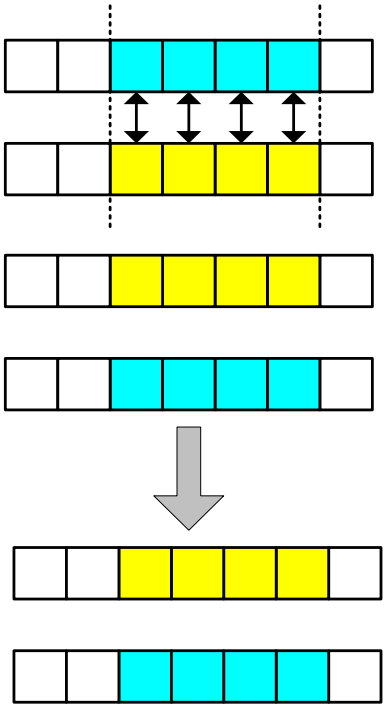
- Design of GAs
  - Solution Encoding (e.g., Integer sequence for scheduling problem)
  - Initial Population
  - Fitness Function (i.e., objective function)
  - Genetic Operators
    - Mutation operators
    - Crossover operators
    - Problem-specific operators
  - Termination and Convergence

- Genetic Optimization

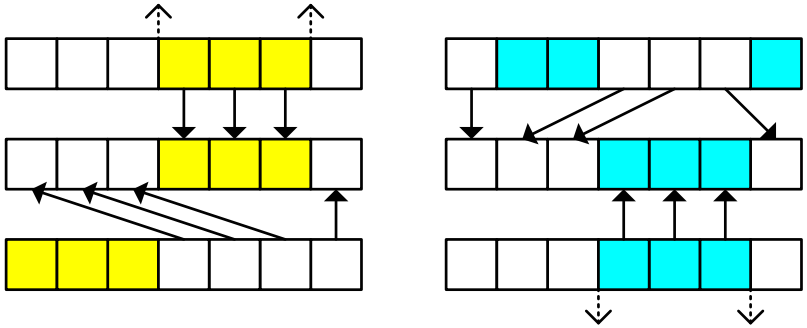


- Genetic Search
  - Step 1: Create initial population of solutions (i.e., project sequences).
  - Step 2: Evaluate those solutions with evaluation model.
  - Step 3: Select the better individual solutions for genetic refinement.
  - Step 4: Create new solutions using mutation, crossover, or other operators.
  - Step 5: Evaluate new solutions with evaluation model.
  - Step 6: Replace most or all previous solutions in the population.
  - Step 7: Stop if the termination rule is satisfied. Otherwise, return to step 3

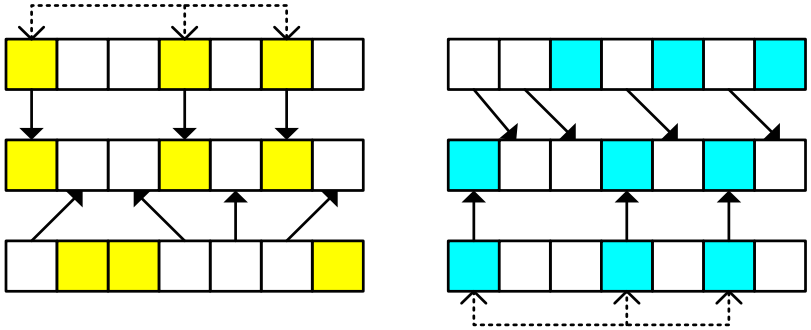
- Crossover operators
  - Partial-Mapped Crossover (PMX)



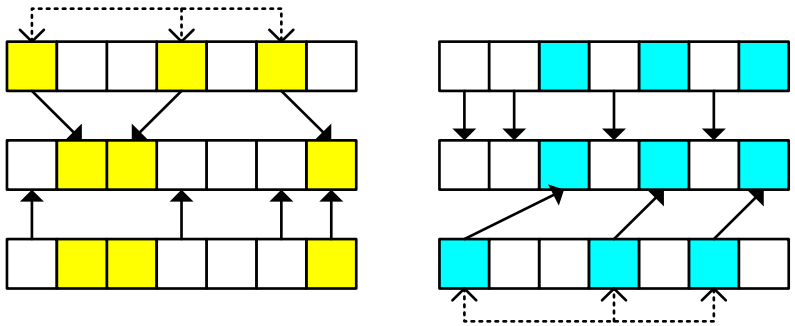
– Order Crossover (OX)



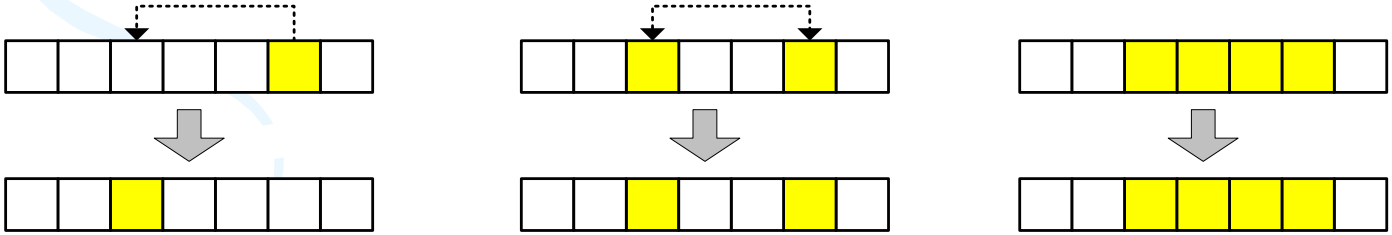
– Position-Based Crossover (PBX)



– Order-Based Crossover (OBX)



- Mutation operators



Insertion Mutation ***Exchange Mutation*** ***Inversion Mutation***

Project Scheduling

- Four steps:
  - Project Evaluation
  - Project Selection
  - Project Sequencing
  - Project Scheduling

