Genetic Algorithms for Selecting and Scheduling Waterway Projects

By Shiaaulir Wang and Paul Schonfeld

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

April 2006

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. It provides a promising demonstration of simulation-based optimization.

Since the developments of simulation and optimization components are largely separable, this testbed model can be used to quickly test optimization improvements without running more detailed and longer-running simulations. The improved optimization models are intended to work with the next generation NaSS waterway simulation model which is developed under the NETS program of the Corps of Engineers. As a testbed, SIMOPT is modified here to consider project construction time and capacity reductions during construction, avoid duplicate evaluations and consider mutually exclusive projects at any locks.

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.

The following sections focus on the issues of optimization, simulation-based optimization modeling and project scheduling. The SIMOPT model is presented to demonstrate the capabilities of a simulation-based optimization model in scheduling waterway improvement projects. It is expected that the optimization methods developed and tested with SIMOPT can then be applied with the next generation NaSS waterway simulation model.

Optimization

Optimization is a mathematical process that searches for the solution which best satisfies a stated objective. Any optimization problem can be formulated with an objective function to be minimized or maximized, and subject to constraints of budgets, capacities, construction times or facility closure times. Various optimization algorithms are available for solving different levels of optimization problems. Calculus, enumerative search, mathematical programming and branch and bound algorithms may be used to solve exactly some optimization problems which are sufficiently small or well behaved . Heuristic optimization methods such as simulated annealing, tabu search, genetic algorithms and swarm intelligence may be tried for problems that are relatively large of have numerous local optima.

If the decision variables are discrete, the optimization problem is a combinatorial optimization problem, whose optimal solution is found from the enumeration, combination and permutation of several discrete elements. Since it is practically very hard to identify the global optimum as number of decision variables becomes large, rather than finding the perfect "optimum solution", we seek a very good (or "near-optimal") solution. In solving a complex optimization problem, the objective function must be repeatedly evaluated. This function might be computed or estimated with a simple equation, a queuing model, and other methods. If the system analyzed is complex enough and subject to probabilistic variations, it is difficult to evaluate it or its objective function without a detailed simulation model.

Simulation-Based Optimization Model

For years, there has been considerable interest in combining simulation and optimization models. With a number of controllable decision variables and an objective function to be maximized or minimized, the optimization model runs the simulation model and eventually determines a combination of the decision variables that produces an optimal or near optimal solution.

A simulation model is commonly used for complex probabilistic systems. Since those systems are hard to evaluate analytically, the objective function is not fully specifiable. There are several advantages of applying simulation models:

- System performance can be estimated under specified operating conditions.
- 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.

A possible simulation-based optimization model is presented in Figure 1. The optimization module first instructs the simulation module to simulate some initial system configurations, i.e. combinations of decision variables for the system. The simulation model evaluates and computes the objective function for each analyzed configuration. Based on the above results, the optimization model selects new combinations of decision variables to be simulated, until further improvements become insignificant. That is, the outputs from these simulations are fed back into the optimization module, which then uses its built-in search algorithm to generate additional configurations to simulate, etc. The whole process is continued, while insuring that all constraints are satisfied, till the termination rule in the optimization module is reached.

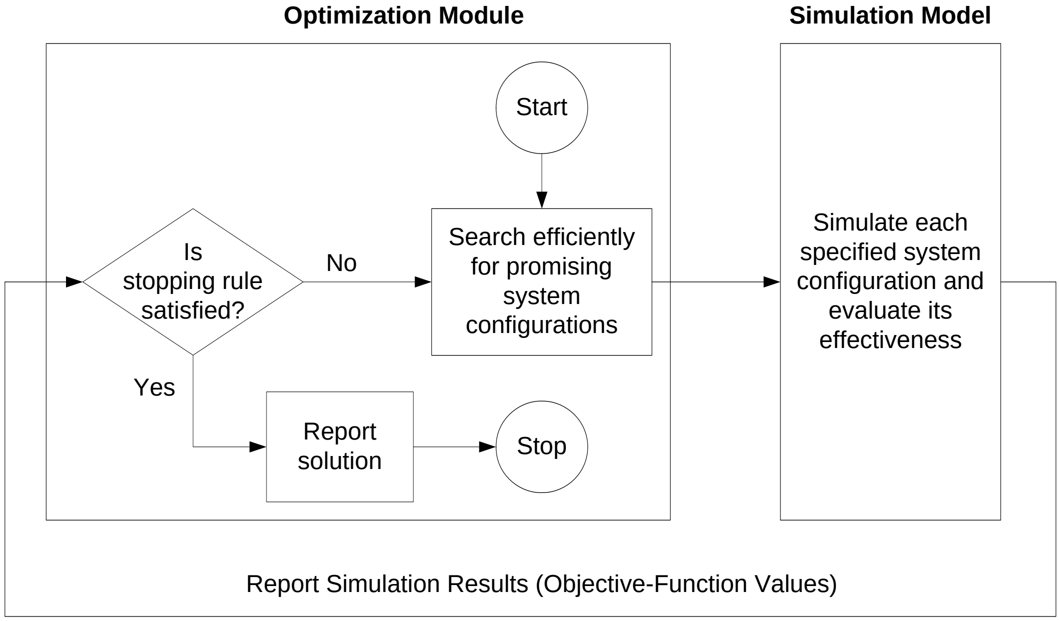


Figure 1 Interaction between Simulation and Optimization

However, although the operational steps are indeed workable, it is important to note that the results are not absolutely guaranteed to be optimal. The optimization results depend on how the options, parameters, and tolerances are specified. A good optimization model can efficiently reach a near-optimal configuration. The difference between the global optimum and the near optimal solution is usually insignificant in practice, considering the uncertainties in inputs and in functional relations.

Genetic Algorithms

Characteristics

An efficient optimization algorithm must satisfy two requirements in finding the global optimum: sufficiently explore the search space and exploit the knowledge gained at the previously visited points. (That search space includes the points representing the various combinations of decision variables.) Rooted in natural genetics and computer science, genetic algorithms (GAs) treat the problem as the environment, and consider a set of possible solutions to the problem as the population. A procedure that (somewhat) mimics the natural evolution is established to select individuals for reproducing offspring according to their “fitness” to the environment (i.e. the problem). Each individual (which constitutes a tentative solution to the problem) in the population is represented by a set of encoded genes called a chromosome. After several generations, the most adapted individuals will survive and have a higher chance of reproducing offspring. If the algorithm is well designed, the population will converge to an optimal solution to the problem.

There are several characteristics distinguishing GAs from other conventional optimization techniques. At any stage in the search GAs work with a set of solutions rather than one single solution. This feature enables GAs to escape from local optima in their multi-directional global search. Besides, no specific function (i.e. formulated objective function) for the mathematical expression of a given problem is required in GAs. Thus GAs are able to handle any kind of objective function and constraints, and are especially suitable when the objective function is quite noisy (i.e. with numerous local optima). The GA search approach is at least partially probabilistic in the way population members are selected for future generations and in the frequency with which various operators are applied.

Design of GAs

Figure 2 shows the basic GA procedure in optimization search process. The application of GAs to a specific problem includes several steps.

1. Solution encoding
   Originally, a potential solution to the problem is encoded into a binary string, called a chromosome, of a given length which depends on the required precision. In terms of problem characteristics, some other ways of representing solutions are necessary, such as integer coding for solving combinatorial optimization problem.
2. Initial population
   Generally, the initial population is randomly generated. If good solutions can be included in the initial population, the optimization time can be reduced somewhat.
3. Fitness function
   When GAs are applied, the fitness function is the objective function to be optimized. The fitness value of each individual solution from a population must be evaluated.
4. Selection
   The individuals in the population are selected to reproduce offspring according to their fitness value. Typically, proportional selection chooses individuals by calculating their relative fitness values. If necessary, scaling and ranking schemes provide alternatives for measuring fitness other than using raw values directly
5. Genetic operators
   Classic GAs provide two types of genetic operators – crossover and mutation. A crossover operator generates the offspring from two parents by swapping their genes at some randomly chosen position of the chromosomes. A mutation operator alters (according to some rules and/or probabilities) one of more genes of one selected parent chromosome in order to increase the population variability.
6. Population replacement
   Replacement creates a new population for the next generation and is strongly related to the selection process. Two issues arise in this phase – sampling space and sampling

mechanism. Along with selection, both of them have a significant influence on selective pressure and thereby on genetic algorithm behavior.

7. Termination and convergence
   Usually, the genetic system is terminated through a pre-specified number of generations. Another termination rule could be as follows: stop the search process after the solution of the best sequence remains unchanged for the last m generations.

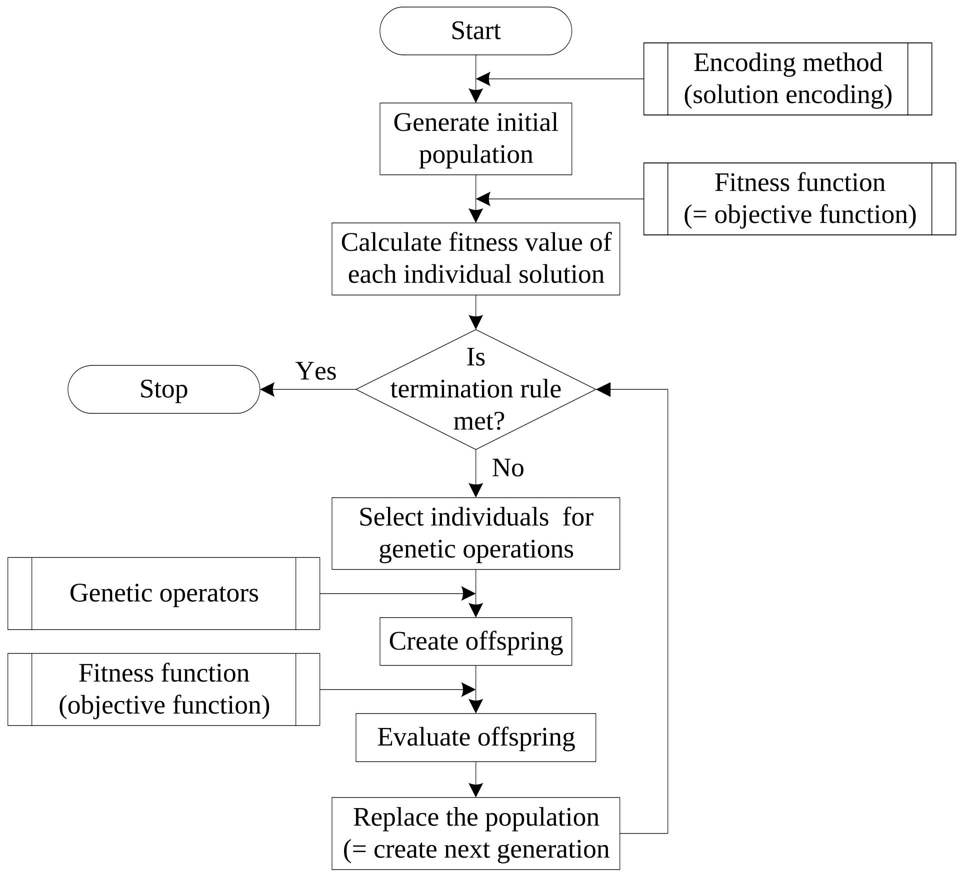


Figure 2 GA Procedure

For the integration of waterway simulation and optimization, a genetic algorithm is chosen to perform the optimization search. Several steps are included in an ordinary genetic algorithm:

