The Navigation Economic Technologies Program

November 7, 2006

NETS

navigation · economics · technologies

Genetic Algorithms for Selecting and Scheduling Waterway Projects

Phase 2 Final Report

US Army Corps of Engineers®

IWR Report 06-NETS-R-10

Navigation Economic Technologies

The purpose of the Navigation Economic Technologies (NETS) research program is to develop a standardized and defensible suite of economic tools for navigation improvement evaluation. NETS addresses specific navigation economic evaluation and modeling issues that have been raised inside and outside the Corps and is responsive to our commitment to develop and use peer-reviewed tools, techniques and procedures as expressed in the Civil Works strategic plan. The new tools and techniques developed by the NETS research program are to be based on 1) reviews of economic theory, 2) current practices across the Corps (and elsewhere), 3) data needs and availability, and 4) peer recommendations.

The NETS research program has two focus points: expansion of the body of knowledge about the economics underlying uses of the waterways; and creation of a toolbox of practical planning models, methods and techniques that can be applied to a variety of situations.

Expanding the Body of Knowledge

NETS will strive to expand the available body of knowledge about core concepts underlying navigation economic models through the development of scientific papers and reports. For example, NETS will explore how the economic benefits of building new navigation projects are affected by market conditions and/or changes in shipper behaviors, particularly decisions to switch to non-water modes of transportation. The results of such studies will help Corps planners determine whether their economic models are based on realistic premises.

Creating a Planning Toolbox

The NETS research program will develop a series of practical tools and techniques that can be used by Corps navigation planners. The centerpiece of these efforts will be a suite of simulation models. The suite will include models for forecasting international and domestic traffic flows and how they may change with project improvements. It will also include a regional traffic routing model that identifies the annual quantities from each origin and the routes used to satisfy the forecasted demand at each destination. Finally, the suite will include a microscopic event model that generates and routes individual shipments through a system from commodity origin to destination to evaluate non-structural and reliability based measures.

This suite of economic models will enable Corps planners across the country to develop consistent, accurate, useful and comparable analyses regarding the likely impact of changes to navigation infrastructure or systems.

NETS research has been accomplished by a team of academicians, contractors and Corps employees in consultation with other Federal agencies, including the US DOT and USDA; and the Corps Planning Centers of Expertise for Inland and Deep Draft Navigation.

For further information on the NETS research program, please contact:

Mr. Keith Hofseth
NETS Technical Director
703-428-6468

Dr. John Singley
NETS Program Manager
703-428-6219

U.S. Department of the Army
Corps of Engineers
Institute for Water Resources
Casey Building, 7701 Telegraph Road
Alexandria, VA 22315-3868

The NETS program was overseen by Mr. Robert Pietrowsky, Director of the Institute for Water Resources.

November 7, 2006

NETS

navigation · economics · technologies

GENETIC ALGORITHMS FOR SELECTING AND SCHEDULING WATERWAY PROJECTS

Phase 2 Final Report

Prepared by:

Shiaaulir Wang
Department of Civil and
Environmental Engineering
University of Maryland, College Park

Paul Schonfeld
Department of Civil and
Environmental Engineering
University of Maryland, College Park

For the:

Institute for Water Resources
U.S. Army Corps of Engineers
Alexandria, Virginia

IWR Report 06-NETS-R-10

www.corpsnets.us

Table of Contents

List of Tables

List of Figures

Abstract

A testbed waterway model (SIMOPT) that combines simulation and optimization has been developed at the University of Maryland. It employs genetic algorithms to solve the problem of evaluating, selecting, sequencing and scheduling waterway improvement projects. Its promising demonstration of simulation-based optimization has been discussed in the previous phase.

The improved optimization model is intended to work with the next generation NaSS waterway simulation model which is being developed under the NETS program of the Corps of Engineers. In this phase some improvements in investment optimization methods are developed and tested on SIMOPT. These include additional constraints on project precedence and regional budgets as well as pre-screening rules to avoid expensive simulation of unpromising or infeasible solutions. ("Solutions" consist here of project implementation schedules.) A simplified evaluator is also proposed as a substitute for microsimulation testing of the optimization algorithms.

Introduction

A problem of great concern to the U. S. Army Corps of Engineers (USACE) is the selection, sequencing and scheduling of the waterway improvement projects, which include chamber construction, expansion, rehabilitation, or maintenance. If numerous projects are considered, a massive combinatorial optimization problem results. This problem is very difficult to solve with conventional optimization approaches. Thus, an investment optimization model based on genetic search algorithms is proposed to solve this large and complex combinatorial problem.

Solving an optimization problem requires evaluation as well as optimization. As a complex and probabilistic system, a waterway network can be analyzed through a detailed simulation model. Thus a simulation-based optimization model is explored for selecting and scheduling waterway projects.

Since the developments of simulation and optimization components are largely separable, SIMOPT can be used to quickly test optimization improvements without running more detailed and longer-running simulations. In the previous phase (Wang and Schonfeld, 2006 NETS Report), project construction time and capacity reductions during construction were considered in the SIMOPT model. Constraints specifying mutually exclusive projects at any locks were also included in the search process. The avoidance of duplicate evaluations during the genetic search is helpful in saving the computation time.

The following sections focus on the enhancements to the optimization model which consider precedence relations and regional budget limitations. Those constraints are also included among prescreening rules designed to avoid simulating infeasible solutions. A procedure for avoiding duplicate simulation runs by prescreening newly generated solutions against the stored previously simulated solutions had already been proposed in the previous phase. The computational advantages of this prescreening process are further explored and assessed here. In order to compare the performance of various mutation and crossover operators, a simplified evaluator is proposed to evaluate the system performance with some of original simulation inputs. Instead of running the simulation models, the proposed evaluator computes the fitness value of any generated project sequences during the genetic search. By substituting for evaluation through simulation, it greatly reduces computation time needed for testing the optimization algorithms.

Improvements to Genetic Algorithms

According to the Scope of Work drafted for GA enhancement (see Appendix), several tasks are included in the current phase, including developing prescreening rules to avoid simulating solutions, considering precedence relations, considering regional budget constraints, and comparing the performance of various mutation and crossover operators. For comparing several GA operators, a simple evaluator is proposed as a substitute for simulation during the testing process.

Precedence Relations

Based on technical, political or geographical considerations, some precedence relations among projects or locations may be imposed on the scheduling process. As in resource-constrained project scheduling problems (RCPS), it may be necessary or preferable to schedule some particular projects ahead of some others. Since construction time may overlap, the precedence relations considered restrict the sequence of project implementation, rather than project completion times. That is, they restrain the project funding sequence rather than completion sequence.

In order to determine the sequence of predecessor/successor projects, precedence constraints define precedence relations among various projects and are represented by an arrow between any two projects with precedence relations. If two projects P_i and P_j are related by a precedence constraint $P_i \rightarrow P_j$, project P_j can only be started when P_i is funded, or later. Given an array of integers $\{x_i\}$ where *i = 1, 2, 3, …, n*, *n* is the number of projects, each element of array represents the scheduled order of one project. The precedence constraint can be formulated as $x_i < x_j$. Similarly, for the case which multiple projects alternatives are considered at one lock location, if projects at two locks L_i and L_j are related by a precedence constraint $L_i \rightarrow L_j$, a project at lock L_i can only be started when a project at lock L_j is funded, or later.

Since precedence constraints define an order of succession among projects, it is important to note that some solutions (i.e., project sequences) would be infeasible and should be prescreened and discarded before being simulated. To impose the precedence constraints, infeasible solutions which violate any one of the precedence relations should be very unlikely to be selected to reproduce offspring in the next generation. Thus, if a sequence violates the precedence constraints, instead of running the simulation to evaluate its performance, its fitness value is assigned a large number (i.e., 10^{15}) which represents the penalty (Tao 2006) in a minimization problem. In a maximization problem, a number close to 0 (i.e., 10^{-15}) is assigned as the fitness value for a sequence violating the precedence constraints. Let a binary variable p_i denote the relevant precedence constraints, $i = 1,2,\ldots, k$, if $p_k = 1$, the k^{th} precedence constraint is satisfied; if $p_k = 0$, the k^{th} precedence constraint is violated. Since k denotes the any given precedence constraint, then the objective function is multiplied by a factor of $\prod_k p_k$. In a minimization problem, when $\prod_k p_k = 0$, the fitness value ends with a large number, i.e., 10^{15}. Otherwise when $\prod_k p_k = 1$, the fitness value is the simulated total system cost.