- Project Interdependence
  - Project benefits and/or costs depend on which other projects are implemented
  - Some capacity improvement projects may mostly shift elsewhere the bottlenecks and delays

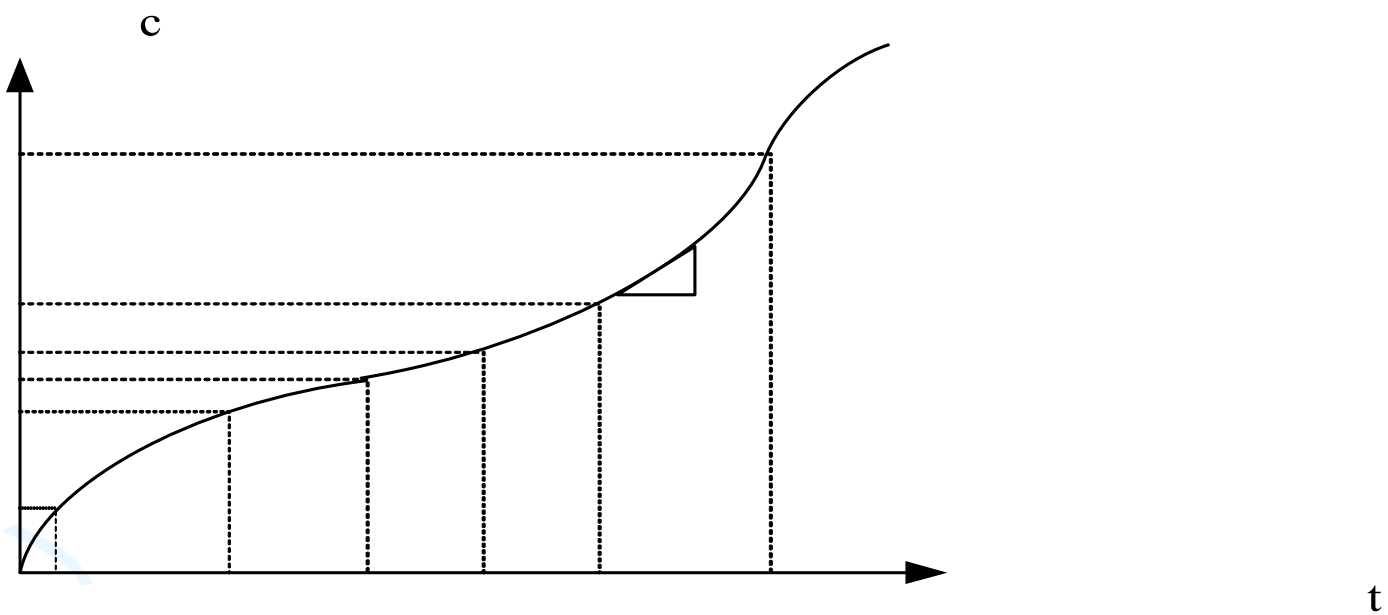
- Methods explored for evaluating schedules of interdependent projects
  - Queuing Metamodels
  - Equilibrium Traffic Assignment
  - Artificial Neural Networks
  - Microscopic Simulation Models

- Methods explored for optimizing the selection, sequencing and scheduling of interdependent projects
  - Swapping Algorithms
  - Branch and Bound
  - Lagrange Relaxation
  - Simulated Annealing
  - Genetic Algorithms

- Relation between project selection, sequencing and scheduling
  - If funds are limited (i.e., insufficient for all worthwhile projects), funds should be used as soon as they become available to complete each project in a sequence as soon as possible.

- Then, a sequence of projects uniquely determines the schedule (i.e., the implementation time of each project).
- Only those projects with implementation times before the end of analysis period are selected. The others are implicitly rejected.

- Project sequence → Project schedule



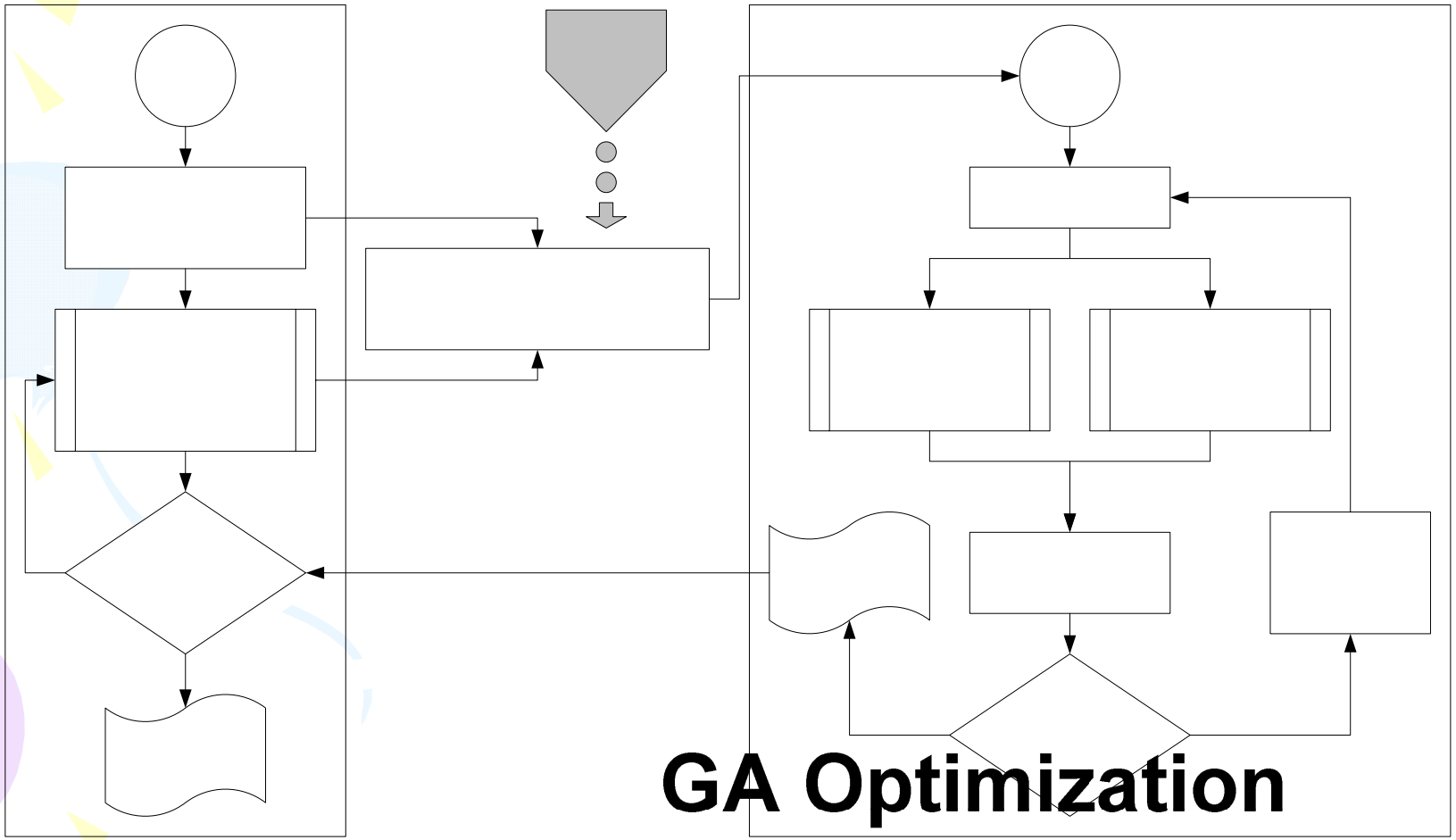
Problem of Scheduling Waterway Improvement Projects