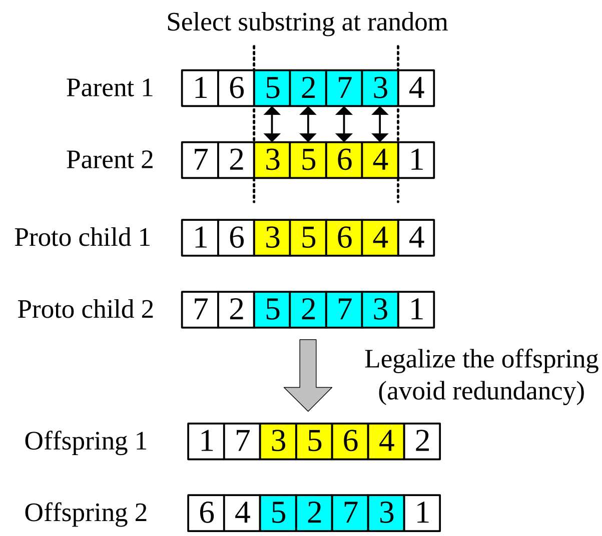
- Step 1: Create initial population of solutions (i.e., project sequences).
- Step 2: Evaluate those solutions (with a simulation model in this study).
- Step 3: Select the better individual solutions for genetic refinement.
- Step 4: Create new solutions using mutation, crossover, or other operators.
- Step 5: Evaluate new solutions.
- 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

Genetic Operators

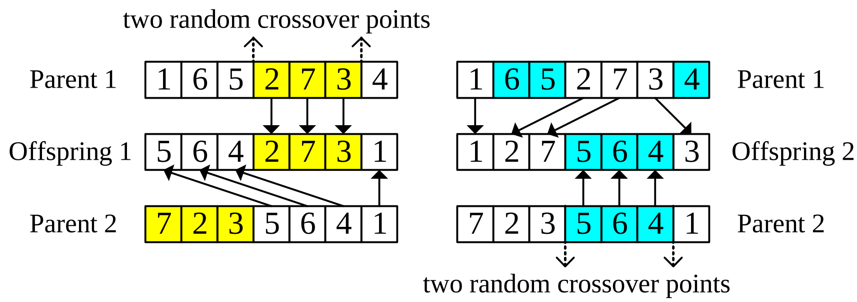
In general, there are two types of genetic operators: mutation operators and crossover operators. During the past decades, several operators have been proposed, widely discussed and served as standard operators for solving sequencing problems. Those are discussed below.

Crossover Operators

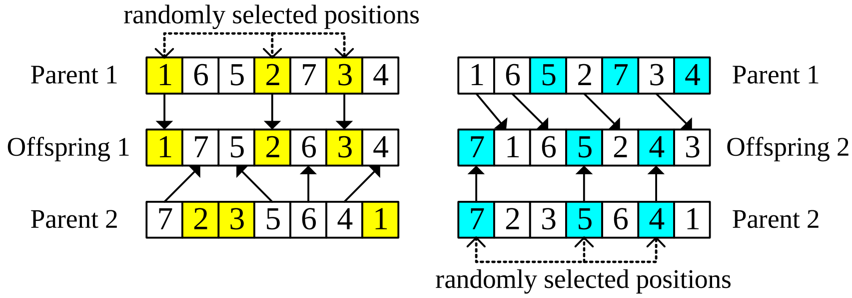
- Partial-Mapped Crossover (PMX)



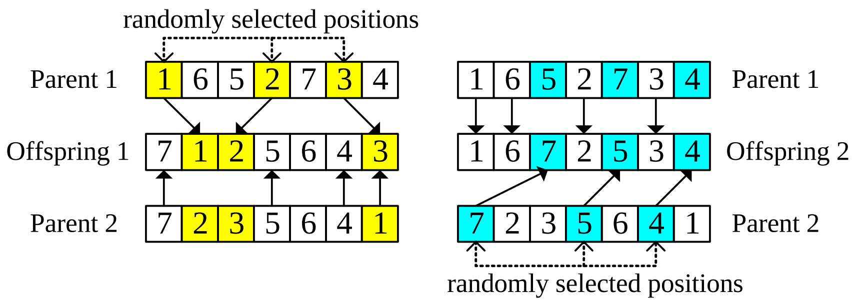
- Order Crossover (OX)



- Position-Based Crossover (PBX)

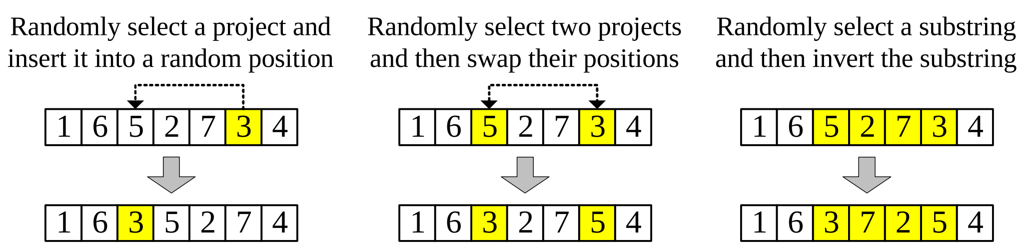


- Order-Based Crossover (OBX)



Mutation Operators

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



(a) IM (b) EM (c) VM

Project Scheduling Problems

Investment planning, also known as capital budgeting, is the process of determining which investments or candidate projects will be funded and pursued to meet the pre-specified objectives over a planning horizon. It includes the tasks of project evaluation, project selection, project sequencing and project scheduling.

With a constrained budget available for various investment combinations, project selection and sequencing is a large combinatorial optimization problem. The solution space increases more than exponentially with problem size, i.e., with the number of projects considered. Furthermore, project interdependence increases the difficulty of solving project scheduling problems. Project benefits and/or costs might depend on which other projects are implemented. Especially in transportation networks, there are traffic interactions between adjacent projects. Some capacity improvement projects may mostly shift elsewhere the bottlenecks and delays. Therefore, those interdependencies make the evaluation even more complex if the improvements from some projects affect the operations and benefits of other projects.

The literature includes various methods of evaluating schedules of interdependent projects, such as queuing metamodels, equilibrium traffic assignment, artificial neural networks and microscopic simulation models. Some optimization approaches are also explored in previous studies, such as swapping algorithms, branch and bound algorithms, Lagrange relaxation, simulated annealing and genetic algorithms.

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 (at 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 3, 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.

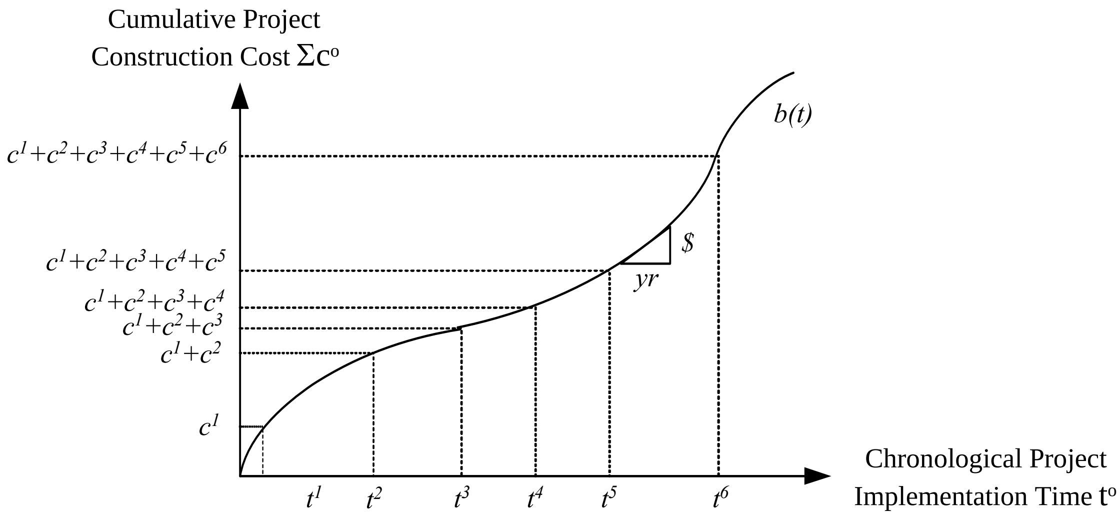


Figure 3 Relations of Budget Flow, Cumulative Cost, Project Sequence, and Project Schedule

If there are N lock improvement projects in N fixed lock locations, the project selection process chooses a subset of n projects from a set of N desirable projects in the most desirable order. Meanwhile, the project scheduling process determines the sequence for implementing projects as well as the timing for the selected projects. Given a sequence and timing of lock capacity expansions, the project evaluation process estimates the system performance, which is usually defined as system delay costs. The performance measures cannot be determined until a project portfolio is specified. When project interdependencies exist, any lock improvement may affect traffic characteristics at other locks. As a practical matter, if there were a large number N of lock improvement projects and only n projects will be selected due to budget constraints, the solution space for project selection and sequencing including all possible combinations and permutations would be

$$\sum_{n=0}^{N} \frac{N!}{n!(N-n)!} \cdot n! = \sum_{n=0}^{N} \frac{N!}{(N-n)!}$$

The above equation indicates that the size of the solution space increases more than exponentially with the number of candidate projects N. If N is not very small, a full enumeration search becomes infeasible for finding the optimal combination among all alternative project sets. For jointly considering project selection, sequencing and scheduling, the solution space is even larger. Through the budget constraints, the size of the project sequencing problem becomes $n!$, which is smaller than that of the original problem, and each of the $n!$ sequence corresponds to a feasible solution.

If the project size (or changed capacity) is lumpy rather than continuous at any project location, the solution space is increased by the factor of $\prod_{i=1}^{n} P_i$, where P_i is the number of possible projects at lock i. The project scheduling problem will then consider more combinations and permutations.

Problems of Scheduling Waterway Improvement Projects

Scheduling waterway improvement projects is considered as a combinatorial optimization problem. The objective function is set to minimize the system costs or maximize the net benefits over a multi-year period. There may be several constraints regarding budgets (possibly by region or type of expense), precedence, mutually exclusivity, minimum improvement steps, construction times, capacities, service quality, and geographic distributions. It is difficult to analytically model the probabilistic features of a waterway system. Hence, a simulation model is adopted for evaluating the system with each schedule. A conceptual approach for combining simulation and optimization models to solve our problem is shown in Figure 4.