Regional Budget Constraints

The large U.S. inland navigation system with numerous rivers and branches is operated by different geographic divisions which may have separate budgets. If there are mutually exclusive projects at some locations, although the number of project combinations increases, some combinations are infeasible due to limited regional budgets.

If funds are limited (i.e., always insufficient for all worthwhile projects), funds should be used as soon as they become available to complete as soon as possible each project in a sequence. That is, as funds become available over time, and assuming that funding is never (anytime throughout the simulated analysis period) sufficient to implement all justifiable projects, then, a sequence of projects uniquely determines the schedule (i.e., the implementation time of each project). Thus each project in the sequence is implemented as soon as the funding stream allows it. Hence, with a constrained budget over time, the optimal project sequence uniquely determines the optimal project schedules. Only those projects with implementation times before the end of analysis period are selected. The others are implicitly rejected, thus, determining the project selection.

As shown in Figure 1, for a given project sequence, the time at which each project is finished can be obtained by comparing the cumulative budgets and cumulative project costs. Then let o_i denote the i^{th} project to be implemented in chronological order and t_i^o denote the time at which o_i is finished. Then t_i^o can be determined by solving the equation $\sum_{j=1}^{i} c_j^o = \int_0^{t_i^o} b(t)dt$, where c_j^o is the capital cost of the j^{th} project to be implemented.

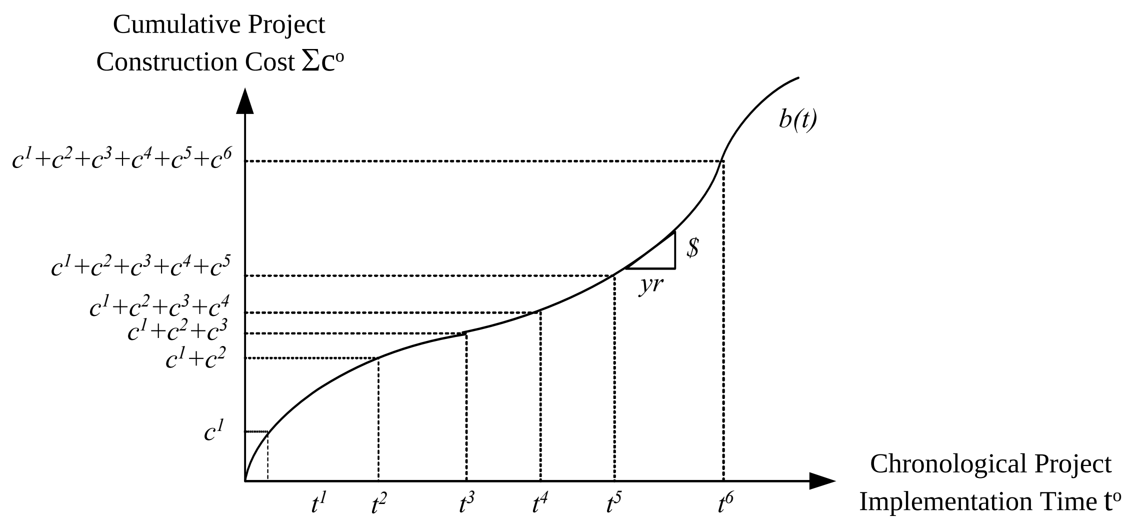


Figure 1 Relations of Budget Flow, Cumulative Cost, Project Sequence, and Project Schedule

The relation and formulation of regional budget constraints is similar to Figure 1 and equation for regional budget constraints is:

$$\sum_{j=1}^{i}\left(c_j^o\right)_k = \int_0^{t_i^o} b_k(t)dt$$

where $\left(c_j^o\right)_k$ is the capital cost of the j^{th} project to be implemented, and $b_k(t)$ is the annual budget in region *k*. If regional budgets are independent, the budget constraint for the problem is easily divided into several regional budget constraints. That is, projects are funded one by one in each region. Funds in one region cannot be used in other regions. Therefore, it is possible that projects from different regions are funded at the same time. The overall implementation sequence is then composed of the implementation sequences in each region.

As shown in Figure 2, a sequence containing only one project at each lock location is refined from the constraint of mutually exclusive projects. If there are different regions and each region has its own improvement projects and budget constraints, three regional implementation sequences are then generated with their own implementation schedules. With regional budget constraints, an overall implementation sequence to be evaluated by the simulation model is rearranged chronologically.

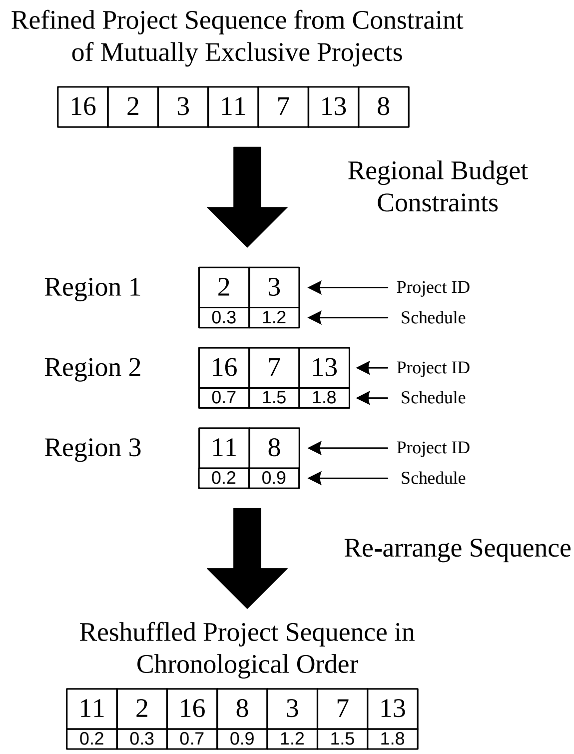


Figure 2 Implementation Sequence with Regional Budget Constraints

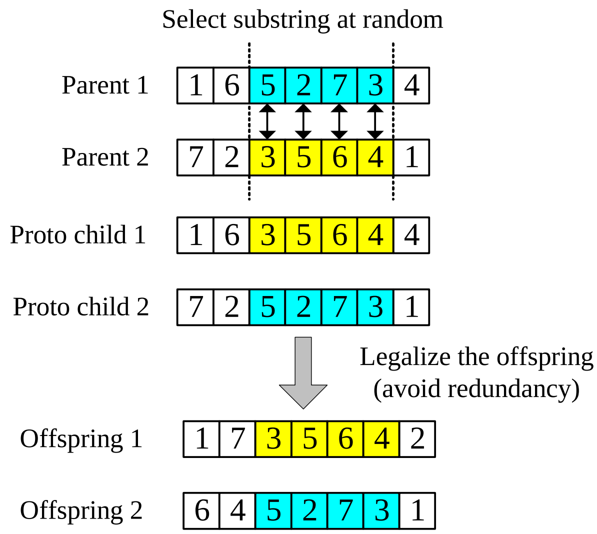
Performance of Genetic Operators

In general, there are two types of genetic operators: mutation operators and crossover operators. Currently, seven operators are equally used often in SIMOPT: partial-mapped crossover (PMX), order crossover (OX), position-based crossover (PBX), order-based crossover (OBX), insertion mutation (IM), exchange mutation (EX), and inversion mutation (VM). Those operators are used to provide population diversities in each generation. Since the mutation/crossover rates and the selection and combination of operators do affect the GA search performance, a question arises about how the population diversity helps locate the optimized solution.