- Combinatorial Optimization Problem
- Decision Variables
  - The sequence of projects and binding budget constraints over time determine the implementation time of each project
  - Projects are implicitly rejected if their implementation times are scheduled beyond the analysis period

- Objective function (evaluated with waterway simulation model), e.g.
  - Minimize the present value of system costs
  - Maximize the present value of net benefits over multi-year analysis period
- Constraints regarding, e.g.
  - Budgets (possibly by region or type of expense), precedence, mutual exclusivity, minimum improvement steps, construction times, capacities, service quality, geographic distributions

- Inputs
  - System network configuration
  - Information on improvement projects
- Outputs
  - Project schedule (implementation timetable of the selected projects)
  - Performance measures

- Integrated waterway simulation and optimization



- SIMOPT
  - Developed at the University of Maryland
  - Serves as prototype for combining and testing simulation and optimization models
  - Composed of
    - Inland Waterway Simulation Model
    - Genetic Optimization Model

- Waterway Simulation Model In SIMOPT
  - Features
    - Data-driven (i.e., network specifiable through data)
    - Demand responses to simulated conditions
    - Lock control alternatives
  - Assumptions
    - Each tow maintains a constant number of barges throughout a trip, even if it must be disassembled while passing through locks. The tow size is determined when the tow is generated.

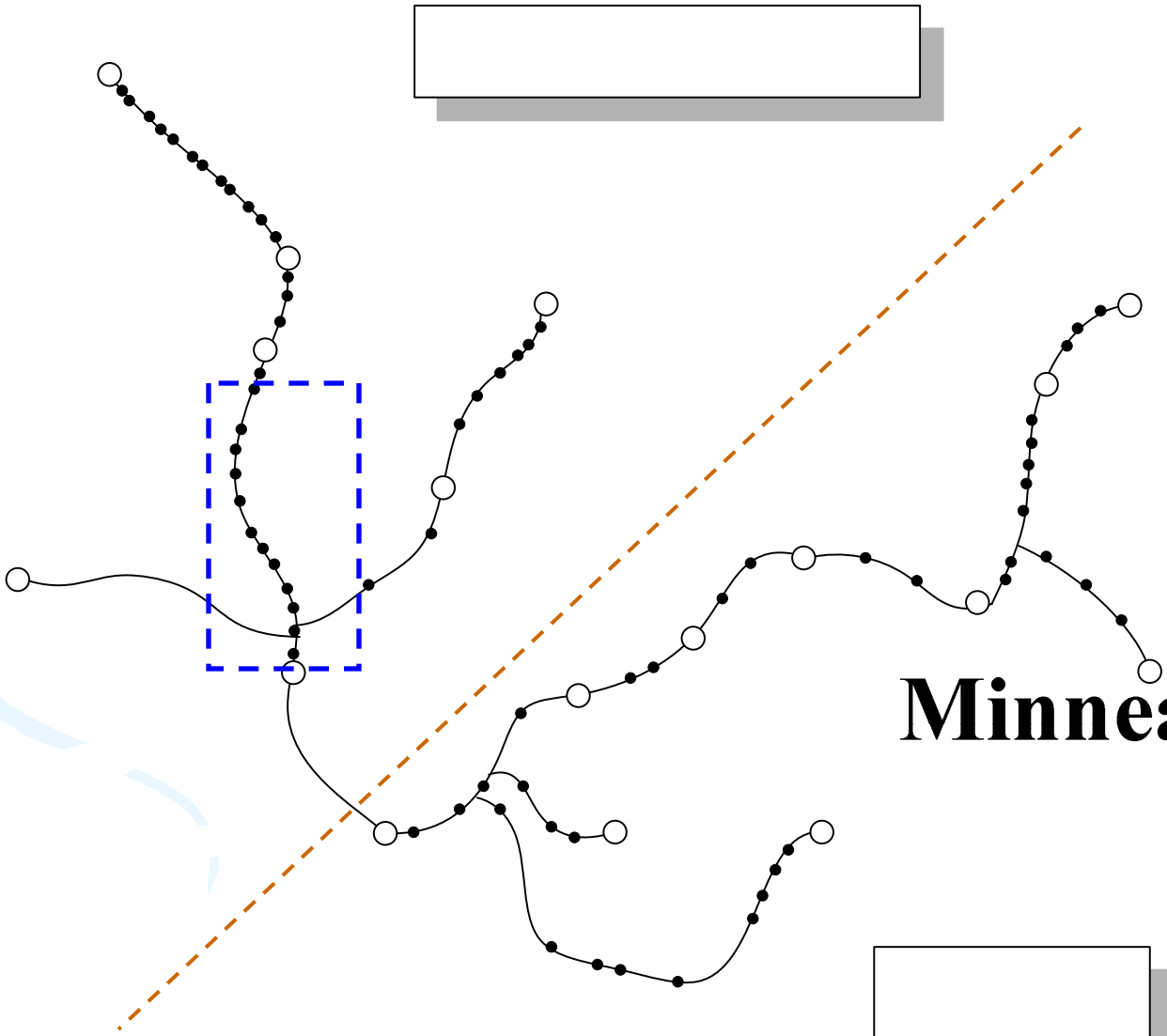
- Tow speed is generated from speed distributions.
- There are no limits on available towboats or barges when tows are generated.
- The queue storage area at each lock is unlimited in both directions.
- All lockage process components are represented with a single service time distribution.