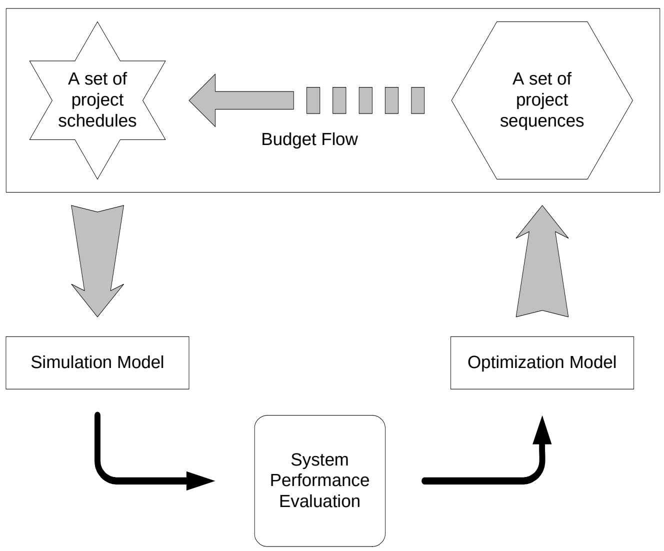


Figure 4 Structure of SIMOPT Problem

The inputs required for this combined simulation-optimization model should be information on improvement projects, system network configuration and network relevant variables. The outputs for these two interacting modules should be performance measures from the simulation model and project schedule (implementation timetable of the selected projects) from the optimization model.

Inland Waterway Simulation Model

Due to the probabilistic features in waterway traffic, a microscopic, discrete-event simulation model is preferred to model the inland waterway operation. The purpose of using a waterway simulation model is to evaluate the performance of inland waterways with specified system characteristics, as well as analyze short-term system variability and control alternatives. In the long run, the system evolution can also be assessed.

Coding is a major aspect of building a complex simulation model. One of the most important features of such a system simulation model is its portability. With portability, a model can be easily reused for other geographic areas or networks. With different levels of details for different study purposes, such a simulation model could have wide applicability for various purposes, such as forecasting, design, control, project selection and scheduling, maintenance planning and scheduling, reliability analysis.

Some major factors should be considered in inland waterway simulation model:

- Probabilistic aspects of waterway traffic, lockage times, travel times, stalls, etc.
- Demand variability
    - Demand sensitivity to service levels
    - Demand sensitivity to construction and closures
    - Demand sensitivity to improvement projects
- Operational lock control alternatives
    - Lock control strategies
    - Chamber interference at multiple-chamber locks
    - Chamber assignment for multiple-chamber locks

Integrated Waterway Simulation and Optimization

The inland waterway simulation model is designed as a discrete-event simulation model. It includes various "network operation events". In addition to those events in the simulation kernel, "project construction events" have been added to update some system variables during the simulation. Those project construction events come from the project implementation schedule whose sequence is generated by the GA. The schedule is then determined based on budget constraints. The project implementation schedule is then fed into the simulation model and evaluated by the simulation model. The integration of simulation model and optimization model is shown in Figure 5. Two blocks show the two separate models for simulation and optimization. They are connected by the information they exchange about decision variables of the project implementation schedule.

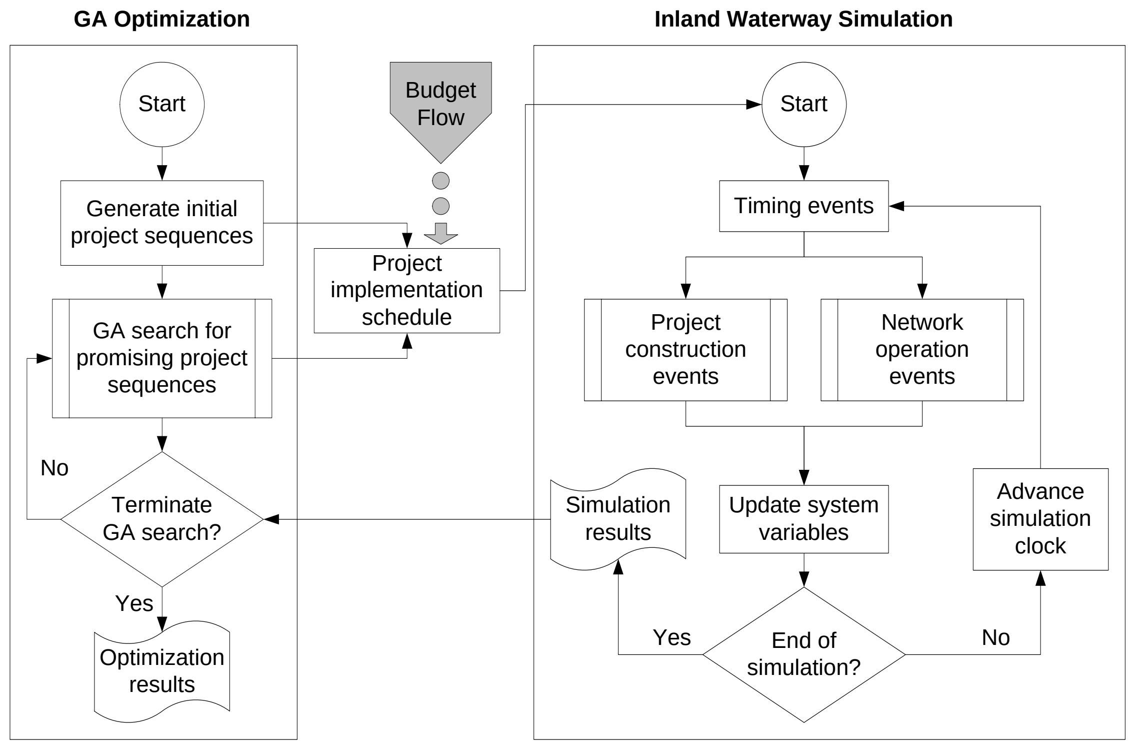


Figure 5 Integration of Waterway Simulation and GA Optimization

SIMOPT

SIMOPT was developed at the University of Maryland and presented to USACE's Institute for Water Resources (IWR) in a meeting on July 27-29, 2005. It serves as a proof of concept model that can be tested and manipulated to help identify problems that may arise in future when a much more complex simulation model is combined with optimization. As a testbed, SIMOPT can run a simulation alone or combine it with a genetic algorithm to optimize project scheduling. After some discussion and refinements, the latest version of SIMOPT was delivered to USACE in late September 2005, accompanied by the SIMOPT presentation file, which mainly serves as a simple user guide for SIMOPT.

SIMOPT Model Assumptions

The simplifying assumptions in the original SIMOPT include the following:

- Simulation Model
    - Each tow maintains a constant number of barges through the entire trip, even if it is necessary to disassemble barges while passing through the locks. That is, a tow's size is assigned when that tow is generated, and there is no refleeting during its trip.

- Each tow maintains a constant speed between its origin and destination ports, either upstream or downstream.
- There is always enough equipment, such as towboats and barges, for waterway shipments wherever needed in the network.
- The queue storage space at each lock is unlimited for both directions.
- Components of lockage process are simplified with a single service time distribution.

- Optimization Model
  - The implementation (i.e., mainly construction) costs of projects are independent and additive. Whenever the ccumulative budget reaches the required construction cost for an additional, that project implementation is completed.
  - The budget is accumulated continuously as a function of time over the planning horizon.
  - The implementation of one project does not yet depend on the existence of the other projects.
  - The increase in lock capacity is indicated by the increased service rate (i.e., the inverse of service time).
  - A capacity is specified without affecting the number of chambers.
  - Lock capacity increases instantaneously after a lock improvement project is completed. After the project selection and sequencing are completed the project completion times can be uniquely determined.
  - There is one and only one improvement project at each lock location. No other alternatives are yet considered.
  - Budget constraints are always binding, i.e., there are never enough funds for all justifiable projects.

Model Features

SIMOPT is built with an inland waterway simulation model (Wang 2001) and a GA search algorithm. The simulation model incorporated in SIMOPT is designed to be a portable, data-driven model which can be applied on various waterway tree networks without re-coding the computation kernel. The optimization model employed in SIMOPT is deliberated with genetic algorithm, especially in solving sequencing problems.

SIMOPT has a simple user interface. It allows users to specify required input files, which should be prepared ahead of running the SIMOPT model, and some other basic parameters such as the duration of the simulation period and the number of simulation replications needed to reduce the variance of the combined stochastic processes of simulation and optimization.

Demonstrations of SIMOPT have exhibited the following features of this model:

- Run Simulation
  - Performance of designed simulation scenario

- Project Evaluation
  - Evaluation of single projects
  - Evaluation of any given project sequence
  - Evaluation of lock control policies
- Run Optimization
  - Optimization of project selection, sequencing and scheduling

Network Examples

Two network examples are provided with the latest version of SIMOPT. One is a simple, artificial test network (shown in Figure 6) and the other is a section of the actual US inland waterway network, the Upper Mississippi River (shown in Figure 7). The latter case is shown in greater detail in Wang's dissertation (referred as "Case Study").

Test Network 1 (Artificial Network)

This artificial network includes 5 ports (5 × 5 O/Ds) and 7 locks (3 two-chamber locks and 4 one-chamber locks). Improvement projects are applied at locks to expand capacity, by doubling capacities at single-chamber locks and expanding the capacities at two-chamber locks. This artificial network is used to show how the network configuration inputs are prepared for the simulation module, which was developed with a data-driven approach. Details of the development of simulation model development are shown in Wang and Schonfeld's 2003 TRB paper.

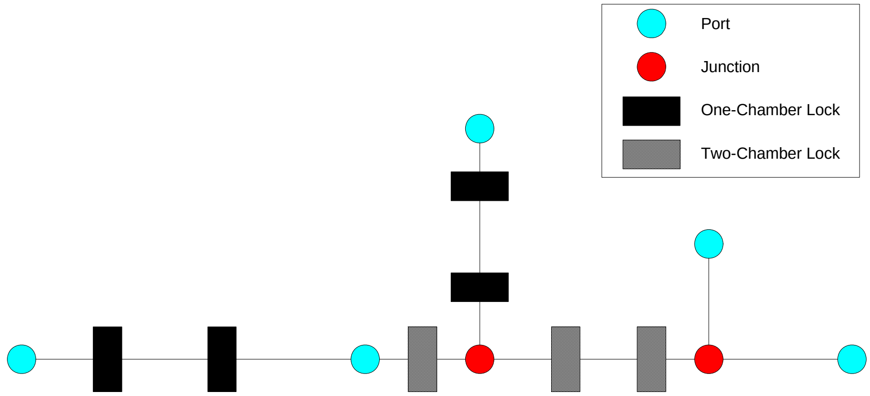


Figure 6 SIMTOP Test Network – Artificial Network

Test Network 2 (Upper Mississippi River)

The simulation model in SIMOPT is capable of simulating a large waterway network, such as Upper Mississippi River area and Ohio River area with 17 major ports and 74 locks. The distance between St. Louis and Cairo exceeds 100 miles, which is enough to eliminate lock interdependence. Therefore, the inland waterway network analyzed here is the Upper Mississippi region area which contains 3 rivers, 7 ports and 36 locks.