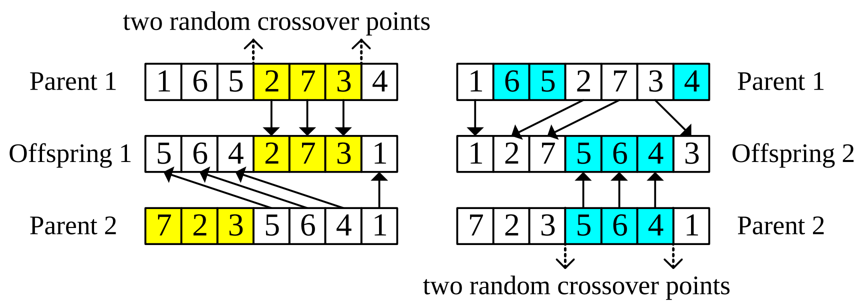
Those seven GA operators are presented below:

Crossover Operators

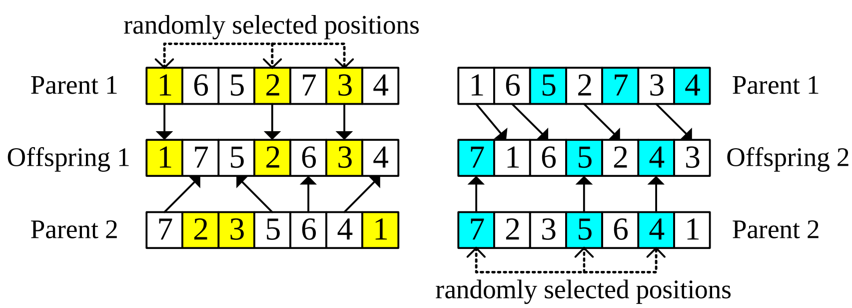
- Partial-Mapped Crossover (PMX)



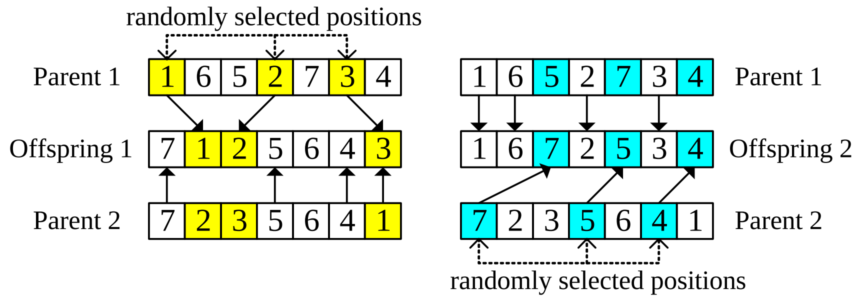
- Order Crossover (OX)



- Position-Based Crossover (PBX)

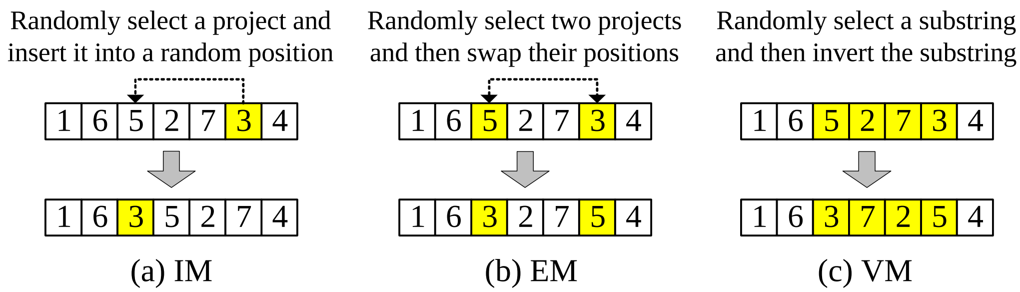


- Order-Based Crossover (OBX)



Mutation Operators

- Insertion Mutation (IM)
- Exchange Mutation (EM)
- Inversion Mutation (VM)



Simplified Evaluator

A detailed waterway simulation model can evaluate the system performance by considering trip variations in demand, type, size and commodity, complex lock features and locking operations, and effects of system closures or project improvements. As discussed above, optimization based on evaluating objective functions with simulation is becoming feasible, but the computation time is a crucial factor. Since the optimization method can be fully separated from the simulation model, the development efforts for these two processes can proceed concurrently. A simplified evaluator is used to approximately estimate the system performance by bypassing some of the detailed factors (as shown in Figure 3). While developing the optimization model, it is faster to use a simple algebraic evaluator rather than a long-running simulation model. It substitutes for the simulation model only while the optimization methods are tested.

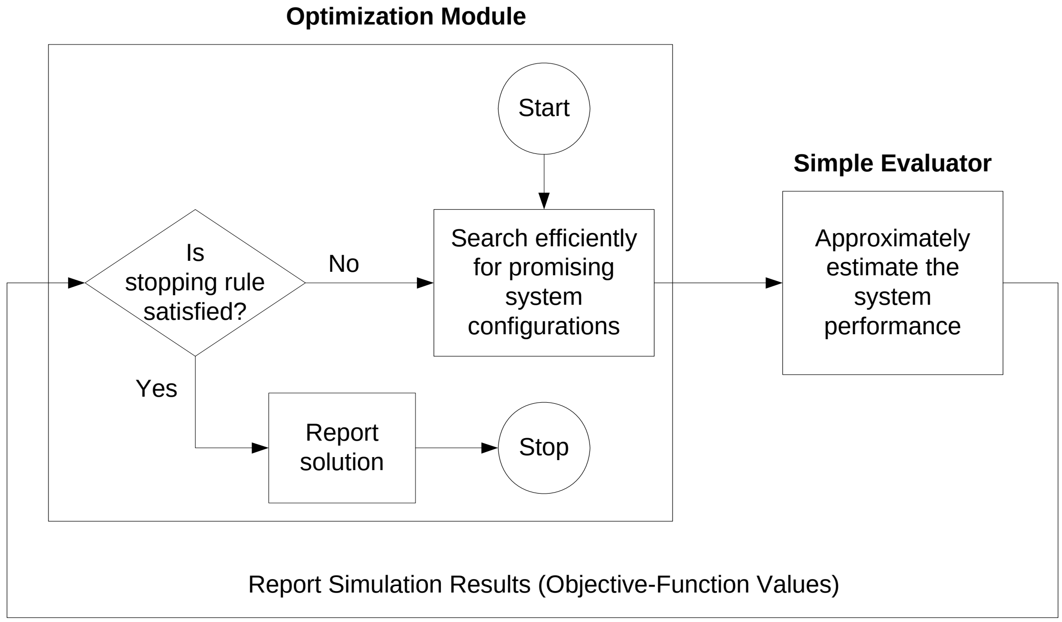


Figure 3 Optimization Model with Simple Evaluation

The simple evaluator proposed here is expected to temporarily replace the waterway simulation model but still use some similar inputs. The simulation model used within SIMOPT can be applied generally to different waterway networks. Therefore, it is desirable that the proposed evaluator can also work on different waterway networks.

In using simple queuing theory to estimate the system performance, i.e., lock delays, the required parameters used in the simple evaluator should include traffic volumes at locks and lock capacities. In considering lock interdependence, we know that as the link distance between locks increases, the lock interdependence decreases. Hence, the distances between locks should be considered in the evaluator.

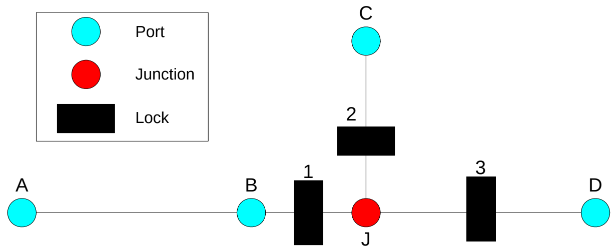
Model Parameters

Three basic parameters are required to estimate the network system performance: traffic volumes, lock capacities and link distances. It is preferable to extract those data from inputs of the simulation model.