- Genetic Optimization Model In SIMOPT
  - Features
    - Chromosome → Project sequence
    - Genetic operators → IM, VM, EM, PMX, OX, PBX, OBX (assigning mutation and crossover rates)
  - Assumptions
    - The implementation costs of projects are independent and additive. Whenever the cumulative budget reaches the level required to complete an additional project, that project goes into operation

- The budget is accumulated continuously as a function of time over the planning horizon
- The implementation of one project does not depend on the existence of other projects
- Lock capacity is proportional to mean service rate
- Capacity changes do not affect the number of chambers
- Lock capacity increases instantaneously when a lock improvement project is completed.
- Only one improvement project is considered at each lock location.
- Budget constraints are always binding, i.e., never enough for all justifiable projects.

- Test of SIMOPT
  - US Inland Waterway
    - Simulation Network includes
      - Upper Mississippi River
      - Illinois River
      - Missouri River
    - Improvement Projects
      - Locks #15, #16, #17, #18, #19, #20, #21, #22, #24, #25
      - All projects double their lock's capacity
    - Accelerated simulation period (1.5 years) with
      - High traffic growth
      - High budget flows

University of Maryland
Department of Civil and
Environmental Engineering



Minneapolis/St. P

Upper and L

St. Anthony

1 #2 #3 #4 #5 #6

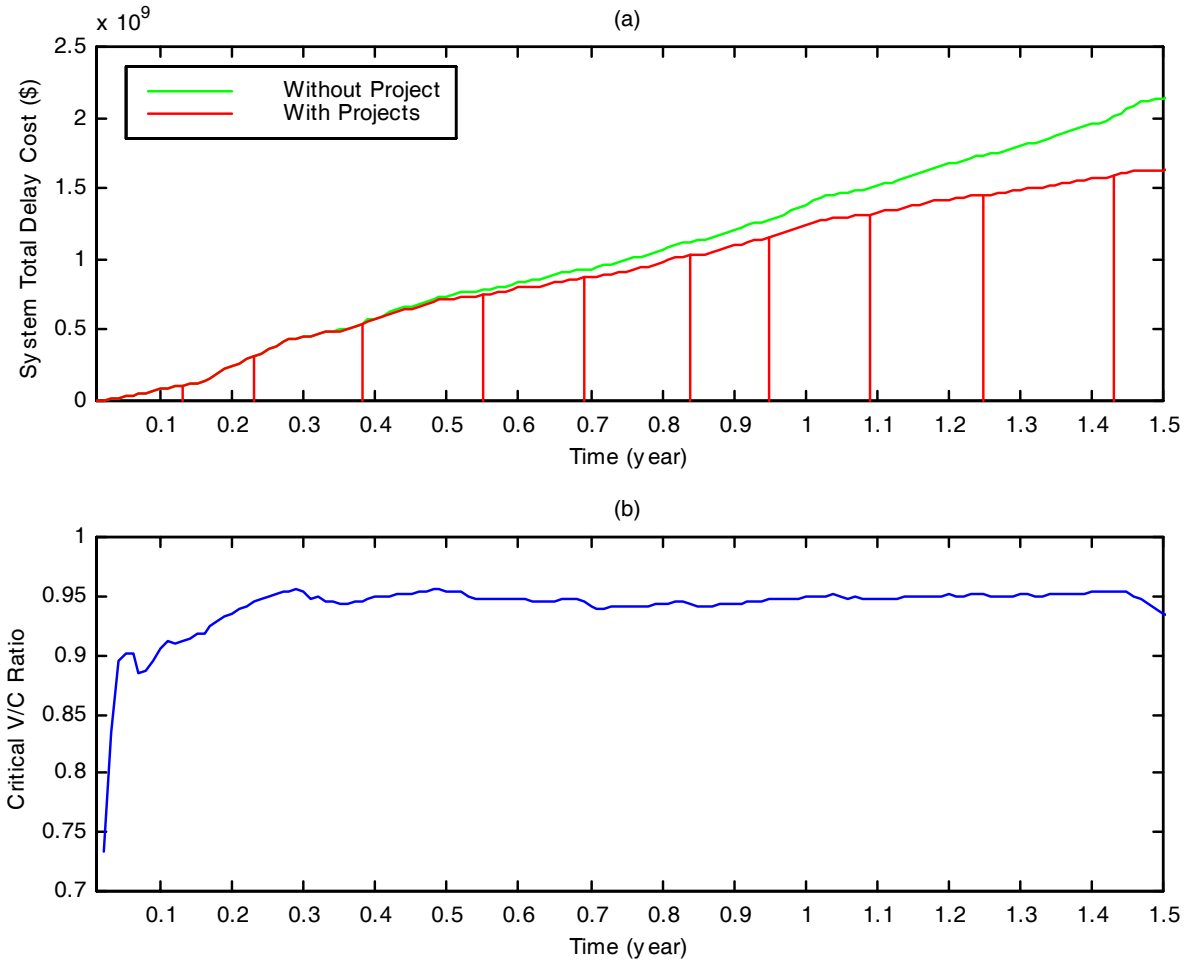
- Simulation Inputs
  - O/D matrix
  - Tow size distribution
  - Speed distribution
  - Service time distribution
  - Number of chambers
  - Control alternatives (F → FCFS, S → SPF)