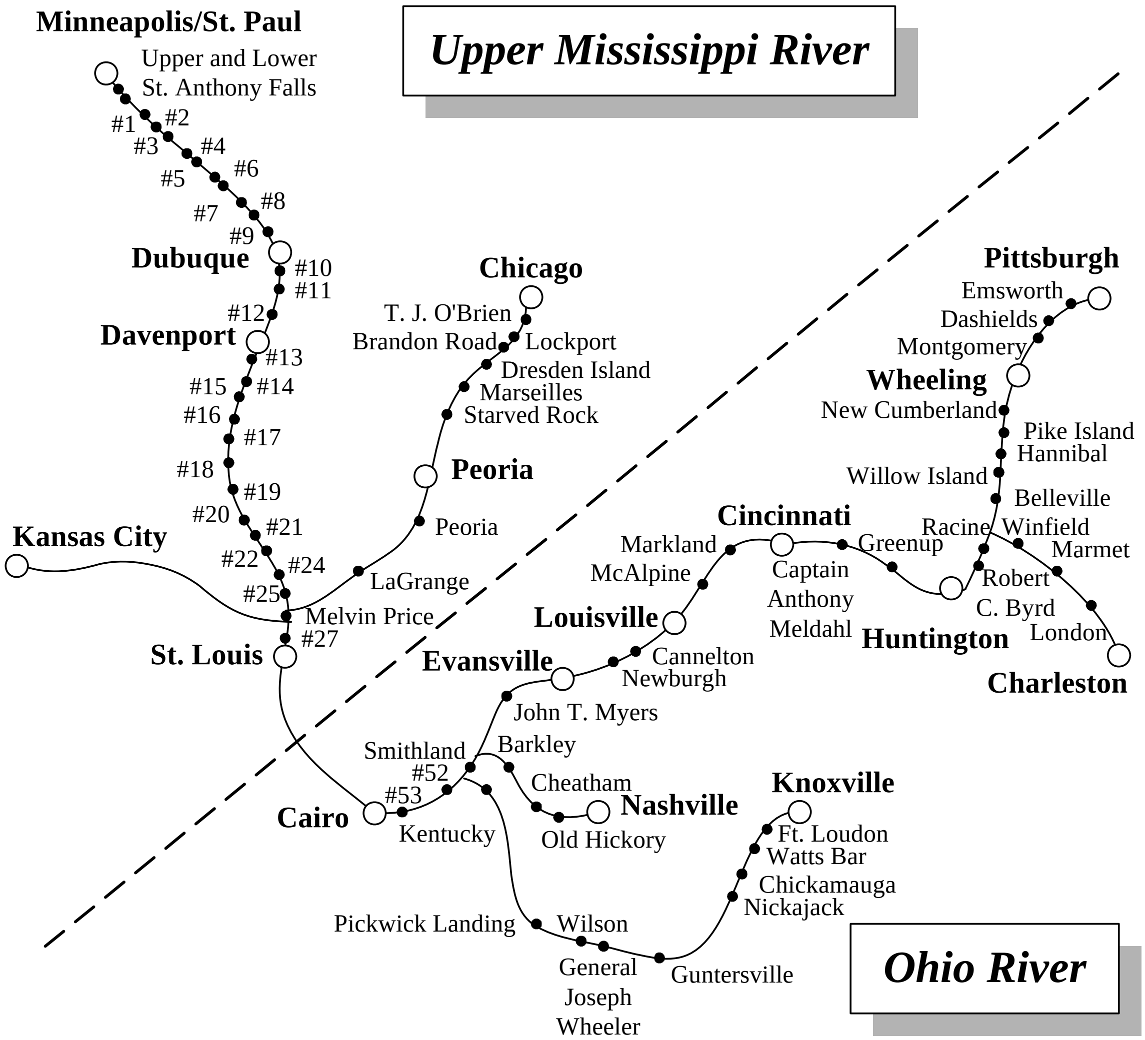


Figure 7 SIMOPT Test Network – Network of Upper Mississippi River

Model Testing

After examining the delays at current locks, locks #15, #16, #17, #18, #19, #20, #21, #22, #24, #25 are selected for improvement projects which double their capacities. Running the simulation model on such a large network, even once, takes considerable time. Besides, hundreds or thousands of evaluations might be necessary to approach the optimal or near optimal solution while using genetic algorithms. Therefore, due to limited computational resources, the simulation is accelerated (within 1.5 years) with high traffic growth and high budget rates.

Simulation Inputs

- O/D matrices

- Matrices of demand growth rates
- Matrices of demand elasticity
- Tow size distribution
- Speed distribution
- Service time distribution
- Link distances
- Number of chambers
- Chamber bias and lockage cuts
- Control alternatives (F → FCFS, S→ SPF)
    - Case 1.1: network-wide FCFS
    - Case 1.2: network-wide FCFS w/ selected SPF

Table 1 Lock Control Settings for SIMOPT

Lock	Control	Lock	Control	Lock	Control	Lock	Control
Up. Falls	F	#8	F	#17	F	#27	F
Lo. 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

Optimization Inputs

- Lock expansion plan

Table 2 Lock Expansion Plans for SIMOPT

Lock Site	Capacity	Cost (10^6)	Current Lock Delays (barge-hrs)	Project Benefit (system total delay savings)
#13	2.0	2.5	3742780	1086416
#16	2.0	1.6	2501000	731052
#17	2.0	2.7	2120250	551020
#18	2.0	2.1	1987090	508484
#19	2.0	1.7	1765470	408528
#20	2.0	2.4	1733540	263210
#21	2.0	2.1	1795420	337892
#22	2.0	1.9	2098990	432320
#24	2.0	2.3	2940650	679700
#25	2.0	2.2	5130450	946204

- Genetic parameters

Table 3 Genetic Parameters for SIMOPT

GA Parameters	*Value*
Population Size	20
Mutation Rate	0.2
Crossover Rate	0.5
Selection	Elite
Sampling Mechanism	Stochastic
Selection Probability	Ranking Scheme
Sampling Space	Large w/ replacement
Termination	5 generations w/o improvement

Optimized Results

Intuitively, if locks are considered individually, the construction projects would be implemented according to the rank of their delay severities, that is #25→#13→#24→#16→#17→#22→#18→#19→#21→#20. Based on two sets of control alternatives designed in previous input tables, the optimized solutions for sequencing and scheduling 10 projects are shown in the following table.

As can be seen in the left side of table, if only physical construction projects are considered and all locks are operated with FCFS, i.e., without changes in lock control, then #22→#16→#25→#13→#18→#24→#19→#21→#20→#17 is the optimized project sequence. It differs from the one ranked according to individual lock delay severities. Also, the rank-based project sequence results in a total delay cost of \$$1.467\times10^9$. The optimized sequence found does have a lower system delay costs of \$$1.448\times10^9$. Further, SPF control improves efficiency and reduces the delays. Therefore, when combining improvement projects with more efficient control at selected locks, the network bottleneck will shift and lock congestion levels will change. The possibility of operating SPF only at selected locks leads to the project sequence shown on the right side of table. Those locks with better control alternatives can have their improvement projects implemented later. The resulting total delay cost is 1.344×10^9.

Table 4 Test Results for SIMOPT

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

The above table illustrates the effect of the optimized project implementation schedule on delay costs and on the volume to capacity (V/C) ratio at the remaining critical bottleneck in the network. Figure 8 indicate the accumulated total delay costs with and without projects over the assumed planning horizon of 1.5 years. The dashed lines indicate the implementation times of the 10 projects. At the end of year 1.5, these improvement projects can save almost 25% of total system delay costs. Figure 8 also presents the change of V/C ratio at the network's bottleneck. Along the time axis, the bottleneck physically shifts over the network as additional projects are implemented. In the current demand model, the elasticity of demand with respect to travel time is determined by a sensitivity coefficient which is specified based on judgment and experience with local conditions. With any positive demand elasticity, lock improvements that reduce delays will attract additional traffic, thus changing the V/C ratio in the network.

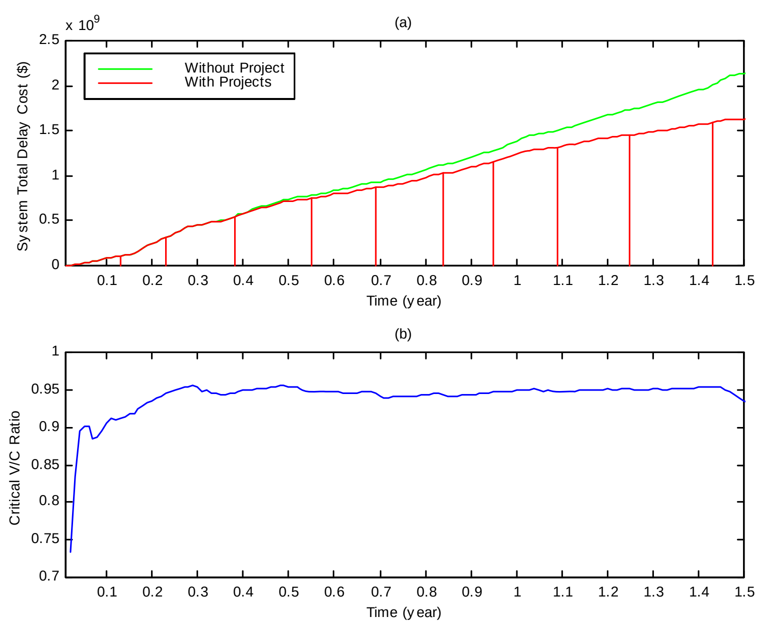


Figure 8 Cost and Network Analysis of SIMOPT Project Implementation

The upper part of Figure 9 compares the results of capital improvement projects and operational control alternatives. As can be seen, the curves intersect around year 0.6. That is, before year 0.6, SPF control can save more delays than capital improvements. The implementation of the first four projects might not be necessary if an effective control alternative is considered. Without projects, the construction costs are also avoided. Finally, the lower part of Figure 9 displays the total delay savings from implementing projects without and with SPF controls. It shows that the system performance can be further improved if more effective lock control and lock expansions are considered jointly.