Since the developed simulation model is generally applicable to different networks, network configuration data provide information on network characteristics, including the locks, ports, links, junctions and their relative relation. Lock capacity is determined by the lock service time. Service time distributions are based on the lock and chamber characteristics, number, and movement directions. Network statistical data used in the simulation model provide such data on service times.

Traffic volume through any lock can be simply determined from the O/D (origin/destination) matrix as long there are no alternate paths through the network.

Figure 4 (a) shows an example of waterway network and its O/D flow volume matrix. Figure 4 (b) shows the O/C paths composed by directional links.



	A	B	C	D
A		2	3	1
B	2		1	2
C	3	1		3
D	1	2	3	

(a)

Trips	**Links**							
	A↔B		**B↔J**		**J↔C**		**J↔D**	
	A→B	**B→A**	**B→J**	**J→B**	**J→C**	**C→J**	**J→D**	**D→J**
A→B	2							
A→C	3		3		3			
A→D	1		1				1	
B→A		2						
B→C			1		1			
B→D			2				2	
C→A		3		3		3		
C→B				1		1		
C→D						3	3	
D→A		1		1				1
D→B				2				2
D→C					3			3

(b)

Figure 4 Example of O/D Matrix and O/D Path

As locks are the "stations" on the links, the traffic volumes at locks are the summations of link volumes from different O/D paths (as shown in Table 1). Since two-way traffic shares the same chambers at locks, total volume at locks is the summation of volumes from two directions, upstream and downstream.

Table 1 Calculation of Traffic Volumes at Locks

	$\sum$(A—B)$_{\text{volume}}$		**$\sum$(B—J)$_{\text{volume}}$**		**$\sum$(J—C)$_{\text{volume}}$**		**$\sum$(J—D)$_{\text{volume}}$**	
	upstr.	dnstr.	upstr.	dnstr.	upstr.	dnstr.	upstr.	dnstr.
Lock#1	-	-	7	7	-	-	-	-
Lock#2	-	-	-	-	7	7	-	-
Lock#3	-	-	-	-	-	-	6	6

Delay Estimation with Lock Improvement Projects

Due to the time required in applying simulation-based genetic search, the sensitivity analysis for problem size vs. GA search time is difficult with our current restricted computation resources. However, for exploring the efficiency of the algorithm used in solving the project scheduling problem, an approximate analytical model may be substituted for the simulation model.

The simplified evaluator is based on simple queuing theory. If each lock is assumed to have Poisson arrivals (with arrival rate $\boldsymbol{\lambda}$) and exponentially distributed service times (with service rate $\boldsymbol{\mu}$), each lock can be independently analyzed as an *M/M/1* queuing system with its average tow delay $\boldsymbol{d}$ estimated as:

$$d = \frac{1}{\mu - \lambda} - \frac{1}{\mu}$$

An *M/D/1* queuing system, a simple case of an *M/G/1* queuing system, has Poisson arrivals, constant service time and a single server. Its average delay per tow is:

$$d = \frac{\rho}{2\mu(1-\rho)} - \frac{1}{\mu}$$

where $$\rho = \lambda / \mu$$

In the current SIMOPT model, service time is estimated with regression models as a function of tow size and number of lockage cuts. Thus an *M/D/1* queue can be appropriately applied. For lock series, if locks are far apart, the delay estimates at locks are additive. For two-way balanced traffic, the analytical solution for the average time spent by a tow in this system with *n* locks can be derived as

$$D = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} \frac{\rho_i}{2 \cdot \mu_i (1 - \rho_i)}$$

where

D	average time in the system (hours/tow)
n	number of locks in series
d_i	average time at lock i (hours/tow)
ρ_i	$\mu_i / 2 \cdot \lambda_i$
λ_i	arrival rate for each direction at lock i (tows/hour)
μ_i	service rate at lock i (tows/hour)

However, unless link distances between locks are quite long, lock delays are not independent, i.e. do not add up in a simple way. In order to factor in the interdependence, the average time spent at lock *i* is adjusted with a simple interdependency factor which is related to the average time spent at the previous locks *i-1* (upstream lock) and *i+1*(downstream lock):

$$d_i^* = d_i - \frac{\sqrt{d_i \cdot d_{i-1}}}{l_{i,i-1}} - \frac{\sqrt{d_i \cdot d_{i+1}}}{l_{i,i+1}}$$

where

d_i	average time at lock *i*
d_i^*	average time at lock *i* if lock interdependence exists
l	link length between two adjacent locks

If locks are close to the junction node in a tree network, the adjustment of lock interdependence is factored by *m* and *n* adjacent locks from upstream and downstream of the branches:

$$d_i^* = d_i - \sum_m \left(\frac{\sqrt{d_i \cdot (d_{i-1})_m}}{(l_{i,i-1})_m} \right) - \sum_n \left(\frac{\sqrt{d_i \cdot (d_{i+1})_n}}{(l_{i,i+1})_n} \right)$$

As shown in Figure 5, if expansion projects are considered at locks, the service rate at lock *i* will increase from μ_i to μ_i' at its scheduled completion time. Given any project sequence, the implementation schedule is determined from the project construction cost and available budget. Then, according to the chronological order, at time T_j, there is one and only one expansion project at lock *i*, where *i* is the order of lock series ($i = 1, 2, \ldots, p$) and *j* is the order of project sequence ($j = 1, 2, \ldots, q$). After time T_j, the new service rate μ_i' is applied immediately at lock *i* and will maintain the same value of μ_i' until the end of the planning period.

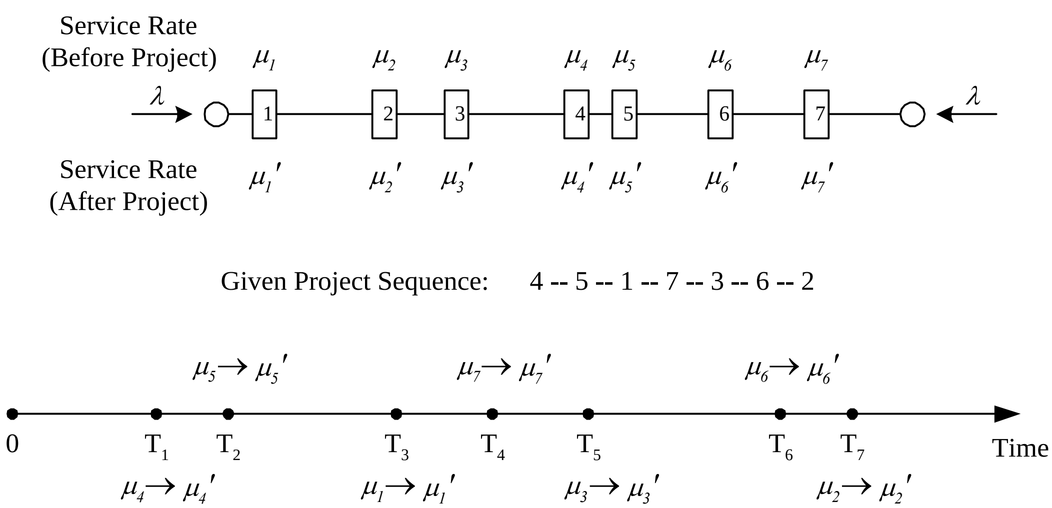


Figure 5 Project Implementation Schedule

Thus, assuming that during construction the new project does not reduce the old project's capacity, with the specified project schedule T_j, $j = 1, 2, \ldots, n$ along the time axis, the total time in the system for all generated tows would be estimated as follows:

$$W = \sum_{j=1}^{q} 2\lambda_i \cdot (T_j - T_{j-1}) \cdot \left(\sum_{i=1}^{p} \left(d_i(t) - \sum_m \frac{\sqrt{d_i(t) \cdot (d_{i-1}(t))_m}}{(l_{i,i-1})_m} - \sum_n \frac{\sqrt{d_i(t) \cdot (d_{i+1}(t))_n}}{(l_{i,i+1})_n} \right) \right)$$

where

W	total time in system (hours)

T_j	the j^{th} timing position in project schedule
λ_i	arrival rate for each direction (tows/hour)
$d_i(t)$	if lock i is expanded with the j^{th} priority (tows/hour)

$$\begin{cases} d_i, & 0 \le t < Tj \\ d_i', & t \ge Tj \end{cases}$$

Model Test (Enhanced SIMOPT)

Test Network

A simple test network is used here for testing proposed simulation-based optimization model (as shown in Figure 6). There are 3 rivers, 5 ports, and 7 locks (4 single-chamber locks and 3 double-chamber locks). Locks are numbered with ID 0, 1, 2, 3, 4, 6, 7. Lock #5 and #8 are dummy locks (network configuration is from Wang, 2002). Not all locks require improvement projects, but all improvement projects are located at real locks. The lock congestion level from baseline simulation is 7→1→6→0→2→4→3, which is ranked from the highest V/C (volume capacity ratio) to lowest V/C.