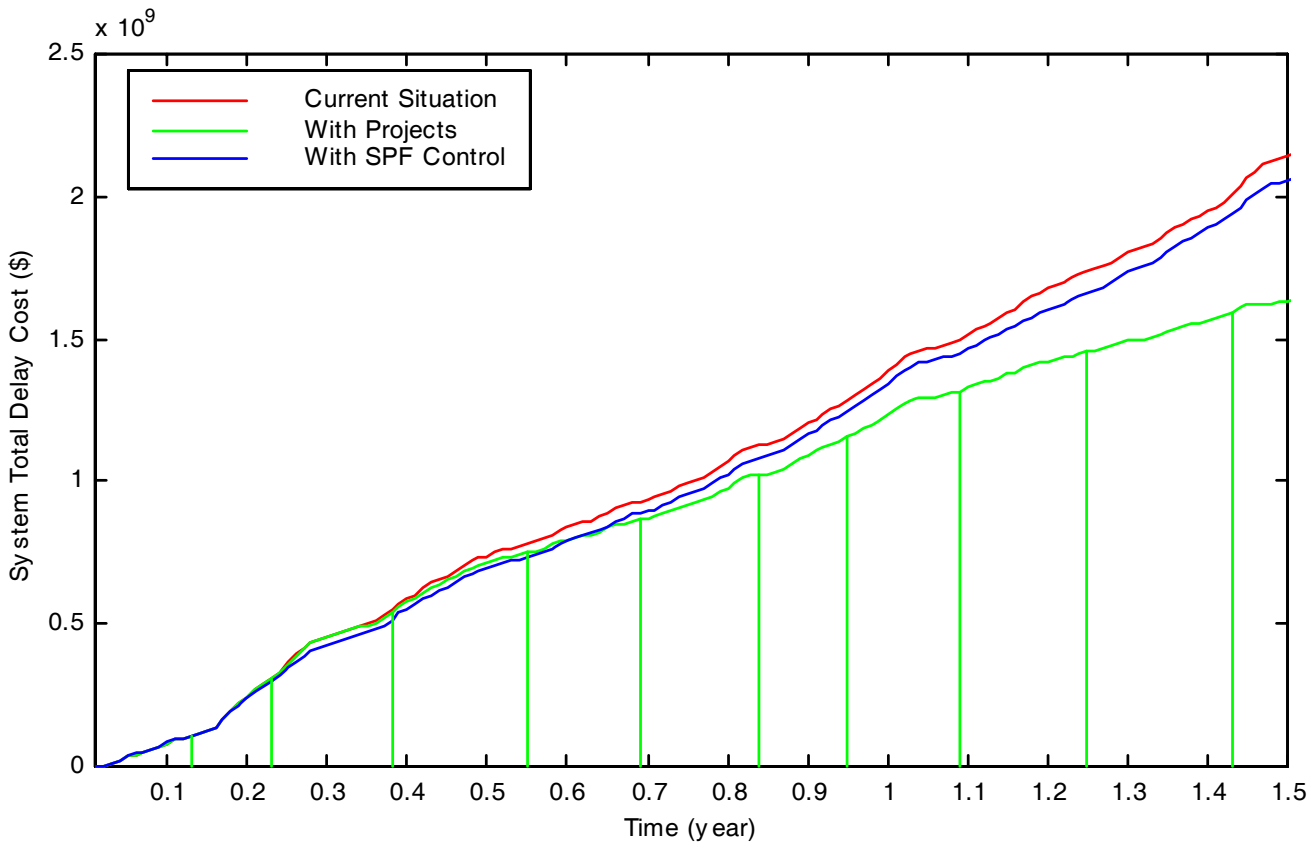
Lock	Control	Lock	Control	Lock	Control	Lock	Control
Upper Falls	F	#8	F	#17	F	#27	F
Lower Falls	F	#9	F	#18	F	LaGrange	F
#1	F	#10	F	#19	F	Peoria	F
#2	F	#11	F	#20	F	Starved Rock	F
#3	F	#12	F	#21	F	Marseilles	F
#4	F	#13	F / S	#22	F / S	Dresden Island	F
#5	F	#14	F	#24	F / S	Brandon Road	F
#6	F	#15	F	#25	F / S	Lockport	F
#7	F	#16	F / S	#26	F	T. J. O'Brien	F