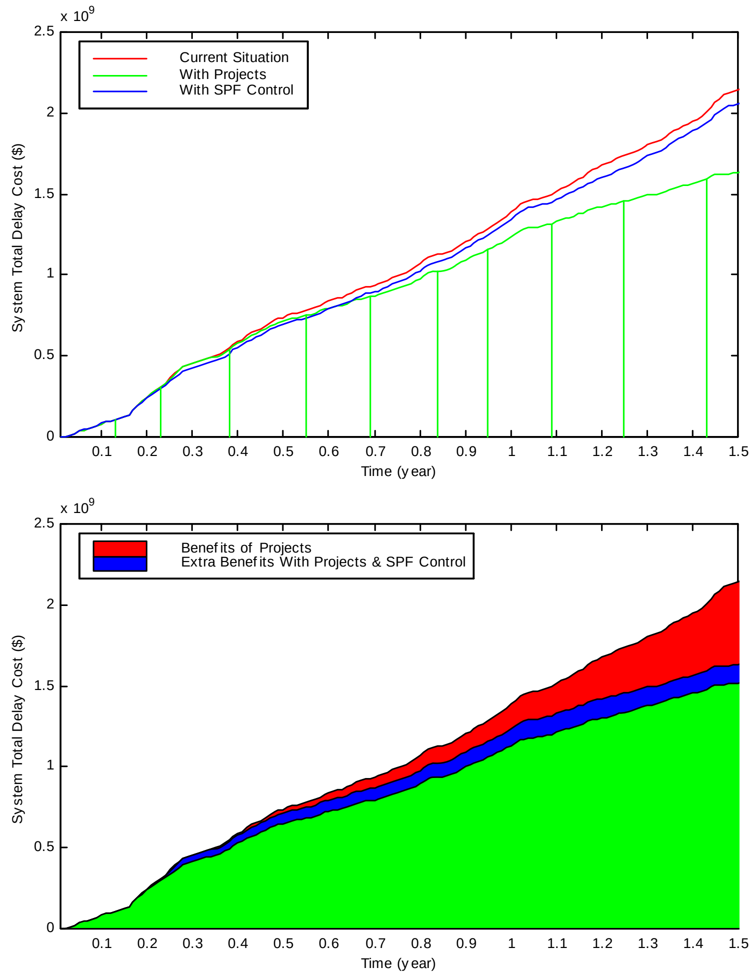


Figure 9 Benefits of Projects with Control Considerations in SIMOPT

NaSS

The Navigation Economics Technologies (NETS) research program is initiated by the US Army Corps of Engineers Institute for Water Resources (IWR) to organize the latest research findings to develop economic tools and techniques for navigation needs. One of the efforts is to improve the analysis models directed primarily at inland navigation, for which the new NaSS (Navigation System Simulation) model is being developed.

Based on the experience with previously developed USACE models and applications, including the basin-level models WAM, ORNIM, and NavSym; and single lock model representations WAM, LCLM and LockSym, the system network model is a discrete event simulation model that generates commodity shipments between ports, moves vessels through reaches and locks, considers flow conservation, takes into account re-fleeting activities at some designated locations, and incorporates shippers response to scheduled or unscheduled closures. For the investment optimization, the SIMOPT model developed at the University of Maryland is used to explore genetic algorithm optimization in conjunction with a network simulation model. Such a model is flexible and adaptable to a wide variety of inland navigation problems addressed by the Corps.

Model Extensions for NaSS

The NaSS design document describes the model's characteristics including the network model, investment optimization model as well as auxiliary tools of data analyzer, result analyzer and data pre-processor. As discussed above, the investment optimization model can be fully separated from system network model in the development stage. After that, the integration of the simulation and optimization models should be a low-risk and straightforward problem. That is, while the optimization models are developed, they may be integrated with either the SIMOPT testbed or with an even simpler evaluation function.

Several needed enhancements to the GA optimization capabilities and simulation complexities were of interest. Thus, the original model assumptions in SIMOPT are reviewed and possible modifications are studied.

In developing future simulation model, the following features should be considered:

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

A more detailed improvement plan could also include:

- Project construction times
- Capacity reductions during construction
- Number and size of chambers
- Maintenance cost
- Failure rates and durations before and after projects.

Some refined optimization features could be included:

- Add 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

Enhanced Work on Genetic Algorithms

According to the Scope of Work drafted for GA enhancement (see Appendix), several tasks are included in the current phase, including considering project construction time and its relevant effects, involving multiple alternatives at the same project location, and increasing the search efficiency in GA optimization process. The task of "optimal timing for projects absent budget constraints" is automatically bound with other tasks.

Project Construction Time

One of the basic assumptions in SIMOPT in solving the project selection / sequencing / scheduling problem is "lock capacity increases instantaneously after a lock improvement project is selected and completed". There is no consideration of project construction time and any possible capacity reduction during the construction period. That is, the system increases lock capacity suddenly, whenever a project is implemented. Therefore, by reviewing the inputs given to the simulation model for project evaluation, the construction related information is simplified and added into data file of project information. As shown in Figure 10, in addition to project ID, project size (i.e., capacity expansion ratio) and project cost, two extra data items are included, namely construction duration and residual capacity ratio (Co. T and Res.).

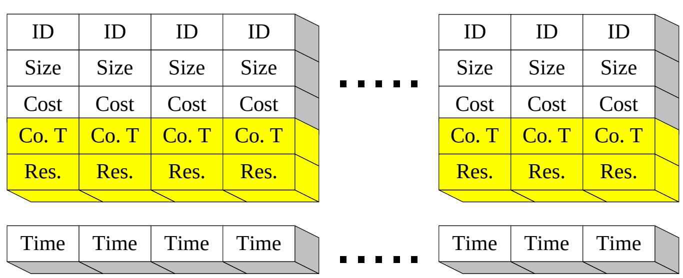
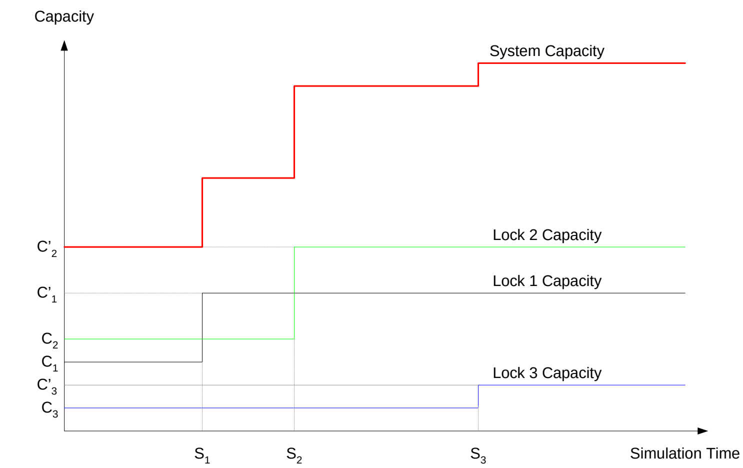
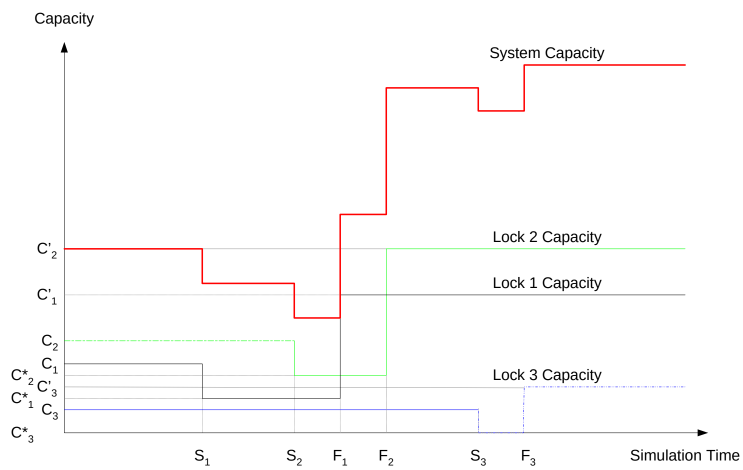


Figure 10 Structure of Chromosome for Considering Project Construction Time

An example is shown in the Figure 11. 3 lock improvement projects are for lock 1, 2, and 3 to increase lock capacities from C_1, C_2, and C_3 to C'_1, C'_2, and C'_3, respectively. Figure 11 (a) shows the lock capacity changes in original SIMOPT without considering construction time and capacity reduction. After considering construction time and capacity reduction during the construction, Figure 11 (b) shows that the project construction will decrease the capacities from C_1, C_2 and C_3 to C^*_1, C^*_2, and C^*_3 during the construction periods of S_1 to F_1, S_2 to F_2, S_3 to F_3, respectively. After construction, the capacities are increased to the improvement levels of C'_1, C'_2, and C'_3.



(a)



(b)

Figure 11 Capacity Changes during the Simulation

Some questions worth considering include the following:

- How will the optimization result (project sequencing) be affected if we consider the construction time and capacity reduction?
- How will the demand react to the increasing delays due to project construction? Should a full equilibrium model, or partial equilibrium model, or elasticity model be applied?
- How will the optimization result be affected if demand is or is not sensitive to the capacity and resulting delays?
- How will the optimization results be affected in comparison with the rank of lock congestion level which might intuitively generate the schedule of lock improvement projects?

In order to consider project construction time and capacity reduction in SIMOPT, some modifications in the simulation model are made. With the implementation schedule calculated from the budget flow and project costs, projects are chronologically introduced into the simulation program and implemented immediately whenever the cumulative budgets reach the construction costs. In addition to the "start project" events in the original SIMOPT, "complete project" events are now added.

Some system variables are updated while the above two projects events are invoked. When an event of starting a project is invoked, lock capacity is reduced to its blockage level and the service rate decreases proportionally. At the same time, the completion time for project construction is calculated to determine when an event of completing a project will be invoked. Similarly, when an event of completing a project is invoked, lock capacity is increased to its expansion level as well as decreased the service time proportionally.

It is possible that the system might explode when a local capacity is reduced to zero or near zero during the construction time, if demand cannot respond to the level of service. In order to avoid infinite queues, an elastic demand model is involved during the simulation. That is, when an event of trip generation is invoked, the generation rate is updated based on the expected and real-time travel times. Let λ_{ij} denote current generation rate for a O_i / D_j pair, r_{ij} denote the annual growth rate and k_{ij} denote the demand elasticity. If the expected travel time is w_{ij} and real-time travel time is z_{ij}, the generation rate is updated as $\lambda_{ij} \cdot (1 + r_{ij})^{t_c - t_p} \cdot (z_{ij} / w_{ij})^{k_{ij}}$, where t_c is the clock time and t_p is the previous generation time.

If considering an alternate transportation mode, such as rail, to ease the possible traffic congestion due to the construction, full equilibrium or partial equilibrium models could be used. The shippers response to maintenance closures (i.e., capacity dropping to zero) when the railroads are the alternate mode to waterways has been modeled in Wang and Schonfeld's 2006 TRB paper. Based on those concepts, the reaction of traffic demand to capacity reductions could be similarly treated with an equilibrium model.