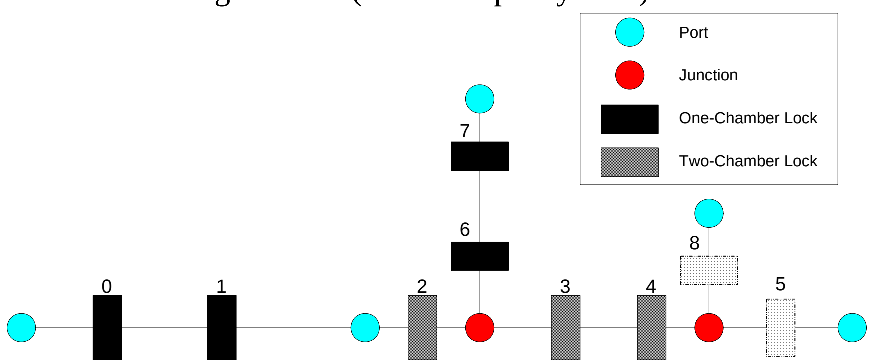


Figure 6 Test Network for SIMOPT Extension

Model Inputs

Simulation inputs include network statistics (O/D trip generation rates, tow size distributions, chamber service time distributions and speed distributions), lock operation (FIFO control, towboats priority, lockage cuts, chamber assignment and chamber bias), demand variables (baseline O/D travel time, annual growth rates), and system variables (simulation period, warm-up period, number of replications) (Wang, 2005). The project relevant inputs include budget rate, project IDs, locations, costs, capacity expansion ratios, regional budget, and precedence relations (as shown in Table 2). The precedence constraints limit the sequence of locks receiving improvement projects. For example, the alternative projects at lock #6 should be funded before the alternative projects at lock #2 and #3. The regional budget constraints restrain the project fund in each region: $\$40 \times 10^6$, $\$70 \times 10^6$, $\$40 \times 10^6$ annually for regions 1, 2, and 3, respectively. For example, the alternative projects at lock #7, #2, and #6 are funded annually by the 2nd regional budget, 70 $\times 10^6$, which is uniformly distributed within each year. The termination rule for GA search is set at 20 generations without further improvement. Mutation and crossover rates are 0.07 and 0.3, respectively. All the tests are run on a Pentium III processor with 3.6 GHz CPU and 1GB memory.

Table 2 Project Information

Project ID	Lock ID	Region Code	Capacity Expansion Ratio	Cost ($\times 10^6$)	Precedence Relations
1	7	2	1.2	17	
2	7	2	1.5	20	
3	7	2	1.8	23	
4	7	2	2.0	27	
5	1	1	1.2	16	
6	1	1	1.5	20	
7	1	1	2.0	26	
8	6	2	1.5	27	6→3, 6→2
9	6	2	2.0	33	
10	0	1	1.2	20	
11	0	1	1.5	12	
12	0	1	2.0	29	
13	2	2	1.1	32	6→2
14	2	2	1.2	35	
15	4	3	1.1	25	
16	4	3	1.2	27	
17	4	3	1.3	31	
18	3	3	1.1	35	6→3

In this test, multiple projects are considered at some lock locations. However, at most one of the alternative projects for each location will be selected in any implementation sequence. There are 18 projects: 4 alternatives at lock #7, 3 alternatives at lock #1, 2 alternatives at lock #6, 3 alternatives at lock #0, 2 alternatives at lock #2, 3 alternatives at lock #4, and 1 alternative at lock #3. With these mutual exclusivity constraints for projects at some locks, the solution space is $7! \times (4 \times 3 \times 2 \times 3 \times 2 \times 3 \times 1) = 2{,}177{,}280$. That is much less than $18! = 6{,}402{,}373{,}705{,}728{,}000$. Additionally, with 2 precedence constraints, the solution space is further reduced to $(C_4^7 \times 4! \times 2!) \times (4 \times 3 \times 2 \times 3 \times 2 \times 3 \times 1) = 725{,}760$.

In order to accelerate the analysis, a high budget flow is assessed. 10 replications are required to complete one simulation evaluation of any candidate solution (i.e., generated project sequence and resulting schedule). The population size in this test is set at 20. An interest rate of 4% is used to compute the discounted present value of cost, assuming that the average time value is $450/tow-hour. In the evolution process, if the generated sequence violates the constraint, its fitness value is assigned a large cost and has nearly no chance being selected as a rents for next generation.

Test Results

Adding Constraints (Precedence Relations and Regional Budget Limitations)

Since the proposed optimization search is probabilistic and requires random numbers in the evolution process, 20 GA search processes for the same problem but with different random seeds are presented in Table 3. Based on 20 search processes, the results show that most searches (15 out of 20 replications) converge to the similar optimized solutions, with optimal total cost of $594,187,215, when 20 unchanged solutions are found in search process. Searches #2 and #9 even converge to lower optimized solutions ($593,965,950). The best solution in most searches has project sequence of 2→6→8→16→11→13→18. On average, approximately 566 solutions are generated in each of the 20 search processes, of which 447 are new. (The other 135 are screened out if they exactly match previously generated solutions.) Among the 447 new solutions 23 (=5%) are infeasible her (i.e., they violate some constraints) and are screened out before being simulated. Thus, on average only 424 solutions are simulated for this problem before an optimized solution is identified and the search stops.

Table 3 Optimized Results

GA Search	# of Gen.	# of Generated Sequences	# of Infeasible Solutions / # of New Solutions (% of Infeasible Solutions)	Optimal Total Cost ($)
1	44	659	25 / 524 (4.8)	594,187,215
2	34	529	19 / 417 (4.6)	593,965,950
3	25	384	12 / 313 (3.8)	594,187,215
4	47	673	37 / 543 (6.8)	594,187,215
5	48	728	21 / 525 (4.0)	594,187,215
6	41	626	28 / 492 (5.7)	594,187,215
7	38	598	8 / 471 (1.7)	594,187,215
8	33	501	39 / 410 (9.5)	601,896,915
9	27	440	14 / 368 (3.8)	593,965,950
10	32	487	16 / 388 (4.1)	594,187,215
11	33	502	25 / 411 (6.1)	594,187,215
12	31	481	17 / 391 (4.3)	594,187,215
13	40	586	28 / 445 (6.3)	594,187,215
14	29	456	19 / 356 (5.3)	601,550,790
15	29	448	22 / 365 (6.0)	594,187,215
16	29	466	20 / 363 (5.5)	594,187,215
17	39	576	21 / 465 (4.5)	601,833,750
18	41	625	22 / 484 (4.5)	594,187,215
19	47	730	28 / 580 (4.8)	594,187,215
20	55	829	41 / 637 (6.4)	594,187,215
Avg.	**37**	**566**	**23 / 447 (5.1)**	**-**

In such a complex combinatorial problem, it is difficult to find the exact globally optimal solution. No existing methods can guarantee finding that global optimum when the problem is realistically large. An experiment is designed to evaluate 25,000 randomly generated solutions to the problem with a sampling process.