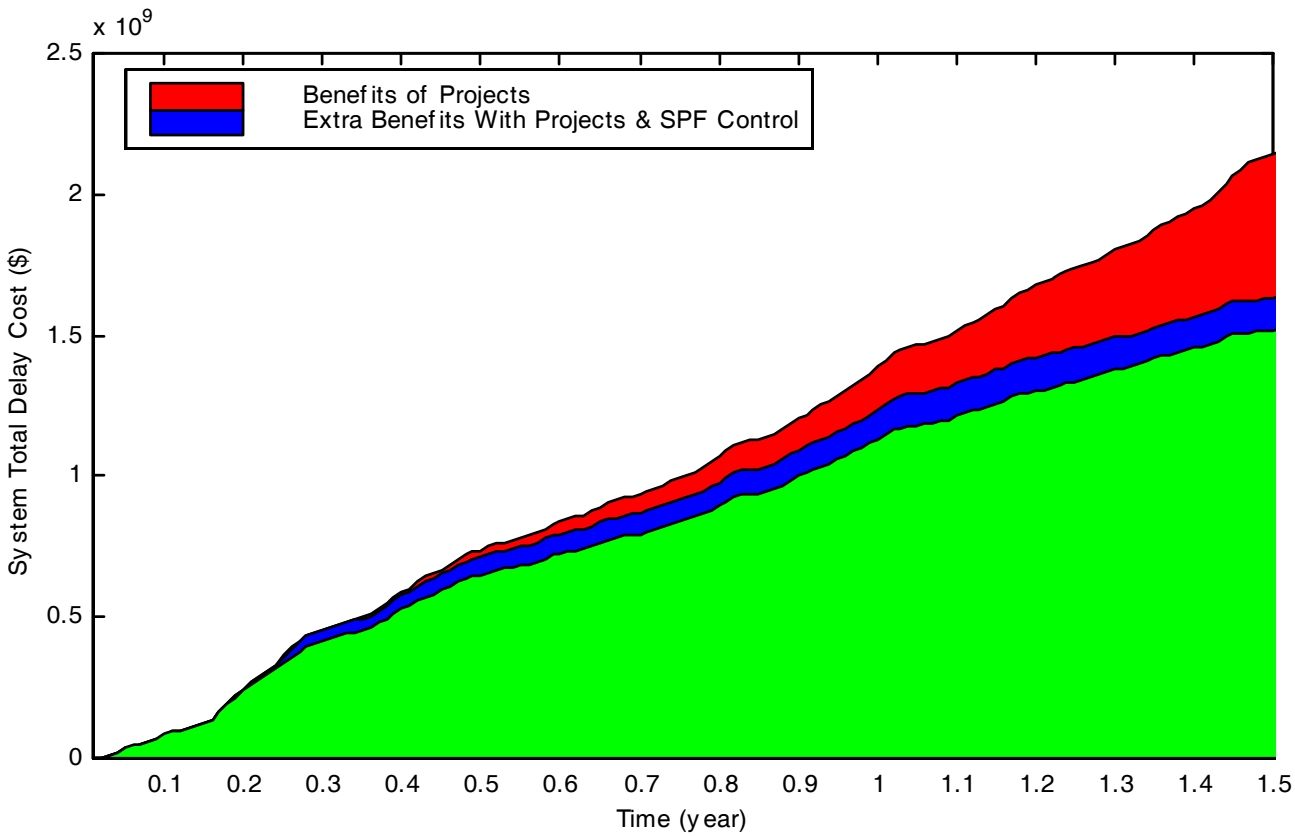
- Optimized Project Sequences

Project Sequence	Network-Wide FCFS		Selected SPF	
	Lock Location	Completion Time (Year)	Lock Location	Completion Time (Year)
1	# 22	0.13	# 13	0.17
2	# 16	0.23	# 16	0.27
3	# 25	0.38	# 18	0.41
4	# 13	0.55	# 19	0.53
5	# 18	0.69	# 17	0.71
6	# 24	0.84	# 20	0.87
7	# 19	0.95	# 22	0.99
8	# 21	1.09	# 25	1.14
9	# 20	1.25	# 21	1.28
10	# 17	1.43	# 24	1.43

Shiaaulir Wang and Paul Schonfeld




x 10^9
2.5
2
1.5
1
0.5
0
System Total Delay Cost ($)
Current Situation
With Projects
With SPF Control
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
Time (year)


x 10^9
2.5
2
1.5
1
0.5
0
System Total Delay Cost ($)
Benefits of Projects
Extra Benefits With Projects & SPF Control
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
Time (year)

- Model Extensions for NaSS
  - In addition to the information about project location, cost, and size, detailed improvement plan may include:
    - Project construction time
    - Capacity reductions during construction
    - Number and size of chambers
    - Maintenance cost
    - Failure rates and durations before and after projects