Project Multiplicity

At any specific lock site, several expansion alternatives with discretely specified capacities may be considered. Two cases may arise for project multiplicity: only one project among those alternatives can be selected, or multiple alternatives could be selected but implemented at different times over the planning period. The first case is straightforward and project costs for different alternatives are independent. However, the project costs in the second case could be interdependent and revised based on the implementation sequence. That is, project cost might include the construction cost for building the new project and deconstruction cost for removing the old project at the same location. In the current phase, the first case is considered with at most one project being selected among the alternatives at each site.

If there are mutually exclusive projects at the same location, i.e. if only one can be selected, we may consider the inclusion of sizing decisions in the project scheduling problem. While combining sizing and scheduling problem, the solution space of fully permutated sequences will be further enlarged through the inclusion of all project alternatives at each lock. That is, if there are N lock locations and m_i (i = 1,…, N) project alternatives for each lock, the total number of solution including all possible combinations and permutations would be $N!\prod_i m_i$. The project constraints must ensure that only one project at each location is selected among all available alternatives. Let X_i be a binary variable. If $X_i = 1$, the project is selected; if $X_i = 0$, the project is not selected. If i denotes the project alternatives, then the project constraints for any location can be formulated as $\sum_i X_i \leq 1$.

In order to consider project multiplicity, the definition of chromosome used in SIMOPT should be redefined or modified. Different ways to define a chromosome could represent the information about project multiplicity. One possible way of encoding project size and schedule is having both decision variables in the same sequence (as shown in Figure 12). That is, a new representation of sequence contains both lock ID (1, 2, 3… etc) and project alternative (A, B, C… etc).

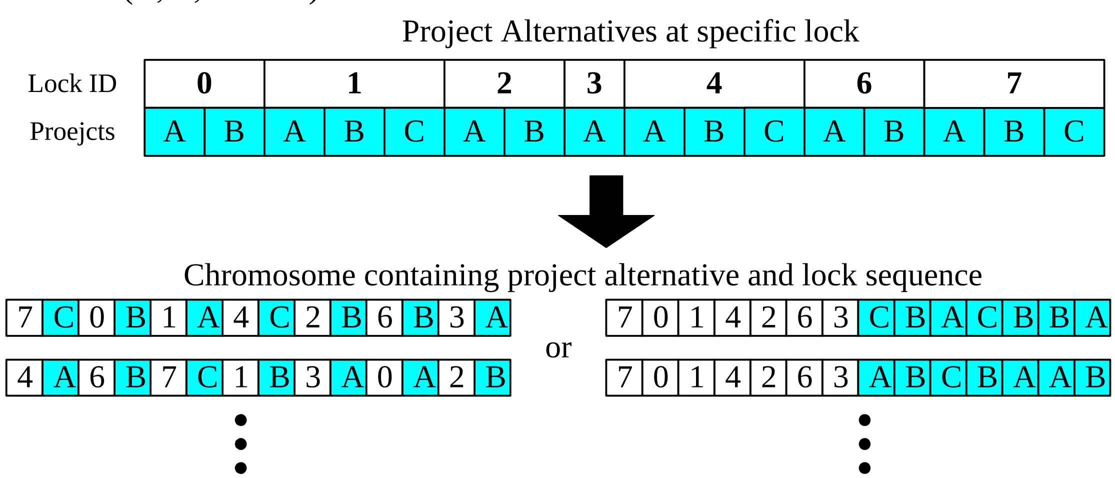


Figure 12 Paired Representation of Chromosome for Mutually Exclusive Projects

However, with this paired representation, both mutation and crossover operators must be redeveloped to avoid illegitimacy in the reproduction process, which creates offspring with invalid sequences or unavailable project alternatives. For example, as shown in Figure 13, an original EM operator developed in SIMOPT yields unavailable project alternative (that is, (a) some alternative is not available at some lock sites), or invalid sequence (that is, (b) unreasonable numbering sequence).

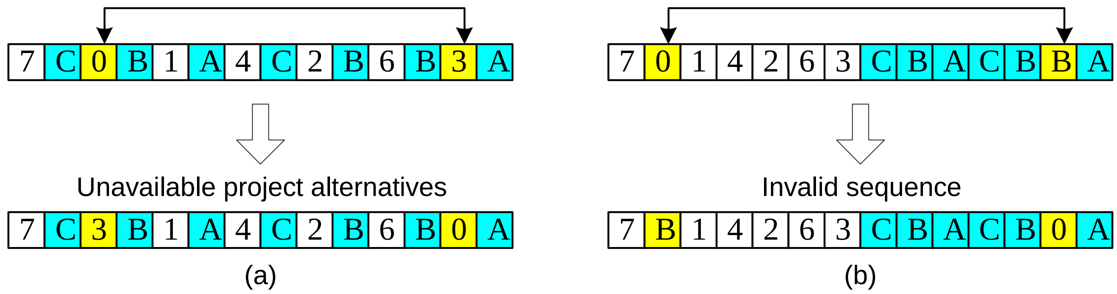


Figure 13 Illegitimacy Generated from Mutation Operator for Paired Representation

Therefore, a new EM operator should be able to swap the lock ID and project alternative together (as a pair) at the same time (as shown in Figure 14 (a)), or perform swapping twice for lock ID and project alternative with matching positions (as shown in Figure 14 (b)). It should also be able to randomize the project alternatives after any swapping (as shown in (c) and (d)). In other words, the genetic operators should be redesigned to be able to characterize legitimately the priority of project locations with corresponding project alternatives.

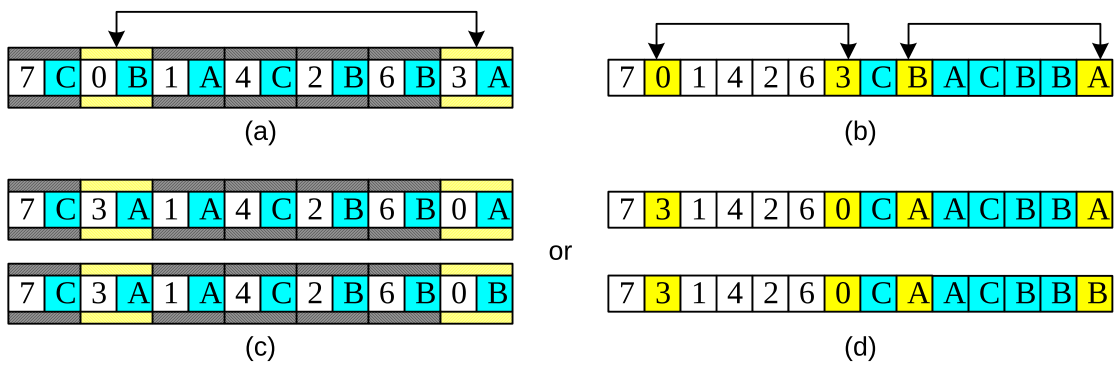


Figure 14 Possible Mutation Operator for Paired Representation

The other way of encoding these two variables together is keeping the same path representation used in SIMOPT but using project ID instead of lock ID in a sequence (as shown in Figure 15). With the original representation, the proposed GA operators in SIMOPT could still be applied on the mutation and crossover processes without any modification to produce the offspring. However, if considering only one alternative for each location, the sequences with full list projects are not the feasible solutions, in the sense that all alternatives will be implemented at different times (as shown in the middle part of the figure). Therefore, it is necessary to have a "refining" scheme embedded to create the feasible solutions for simulation evaluation. That is, instead of sequences with full lists of projects, a shorter sequence whose list of projects has only one project at each lock should be formed after the "refining" procedure (as shown in the lower part of Figure 15).

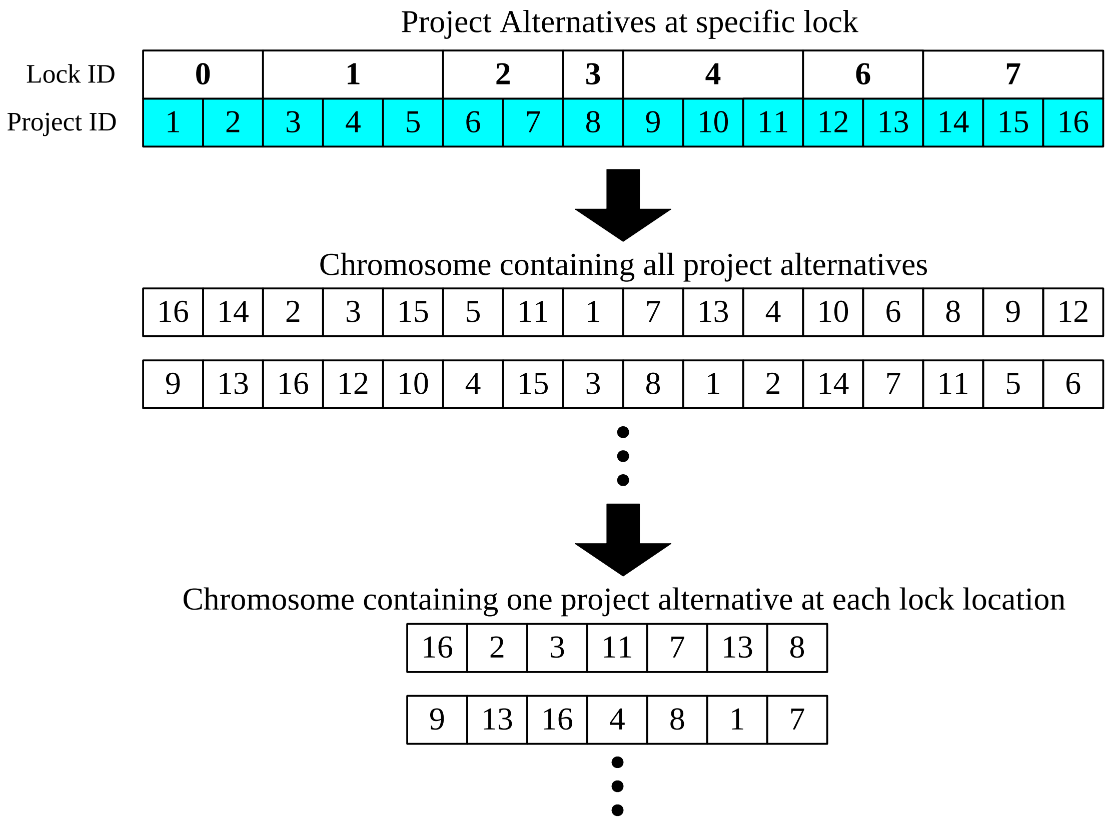


Figure 15 Path Representation of Chromosome for Mutually Exclusive Projects

The simplest way is to keep only one project at any lock and discarding the other projects at the same lock locations in any full-list sequence. As shown in Figure 16, whenever the first project alternative at one lock is selected, a "refining" technique will automatically discard the other project alternatives at the same lock. As noted, all the mutation and crossover operators are applied on the full-list chromosomes, not the refined chromosomes. Before starting any simulation evaluation, chromosome refining processes are performed on all produced offspring from any mutation or crossover operations.