The solution space for this test case contains $7! \times (4 \times 3 \times 2 \times 3 \times 2 \times 3 \times 1) = 2{,}177{,}280$ solutions. 25,000 solutions cover approximately 1.1% of the solution space. 12,454 out of our 25,000 randomly generated solutions are infeasible ones which violate precedence constraints. Among the 12,546 feasible solutions, the best fitness value in this sample is 593,965,950, while the worst one is 769,730,300. The sample mean is 678,628,766 and the standard deviation is 31,650,572.

Since the sample is randomly generated, the fitted distribution should approximate the actual distribution of fitness values for all possible solutions in the search space. Excluding the infeasible solutions, the distribution for those remaining 12,546 sampled solutions is shown in Figure 7. Based on the plotted histograms, the best fitting distributions for a somewhat uneven bell shape might be the normal or the lognormal distributions. Figure 7 shows those 12,546 sample solutions fitted with normal distribution, normal (μ, σ^2), in which μ and σ^2 are sample mean variance. The values of μ and σ for fitted normal distribution are 678,629,000 and 31,650,600, respectively.

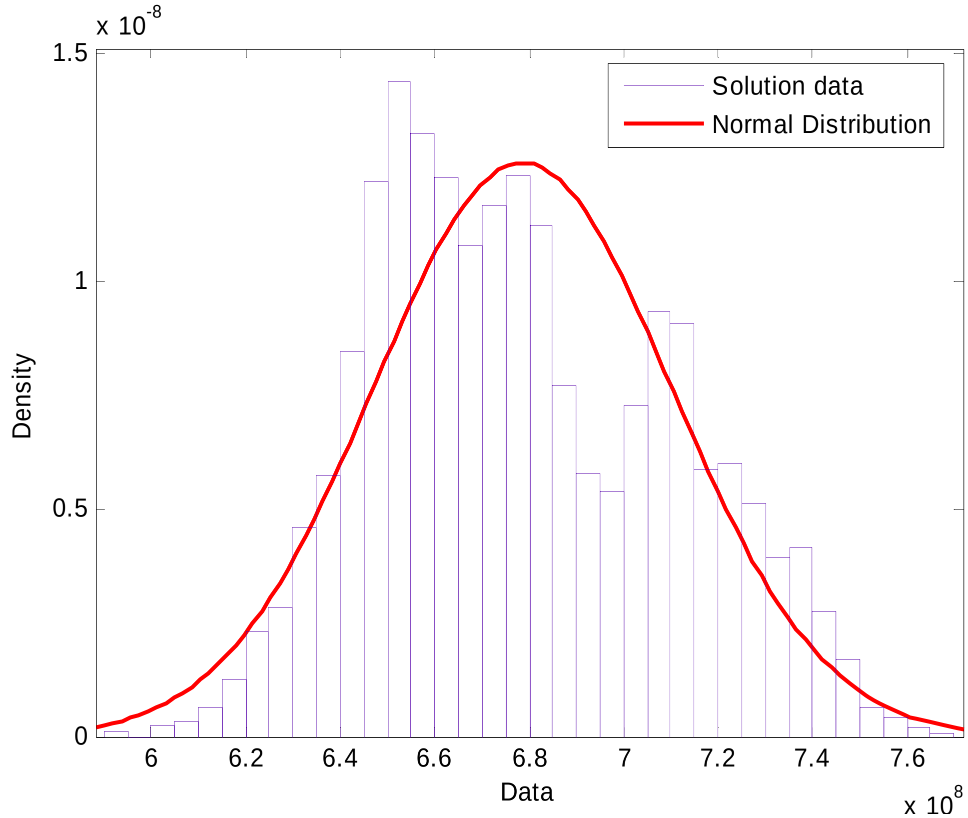


Figure 7 Fitted Normal Distributions

Among the randomly generated solutions, only one among 25,000 (.004%) has a fitness value of 593,965,950 and six among 25,000 (.024%) have a fitness value of 594,187,215, which are the values found in two and fifteen, respectively, of the twenty GA searches. These optimized solutions are located at the extreme low end of the distribution. We can fit various standard distributions (such as the normal distribution shown in Fig. 7) to the actually observed results, but all of them overestimate the probability of finding better solutions, since they have long tails on their left side implying at least some minute probability of finding solutions with very low or even negative costs. Thus, very conservatively, based on the fitted normal distribution with a mean of 678,629,000 and standard deviation of 31,650,600, we would estimate that the probability of finding a solution better than 593,965,950 at .0035 and better than 571,019,000 at .0003. In practice further investigation of the finite lower bounds of the distribution of random solutions, may confirm that no solutions significantly better than 593,965,950 can be obtained for this problem, It should also be remembered that estimation errors in the input information regarding demand, project costs, lock reliability and tow characteristics, limit the usefulness of further searching for mathematical solutions that are globally optimal but limited in accuracy by the input data.

Hence, the solutions optimized through GA searches, although not necessarily globally optimal, are still extremely good when compared with other random solutions in the solution space and leave only a very small probability that significant improvements might still be found by letting the GA search run further. That practically shows the reliability and validity of the proposed search algorithm.

Compared with random search, which might search though most of the solution space to find an optimized solution, the above GA search only takes only 2% as much effort to locate a near-optimal solution. That is, with random search, the possibility of finding such a well-optimized solution in 500 evaluations is 0.02. We would expect that such relative advantage of genetic search over a random search would increase as the problem gets larger (i.e., as the number of project permutations considered increases).

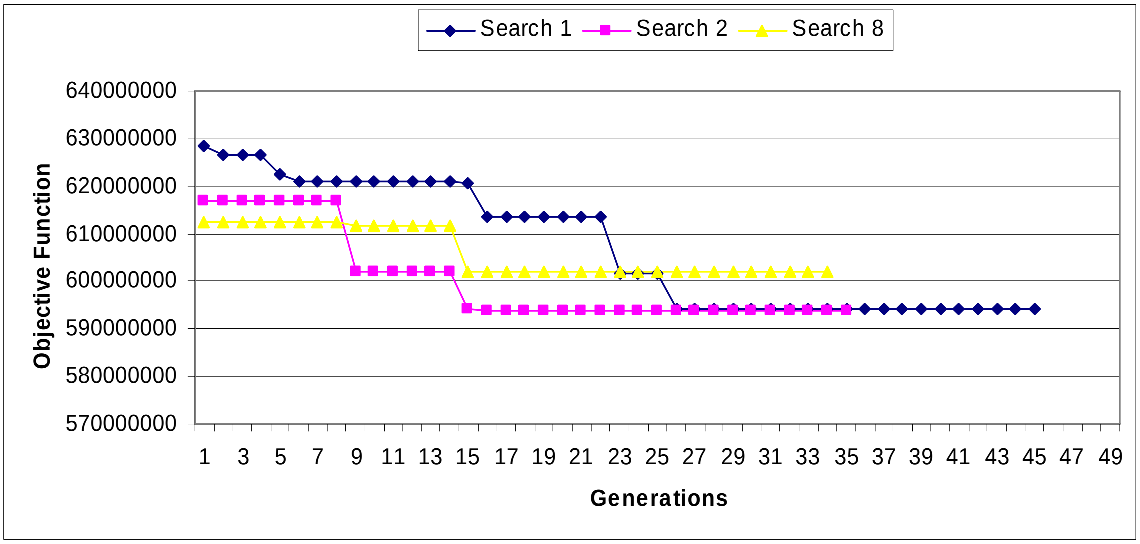
Since all the evaluated solutions are recorded in order to avoid duplicate simulation runs, the number of generated sequences is always larger than the number of evaluated solutions. With our prescreening procedure, whenever the generated sequence is found in the recorded solution list, the sequence will not be evaluated again, thus saving the simulation evaluation times. As the number of search generations increases, the discrepancy between the number of evaluated solutions and number of generated sequences also increases. Approximately 46 seconds are required to evaluate one solution, which is averaged from 10 simulation replications. The prescreening process, however, takes less than 1 second to determine if the generated sequence has been evaluated. Therefore, it is worth prescreening any new solution against the recorded solutions whenever a sequence is generated to avoid the duplication of simulation. As can be seen in this example, approximately one fourth of generated sequences result in duplicate solutions. We thus save one fourth of simulation runs by avoiding duplicated evaluations.