– Simulation model should be able to

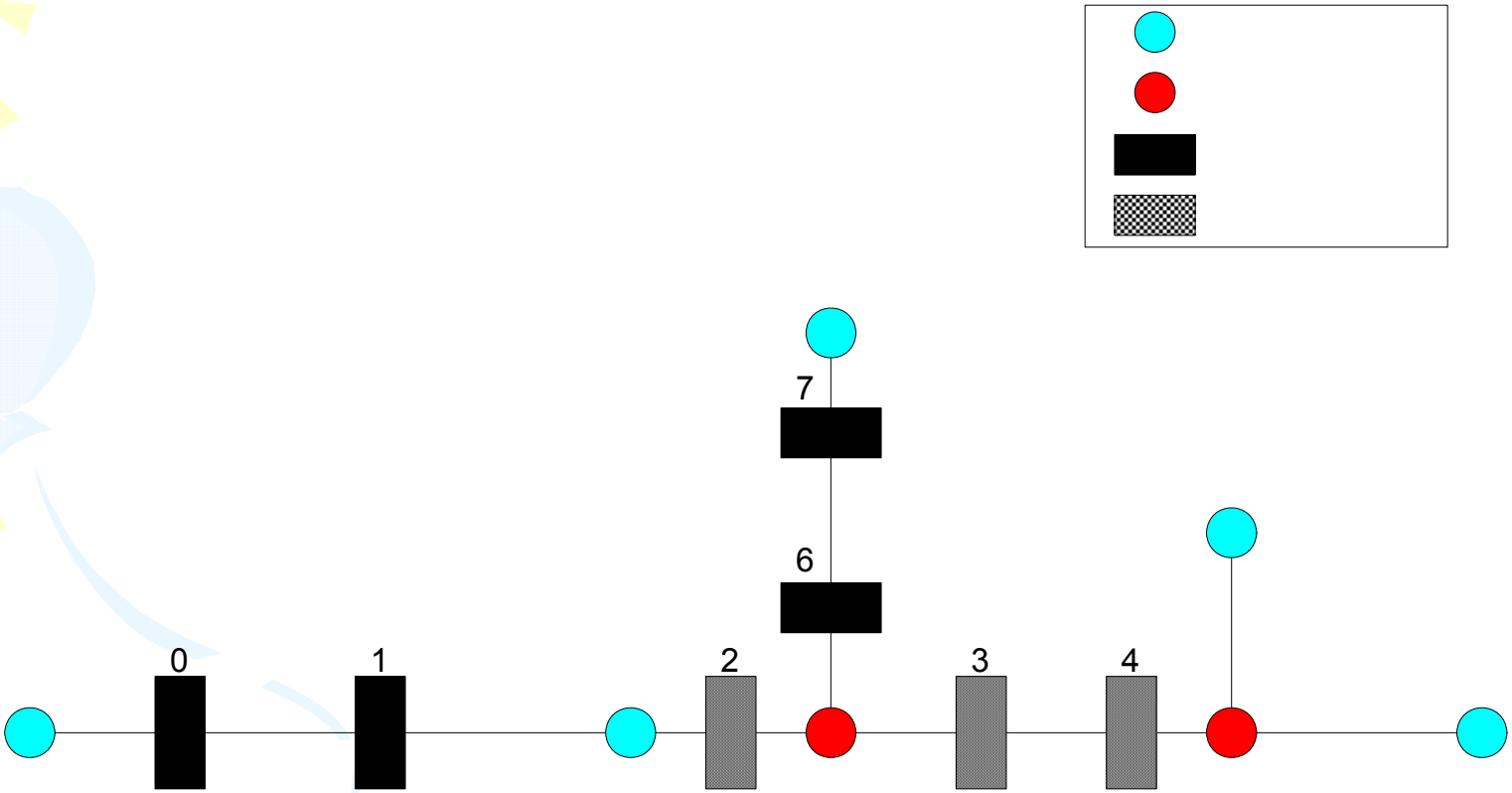
- Consider demand response to network improvements during simulation
- Consider demand diversion due to construction and other service interruptions
- Update system characteristics during the simulation
- Change lockage behavior if a parallel chamber is added
- Change lock control policies as congestion increases

– Refined optimization features:

- Additional constraints (e.g., precedence, mutual exclusivity, available budgets, regional distribution of projects, complementarities among projects)
- Improve search algorithm by creating “smart” operators
- Develop prescreening rules to avoid unpromising solutions
- Avoid re-simulating previous solutions
- Develop parallel processing capabilities

- Further Numerical Examples
  - Optimized project sequences with and without considering construction time and capacity reduction
  - Optimized project sequences with and without considering multiple alternatives at the same locations

- Network Example



- Parameters

Simulation	Time Value	450 $ / barge-minute
	Budget Rate	150×10^6 $ / year
	Demand Growth Rate	2.0% per year
	Replications	10
	Simulation Period	2.5 years
	Warm-Up Period	1 year

Genetic Algorithm	Population Size	50
	Selection Probability	Ranking of fitness value
	Sampling Mechanism	Elitist selection & stochastic sampling
	Mutation Rate	0.07
	Crossover Rate	0.3
	Replacement	Replace worst parents
	Termination	20 generations w/o improvement

- Problem Characteristics
  - Objective function: Minimize Total Cost ($)
  - Not all locks require improvement projects
  - Demand grows over time at specified annual growth rate (2% per year)
  - Total cost includes total delay cost (barge-minutes) and construction cost
  - Project construction starts at the time when required cost is accumulated

- Case1: Construction Time / Capacity Reduction
  - Project Characteristics

Project No.	Lock Location	Capacity Expansion Ratio	Project Cost ($M)
1	7	2	17
2	1	2	16
3	6	2	23
4	0	2	19
5	2	1.1	22
6	4	1.1	21
7	3	1.1	25

Project No.	Lock Location	Capacity Expansion Ratio	Project Cost ($M)	Construction Time (yr)	Residual Capacity Ratio
1	7	2	17	0.17	0.2
2	1	2	16	0.09	0.5
3	6	2	23	0.12	1
4	0	2	19	0.11	0.5
5	2	1.1	22	0.03	0.8
6	4	1.3	21	0.09	0.2
7	3	1.1	25	0.04	0.5