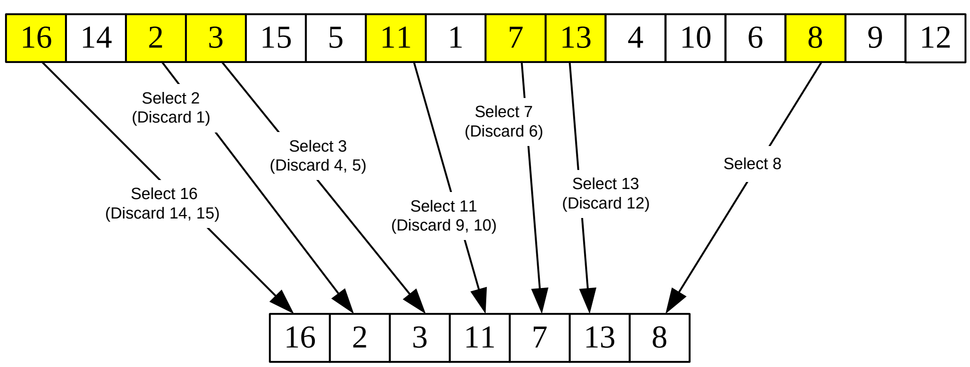


Figure 16 Proposed Refining Technique to Create Feasible Solutions for Mutually Exclusive Projects

In order to allow project multiplicity in SIMOPT, some modifications in the GA optimization model are made. In SIMOPT, the structure of the designed chromosome is shown in Figure 17. Each project initially includes information about project ID, project size and project cost. The project ID automatically indicates the project location. The implementation time for each project will be determined after the project sequence is generated and bounded with budget flow.

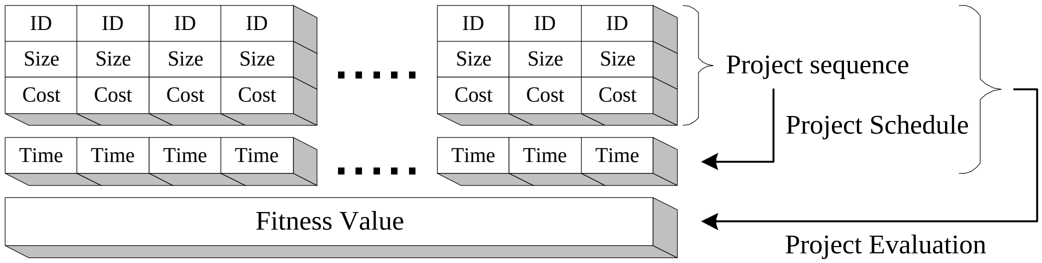


Figure 17 Structure of Chromosome Defined in SIMOPT

More information should be added into the chromosome definition when multiple alternatives are available at some lock locations. That is, in addition to project ID, lock ID should be provided (as shown in Figure 18, denoted as P.ID and L.ID). In this newly defined chromosome with multiple project alternatives per lock, lock ID is not unique anymore for each project but project ID is.

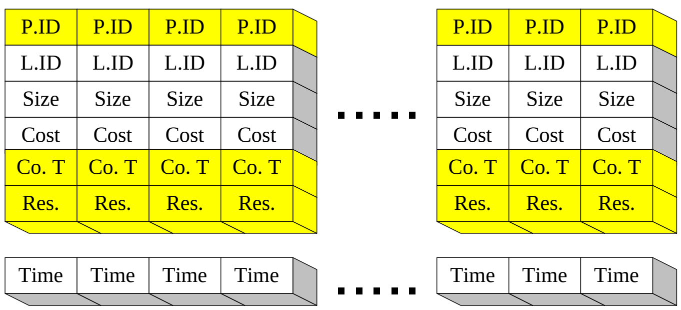


Figure 18 Modified Structure of Chromosome for Mutually Exclusive Projects

Time Efficiency

It is conceivable that some sequences that have been evaluated in previous generations are created again in the current generation. It is expected that combining the two stochastic processes of simulation and GA optimization will be time consuming. A significant time is required for evaluating each generated project sequence through simulation; especially numbers of replication is involved for variance reduction. Therefore, in order to reduce optimization search time, avoidance of duplicate simulation runs is considered.

A genetic approach is usually based on a memoryless evolutionary procedure. In contrast, another meta-heuristic approach called tabu search is designed with an adaptive memory which records solutions visited during the search. With this feature, the implementation of procedures can search the solution space economically and effectively. Thus inspired by the idea provided in tabu search, the proposed GA is further modified as memorized evolutionary model. That is, the evaluated solutions in GA optimization are memorized in each generation. With the intension of avoiding duplications, it is a key step to search through memorized solutions before performing the simulation.

In order to avoid re-evaluating the same project sequences, each evaluated sequence and its evaluation results are recorded in a *deque* (short for "double-ended-queue") data structure. If the newly generated sequence appears in the recorded solution pool, its evaluation result is directly assigned from memory rather than re-obtained simulation. Later, all newly generated sequences are pre-screened to identify those previously simulated ones before any time-consuming simulation is performed. Compared with the time for multiple simulation runs, it would be still worthwhile to spend time on checking throughout the recorded sequences.

In a *deque* data structure, the length of list is unlimited. It is also not necessary to declare a bulky memory space as for array data structure before starting the optimization process. Since a *deque* structure provides rapid insertions and deletions at its front or back of the structure, it is easy to add any newly evaluated solution onto the end of list. It also allows direct access to any stored element. Whenever an evaluated sequence has been found as the same sequence with the one being going to be evaluated, the stored fitness value can be directly assigned to the fitness result instead of duplicating simulation runs.

Thus, as shown in Figure 19, the evaluated sequences are stored in a *deque* and each element contains information about project sequences and their fitness values. During pre-screening, a newly generated sequence is compared with the recorded sequences, a "solution list". As long as an exact sequence is found in the solution list, the recorded fitness value is directly assigned to the new generated sequence and the simulation evaluation is skipped. If no exact match is found among previously evaluated sequences, the new sequence is simulated and added into the solution list with its newly evaluated fitness value.

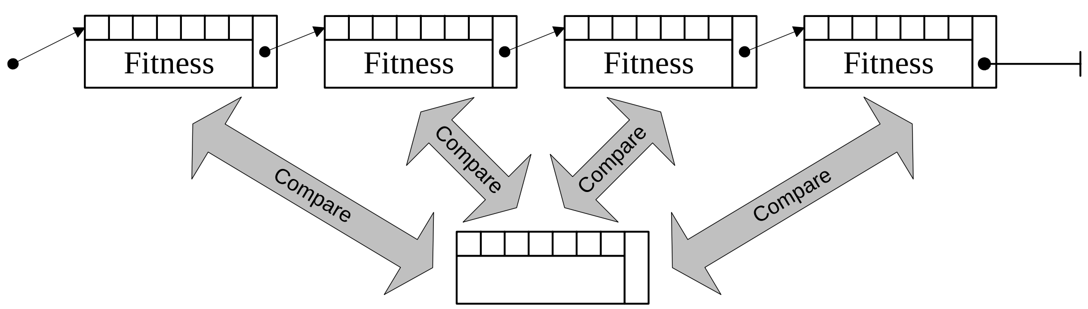


Figure 19 Deque Data Structure

The comparison between two sequences is performed project by project. The sequence comparing process is stopped whenever any of project elements is found different (as shown in Figure 20).

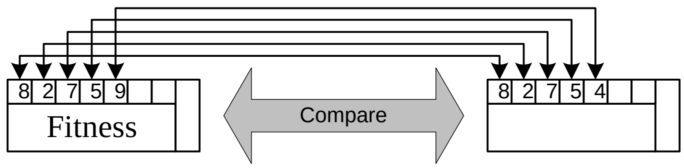


Figure 20 Sequence Comparison

It is noted that the comparison of sequences is straightforward if there are no mutually exclusive projects, since the full list of projects is the same as the full list of lock locations. However, with mutually exclusive projects, the comparison results could be different. Two types of sequences are created when considering mutually exclusive projects. Full sequences of project alternatives are generated from the offspring production process. Partial sequences with only one project per lock are "refined" for evaluation by simulation. To avoid duplication in the evaluation process, we should compare the refined partial sequences, rather than the full sequences. That is, as shown in Figure 21, after the "refining" process (performed in the case of mutually exclusive projects), different full sequences of project alternatives could become the same partial sequence with only one project per lock.

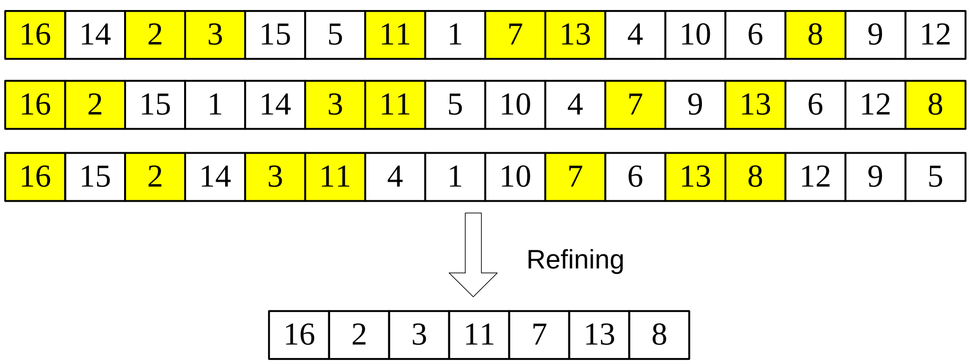


Figure 21 Refined Sequence

Therefore, in order to save simulation time even efficiently, it is better to have a solution list recording the "refined" partial sequences rather than the "original" full sequences.

Model Test (Enhanced SIMOPT)

Test Network

The test network used in SIMOPT demonstration is used here for testing any enhanced GA techniques proposed in this phase (as shown in Figure 22). 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 (refer to the "SIMOPT" presentation, July 2005). Not all locks require improvement projects, but all improvement projects are at real locks.