With random generation, some infeasible solutions are generated during the search. Since a large penalty cost is assigned to the infeasible solutions, those solutions have only a tiny chance of being selected as parents producing offspring, Thus, the largest fraction of infeasible solutions is found in the initial population. Very few reproduced offspring are infeasible since their parents are mostly feasible solutions. In the above GA search processes, approximately 5% of solutions are infeasible among all generated solutions. The proposed GA search method does bypass most infeasible solutions in the solution space while infeasible solutions are very unlikely generated in search process.

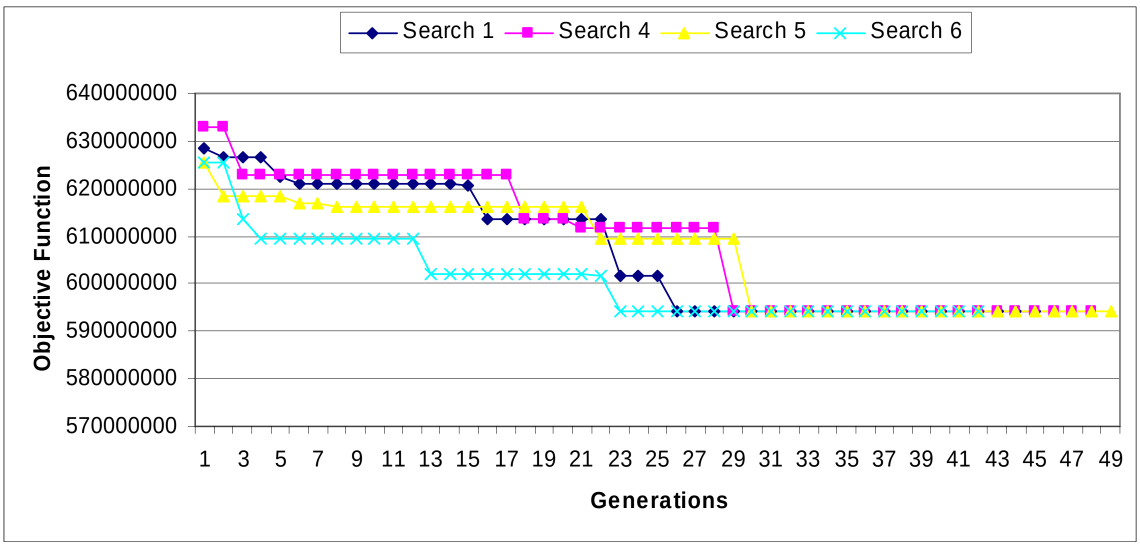
Table 4 shows the optimized solutions and four project implementation schedules for four of the searches which produce 4 different optimized solutions in 20 search replications. It indicates that the objective function for this complex combinatorial problem does have several local optima. As can be seen, with independent regional budget constraints, more than one project from different regions may be funded at the same time (e.g., project #11 at lock #0 in region #1 and project #13 at lock #2 in region #2). The evolution of objective values from those 2 replications is plotted in Figure 8. The optimized solutions step down relatively quickly in early generations and converge at the end of genetic search when the best solution in the previous generation is always saved in the current generation.

Table 4 Optimized Solutions

GA Search 1 Total Cost ($594,187,215)				GA Search 2 Total Cost ($593,965,950)			
Project #	**Lock #**	**Region #**	**Time**	**Project #**	**Lock #**	**Region #**	**Time**
2	7	2	0.29	8	6	2	0.39
6	1	1	0.50	6	1	1	0.50
8	6	2	0.67	2	7	2	0.67
16	4	3	0.68	16	4	3	0.68
11	0	1	1.13	11	0	1	1.13
13	2	2	1.13	13	2	2	1.13
18	3	3	1.55	18	3	3	1.55
GA Search 8 Total Cost ($601,896,915)				**GA Search 14 Total Cost ($601,550,790)**			
Project #	**Lock #**	**Region #**	**Time**	**Project #**	**Lock #**	**Region #**	**Time**
2	7	2	0.29	2	7	2	0.29
6	1	1	0.50	11	0	1	0.63
8	6	2	0.67	8	6	2	0.67
16	4	3	0.68	16	4	3	0.68
11	0	1	1.13	6	1	1	1.13
14	2	2	1.17	13	2	2	1.13
18	3	3	1.55	18	3	3	1.55



(a)



(b)

Figure 8 GA Search Performance

In this test example, the optimized solution (with an objective value of $594,187,215) located in most search processes (15 out of 20) is slightly inferior to the best one (with an objective value of $593,965,950) found in two other search processes. The discrepancy between these two final objective values is due to the sequence rather than choice of projects. That is, most local optimal solutions contain the same projects (projects #2, #6, #8, #11, #13, #16, #18), but different implementation sequences (2→6→8→16→11→13→18, or 8→6→2→16→11→13→18). Therefore, increasing the mutation rate or the number of search generations might be helpful in finding the globally optimal solution.

Using the 3rd, 10th, 11th, 12th, 15th, 16th search processes as an example, Table 5 shows how the parameters specifying mutation rate or the number of search generations affect the optimization results. Scenarios with “-2”, “-3” and “-4”are presented with different GA parameters, mutation rate (0.07 or 0.7) and termination conditions (20 or 50 unchanged solutions). Scenarios “-1” are taken from the original searches in Table 3, with 0.07 for mutation rate and 20 for number of unchanged solutions.

Table 5 Search Scenarios with Different GA Parameters

Scenario	GA Parameters		Search Results			
	Mutation Rate	# of Unchanged Solutions	# of Gen.	# of Generated Sequences	# of Evaluated Solutions	Optimal Total Cost ($)
3-1	0.07	20	25	384	313	594,187,215
15-1	0.07	20	29	448	365	594,187,215
16-1	0.07	20	29	466	363	594,187,215
12-1	0.07	20	31	481	391	594,187,215
10-1	0.07	20	32	487	388	594,187,215
11-1	0.07	20	33	502	411	594,187,215
3-2	0.7	20	21	609	357	594,187,215
15-2	0.7	20	32	905	515	593,965,950
16-2	0.7	20	44	1191	667	593,965,950
12-2	0.7	20	38	1073	622	593,965,950
10-2	0.7	20	54	1491	793	593,965,950
11-2	0.7	20	41	1149	662	593,965,950
3-3	0.07	50	92	1393	932	593,965,950
15-3	0.07	50	95	1393	969	593,965,950
16-3	0.07	50	85	1271	895	593,965,950
12-3	0.07	50	99	1474	973	593,965,950
10-3	0.07	50	92	1378	932	593,965,950
11-3	0.07	50	88	1318	947	593,965,950
3-4	0.7	50	51	1408	706	594,187,215
15-4	0.7	50	62	1721	869	593,965,950
16-4	0.7	50	74	1991	1018	593,965,950
12-4	0.7	50	68	1916	936	593,965,950
10-4	0.7	50	84	2282	1099	593,965,950
11-4	0.7	50	71	1987	991	593,965,950

As can be seen, the scenarios with increasing mutation rate do increase the number of offspring, i.e., the number of generated sequences. Convergence in fewer generations is also shown in those scenarios. When increasing the search time, i.e., by allowing 50 rather than 20 generations without any improvement, more generations are required in GA search but better solutions may be found. Changing either or both of those GA parameters does help in finding lower optimized solutions in most cases (except scenarios 3-2 and 3-4).

Performance Comparison of GA Operators

The proposed simple evaluator is used for testing the performance of different GA operators. Seven test scenarios are first designed to test 7 different operators. 30 search processes are conducted to show the effects of the GA's probabilistic nature.. The mutation rate and crossover rate are set as 0.5 for all test scenarios. All searches are terminated after 100 generations without any improvement.