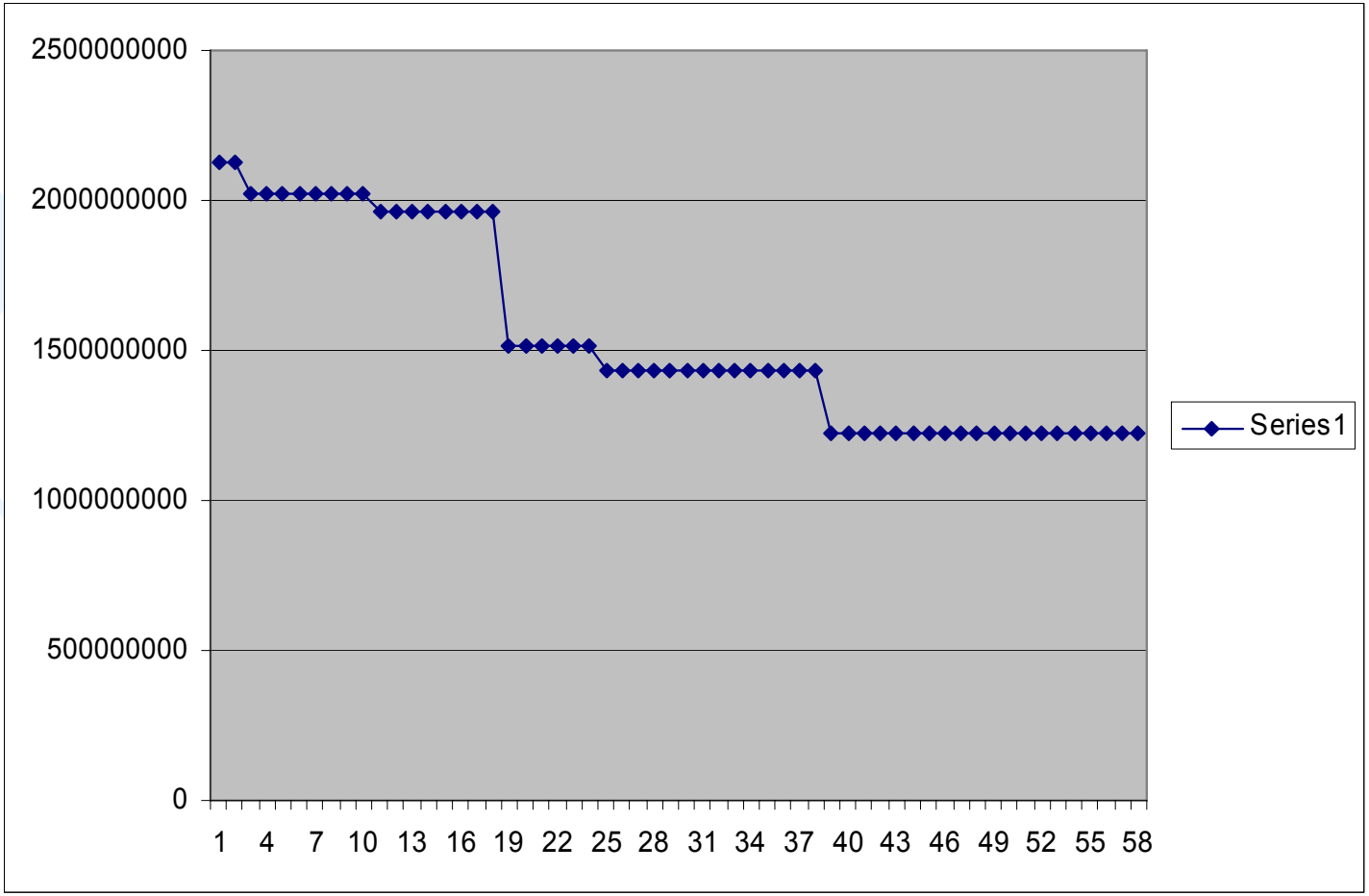
– Problem Characteristics
  - Only one project for each lock
  - Solution space = 7! (~*5,040*)

– Optimized Sequences and Implementation Schedule

Construction Time / Capacity Reduction	Optimized Sequence (Lock Location)	Total Cost
NO	1→0→7→6→2→4→3	319,707,226
YES	1→6→7→2→4→3→0	1,225,828,520

w/o Construction Time and Capacity Reduction					w/ Construction Time and Capacity Reduction			
Project No.	Lock	Time Table (Yr)			Project No.	Lock	Time Table (Yr)	
		Build	Open				Build	Open
2	1	0.11	0.11		2	1	0.11	0.2
4	0	0.23	0.23		3	6	0.26	0.38
1	7	0.35	0.35		1	7	0.37	0.54
3	6	0.5	0.5		5	2	0.52	0.55
5	2	0.65	0.65		6	4	0.66	0.75
6	4	0.79	0.79		7	3	0.83	0.87
7	3	0.95	0.95		4	0	0.95	1.06
Computation time = 10792 sec Number of generations = 21					Computation time = 23158 sec Number of generations = 58			

- GA performance (with construction time & cost)



- Case 2: Mutually Exclusive Projects
  - Project Characteristics

Project No.	Lock Location	Capacity Expansion Ratio	Project Cost ($M)	Construction Time (yr)	Residual Capacity Ratio
1	7	1.5	10	0.1	1
2	7	1.8	13	0.13	0.8
3	7	2	17	0.17	0.2
4	1	1.5	10	0.05	1
5	1	2	16	0.09	0.5
6	6	2	23	0.12	1
7	0	1.5	15	0.1	1
8	0	2	19	0.11	0.5
9	2	1.1	22	0.03	0.8
10	4	1.1	15	0.01	1
11	4	1.2	17	0.05	0.5
12	4	1.3	21	0.09	0.2
13	3	1.1	25	0.04	0.5

– Problem Characteristics

- More than one project at some locks
- Only one project is considered at each lock
- Construction time and capacity reduction is considered
- Solution Space

  7!×3!×2!×2!×3! (~*725,760*) < 13! (~*6,227,020,800*)

– Optimized Sequences

Construction Time / Capacity Reduction	Mutually Exclusive Projects	Optimized Sequence (Lock Location)	Total Cost
YES	NO	1→6→7→2→4→3→0	1,225,828,520
YES	YES	7→0→1→6→4→3→2	344,908,155

w/o Construction Time and Capacity Reduction					w/ Construction Time and Capacity Reduction			
Project No.	Lock	Time Table (Yr)			Project No.	Lock	Time Table (Yr)	
		Build	Open				Build	Open
2	1	0.11	0.2		1	7	0.07	0.17
3	6	0.26	0.38		7	0	0.37	0.47
1	7	0.37	0.54		4	1	0.56	0.61
5	2	0.52	0.55		6	6	0.82	0.94
6	4	0.66	0.75		10	4	0.92	0.93
7	3	0.83	0.87		13	3	1.09	1.13
4	0	0.95	1.06		9	2	1.23	1.26
Computation time = 23158 sec Number of generations = 58					Computation time = 41766 sec Number of generations = 24			

Conclusions

- Such optimization based on simulation is becoming feasible, but computational efficiency is crucial
- The optimization method can and should be separated from the simulation model
- GA's can use "smart" operators that exploit problem structure

- GA's are relatively easy to run on parallel processors
- The GA's developed for network-level optimization seem adaptable for optimizing lock-level enhancement projects