Table 6 shows the number of generations needed with different GA operators to find the optimized solution. 30 GA searches are run for each operator. Each search finds the same optimized solution. On average, all GA crossover operators perform similarly based on the number of generations required for finding an optimized solution. However, in each individual search, the OX operator outperforms the other operators by a factor of 12 among 30 GA searches. The VM operator also outperforms the other two mutation operators.

Table 6 Number of Generations for GA Operators

GA Search	IM	SM	VM	OX	PMX	PBX	OBX
1	126	138	142	132	151	176	135
2	109	105	105	106	106	125	103
3	115	158	111	109	106	104	111
4	110	108	113	106	109	113	113
5	143	153	185	121	115	106	161
6	104	106	105	104	116	106	102
7	121	134	155	105	114	126	118
8	114	135	142	105	120	123	141
9	149	135	106	130	133	113	122
10	125	112	108	139	133	127	130
11	110	102	181	114	108	101	117
12	155	155	114	118	100	110	127
13	141	126	104	164	119	107	143
14	131	131	131	111	131	118	196
15	126	161	161	102	102	113	109
16	152	174	133	110	126	113	130
17	132	117	114	109	109	109	109
18	117	117	117	178	120	137	140
19	111	111	195	127	135	126	127
20	135	114	196	110	105	113	109
21	115	193	152	105	105	115	114
22	134	110	148	120	203	117	105
23	118	110	136	235	134	111	111
24	112	119	119	100	100	122	116
25	116	116	124	141	122	123	110
26	137	124	137	109	126	120	108
27	118	114	118	110	145	130	114
28	120	146	173	115	123	124	107
29	137	107	131	131	108	117	117
30	137	137	129	120	120	112	131
Avg.	**125.67**	**128.93**	**133.67**	**122.87**	**121.47**	**118.57**	**122.53**
Outperform times	**3**	**4**	**7**	**12**	**8**	**4**	**4**

Conclusions

In general, when solving an optimization problem, introducing more constraints makes problems more specific but also more complex for modeling. With more constraints, the solution space of an optimization problem is reduced, which may then decrease the required search efforts. The performance of a GA is also related to its operators that help create diversities among search solutions. Since the optimization method can be fully separated from the simulation model, the efficiency of different GA operators can be tested with a simple evaluator function that temporarily substitutes for microsimualtion.

When considering precedence relations, the precedence constraints are specified as project information inputs. There could be multiple relevant precedence relations at lock locations. Whenever a project sequence is generated, all the precedence constraints are checked before starting simulation evaluation. If any one of the precedence constraints is violated, the objective value of this generated sequence is assigned a large cost as penalty. With penalties, those sequences that violate the precedence constraints are very unlikely selected to reproduce offspring. The solution space is greatly reduced after considering precedence constraints. With our designed GA search, the largest fraction of infeasible solutions is found in the initial population since those infeasible solutions have only a very slight chance of being selected as parents for future solutions.

When considering regional budget constraints, all the projects are labeled with a regional code based on their lock locations. Independent regional budgets which cannot be shared among regions fund projects within their regions. Similarly to the overall budget constraint, the regional budget constraints are applied in determining the project implementation schedule. Those regional project implementation schedules then determine the overall project implementation schedule, which is evaluated through simulation. With regional budget constraints, projects are not necessarily funded one at a time and may be funded simultaneously in different regions. The modified SIMOPT is able to consider regional budget constraints while solving the problem of sequencing and scheduling mutually exclusive projects.

To reduce running time in a simulation-based optimization model, any newly evaluated solution is recorded in a “solution list”. Whenever a new sequence is produced from mutation or crossover operations, a pre-screening process is first performed to check throughout the solution list. If that solution is also found in the list, its simulation is omitted and its fitness value is directly assigned from the saved records. The search scenarios show discrepancies between the number of generated sequences and the number of actually evaluated solutions, due to avoiding duplicate simulations of the same solutions. The search time saved from not simulating those duplicated sequences (i.e., unevaluated solutions) is even larger if the number of generations is increased.

In comparing the search performance of designed GA operators (mutation and crossover operators), it is faster to use a simple evaluator rather than a long-running simulation model. The required parameters used in the simple evaluator include traffic volumes at locks and lock capacities. To consider lock interdependence, the distances between locks are included in the evaluator. The performance of different GA operators is then compared by using the simple evaluator in GA optimization. The results show that OX operator slightly outperforms the other operators and the VM operator outperforms the other two mutation operators.

According to previous plans, it seems desirable at this stage to proceed with further improvements in the optimization algorithms, including (1) smarter problem-specific operators, (2) multiple alternatives at given locations that may be implemented at different times, (3) ability to consider tradeoffs between construction times and costs, (4) adaptations for network-level maintenance planning and scheduling, (5) adaptations for lock-component-level maintenance planning and scheduling, and (6) distributed processing of the optimization on multiple parallel computers.

References

Wang, S. and Schonfeld, P. (2002), "Development of Generalized Waterway Simulation Model for Waterway Operations", Transportation Research Board 2002 Annual Meeting (CD-ROM 02-2194)

Wang, S. and Schonfeld, P (2005), "Scheduling Interdependent Waterway Projects Through Simulation and Genetic Optimization", ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering, May/June 2005, Vol. 131, No. 3, pp. 89-97

Wang, S and Schonfeld, P., "Genetic Algorithms for Selecting and Scheduling Waterway Projects (Phase 1)", NETS Report, April 2006

Tao, X. and Schonfeld, P. (2006), "Selection and Scheduling of Interdependent Transportation Projects with Island Models", forthcoming in Transportation Research Record, TRB.

Appendix

GA Phase 2 Scope of Work

In the Design Document development phase, a "testbed" simulation-optimization model was used to demonstrate the feasibility of using simulation and GA optimization to determine optimal solutions to problems requiring simulation as the objective function evaluation tool. During that demonstration, several needed enhancements to the GA optimization capabilities were identified. The following tasks describe those activities which are related to enhancing the capabilities of the GA optimization model.

Task 1 Genetic algorithm

1.1 Compare the performance of various mutation and crossover operators

1.2 Add prescreening rules to avoid simulating solutions that are unpromising or violate constraints

Task 2 Evaluation / Simulation model

2.1 provide simple evaluator for testing optimization algorithms

Task 3 Project selection / sequencing / scheduling

3.1 Consider precedence relations

3.2 Consider more budget constraints (e.g. regional limits, new construction vs. matintenance)

Task 4 Continued participation on NaSS team

4.1 Continue to participate in teleconferences and face-to-face meetings. At the time of scope development it is anticipated that bi-weekly teleconferences will continue throughout the period of this scope. In addition, at least one face-to-face meeting between team members is anticipated.

4.2 Specific assignments. It is anticipated issues and activities will arise during the period of this scope for which CEE-UMD will be tasked. If the level of effort involved requires significant additional time and resources, this scope may be modified to provide additional funds and time to CEE-UMD.

navigation · economics · technologies

The NETS research program is developing a series of practical tools and techniques that can be used by Corps navigation planners across the country to develop consistent, accurate, useful and comparable information regarding the likely impact of proposed changes to navigation infrastructure or systems.

The centerpiece of these efforts will be a suite of simulation models. This suite will include:

- A model for forecasting international and domestic traffic flows and how they may be affected by project improvements.
- A regional traffic routing model that will identify the annual quantities of commodities coming from various origin points and the routes used to satisfy forecasted demand at each destination.
- A microscopic event model that will generate routes for individual shipments from commodity origin to destination in order to evaluate non-structural and reliability measures.

As these models and other tools are finalized they will be available on the NETS web site:

http://www.corpsnets.us/toolbox.cfm

The NETS bookshelf contains the NETS body of knowledge in the form of final reports, models, and policy guidance. Documents are posted as they become available and can be accessed here:

http://www.corpsnets.us/bookshelf.cfm

navigation · economics · technologies