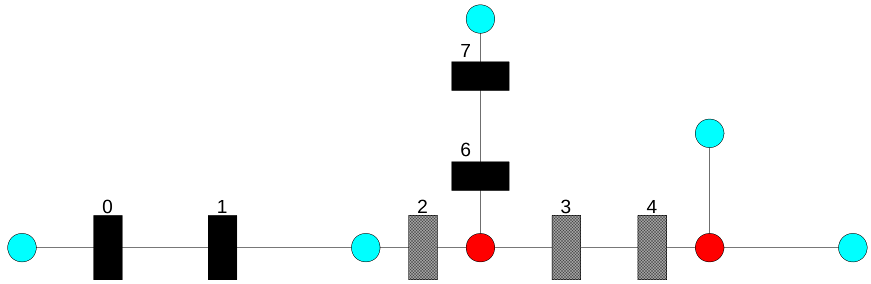


Figure 22 Test Network for SIMOPT Extension

Input Parameters

Input Statistics

- O/D matrix with trip generation rates
- Tow size distributions
- Chamber service time distributions
- Speed distributions

Lock Operation

- FIFO control
- Towboats priority (tow w/o barge)
- Lockage cuts
    - Main: always 1 cut
    - Auxiliary: 2 cuts for tows with more than 9 barges
- Chamber assignment
- Chamber bias (main chamber is preferred for tows with more than 7 barges)

Demand

- Annual growth rates for each O/D pair
- Elasticity for each OD pair

Base Case Run

- Lock congestion level (from highest V/C to lowest V/C): 7→1→6→0→2→4→3
- Average O/D travel time

System Parameters

- Simulation parameters

Table 5 Simulation Parameters in SIMOPT Extension

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

- Optimization parameters

Table 6 Optimization Parameters in SIMOPT Extension

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

Testing Results

All the test results are presented with three cases obtained with the recently modified SIMOPT: (1) Considering construction times, (2) Considering mutually exclusive projects, and (3) Avoiding duplicated simulation runs. In those test cases, it is assumed that the project construction starts at the time when required cost is accumulated. The current objective function is set to minimize the total cost which includes system total delay cost (barge-minute) and project construction cost. All the cases are run on a Pentium III machine with 3.6 GHz CPU and 1GB memory.

Case 1: Considering Construction Times

In this case, only one project is considered at each single lock. Project information is detailed in blockage duration for the construction and capacity reduction ratio during the construction time as well as project size (capacity expansion ratio) and project cost. Two scenarios are proposed. One (case 1.1) serves as the base case in which construction time is neglected, as in the original SIMOPT. (The implicit assumption is that construction is instantaneous.) The other (case 1.2) considers construction time and its relevant effects such as capacity reduction and demand response.

Inputs of Lock Improvement Projects

- Project ID
- Lock ID
- Project size – capacity expansion ratio
- Project cost ($ M)

- Project duration – construction time(year)
- Project blockage – residual capacity ratio

Table 7 Project Information for Case 1.1 (Baseline without Construction Times)

Project ID	Lock ID	Size	Cost
1	7	2.0	17
2	1	2.0	16
3	6	2.0	23
4	0	2.0	19
5	2	1.1	22
6	4	1.3	21
7	3	1.1	25

Table 8 Project Information for Case 1.2 (Considering Construction Times)

Project ID	Lock ID	Size	Cost	Duration	Blockage
1	7	2.0	17	0.17	0.2
2	1	2.0	16	0.09	0.5
3	6	2.0	23	0.12	1.0
4	0	2.0	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

Optimized Project Sequences and Implementation Schedules

Since there are 7 projects to be sequenced in Table 7 or Table 8, the solution space is 7! = 5,040. For testing purposes, this is not a huge number. The optimized project sequences and their implementation schedules are shown in Table 9 and Table 10. The optimized results are quite different for the two scenarios. While considering construction time and capacity reduction, the total cost increases considerably due to increasing traffic delays during the construction period. That is, inclusion of construction time and the capacity reduction during construction in the simulation is important and significantly affects the optimization results.

Table 9 Optimized Results for Case 1 (Considering Construction Times)

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

Table 10 Additional Optimized Results for Case 1 (Considering Construction Times)

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 Search Performance

Based on case 1.2, Figure 23 shows as an example of GA search performance. The best sequence in each generation is always saved, so the solution can never get worse over successive generations. However, the rate of improvement decreases over successive generations until further improvement become very unlikely. From the first generation to the termination, there is an approximately 60% improvement in the optimized solutions.

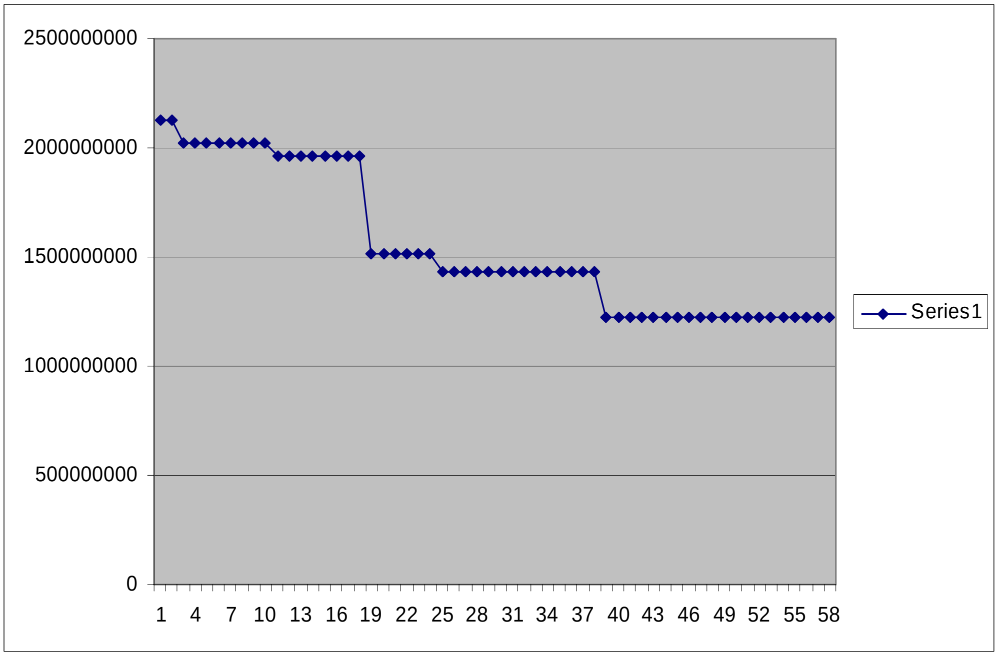


Figure 23 GA Search Performance

Case 2: Considering Mutually Exclusive Projects

In this case, 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. Construction time and capacity reduction during the construction period are considered. Similarly, two scenarios (case 2.1 and case 2.2) are proposed for case 2: with or without considering mutually exclusive projects. Case 2.1 is actually the previous case 1.2.

Inputs of Lock Improvement Projects

- Project ID
- Lock ID
- Project size – capacity expansion ratio
- Project cost ($ M)
- Project duration – construction time(year)
- Project blockage – residual capacity ratio

Table 11 Project Information for Case 2 (Considering Mutually Exclusive Projects)

Project ID	Lock ID	Size	Cost	Duration	Blockage
1	7	1.5	10	0.10	1.0
2	7	1.8	13	0.13	0.8
3	7	2.0	17	0.17	0.2
4	1	1.5	10	0.05	1.0
5	1	2.0	16	0.09	0.5
6	6	2.0	23	0.12	1.0
7	0	1.5	15	0.10	1.0
8	0	2.0	19	0.11	0.5
9	2	1.1	22	0.03	0.8
10	4	1.1	15	0.01	1.0
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

Optimized Project Sequences and Implementation Scheduled

Here, there are 13 projects: 3 alternatives at lock #7, 2 alternatives at lock #1, 1 alternative at lock #6, 2 alternatives at lock #0, 2 alternatives at lock #2, 3 alternatives at lock #4, and 1 alternative at lock #3. The solution space is $7! \times 3! \times 2! \times 2! \times 3! =$ 725,760. That is, much less than $13! =$ 6,227,020,800. The optimized project sequences and implementation schedules are shown in following tables.

Table 12 Optimized Results for Case 2 (Considering Mutually Exclusive Projects)

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

Table 13 Additional Optimized Results for Case 2 (Considering Mutually Exclusive Projects)

w/o Mutually Exclusive Projects					w/ Mutually Exclusive Projects			
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			

Case 3: Avoiding Duplicated Simulation Runs

In this case, newly produced sequences are prescreened to avoid re-simulating previous ones. Therefore, two scenarios are proposed to compare the differences of required genetic search times.

The first scenario serves as base case without any pre-screening action for the evaluated solutions before the simulation. The second scenario considers the pre-screening process to avoid duplicated simulation runs, but may require some search time in the pre-screening process. In order to perform the pre-screening process, the search comparison is conducted after a full list of projects is refined as a feasible sequence, in which only one project is selected at each lock. Instead of comparing sequences whenever a full list of project alternatives is generated, this will eliminate all the possible simulation duplications, since different full lists of sequences might result in the same project lists after the “refining” procedure.

Computation Times for Optimization Search

Most inputs in this case are the same as in the second case. In order to generate more varieties, the population size is increased to 100 in this case. Search time for pre-screening process is expected to increase when the number of recorded solutions increases. After generations in GA’s, the number of recorded solutions could be so large that considerable time is spent searching through the whole list for sequence comparison. The additional “solution search process” might reduce the time-saving effect from the pre-screening step. However, in a simulation-based optimization model, the pre-screening time seems negligible compared to the time for multiple simulation replications. The optimized solution found in this case is shown as project sequence 4→7→1→6→10→13→9 with total cost of $342,086,655. This result differs slightly from the result in case 2.2 due to some changes in input parameters, such as the population size of 100.

Comparative results for GA search time are shown in Table 14. With pre-screening, the GA search time decreases by approximately 20%. If the number of generations increases, time savings from pre-screening should increase.

Table 14 Results for Case 3 (Avoiding Duplicated Simulation Runs)

Construction Time / Capacity Reduction	Mutually Exclusive Project	Pre-screening Solutions	# of Generations	GA Search Time (sec)
YES	YES	NO	21	129641
YES	YES	YES (refined list)	21	104604

Verification of GA Optimization Model

In such a complex combinatorial problem, it is not easy to find the exact optimal solution; at least no existing methods can guarantee finding the global minimum. Verifying the goodness of the solution optimized by the proposed algorithm is also difficult. Therefore, in order to statistically test the effectiveness of the algorithm, an experiment is designed to evaluate 20,000 randomly generated solutions to the problem with a sampling process.

Using case 2.2 with mutually exclusive projects as an example, the solution space contains 725,760 (= 7! × 3! × 2! × 2! × 3!) solutions. 20,000 solutions cover approximately 3% of the solution space. From those observations, the best fitness value in this sample is 0.34521×10^9, while the worst one is 9.8882×10^9. The sample mean is 2.3769×10^9 and the standard deviation is 1.7497×10^9.

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. The distribution for those 20,000 sampled solutions is shown in Figure 24 with different histogram scalars, namely 20 and 100. From Figure 24(a), there is one higher peak around value of 1.0×10^9 and one lower peak around value of 5.5×10^9. From Figure 24 (b), two higher peaks around the values of 0.5×10^9 and 1.2×10^9 can be observed. Based on the plotted histograms, the best fitting distribution with uneven bell shape might be the gamma distribution or the lognormal distribution.

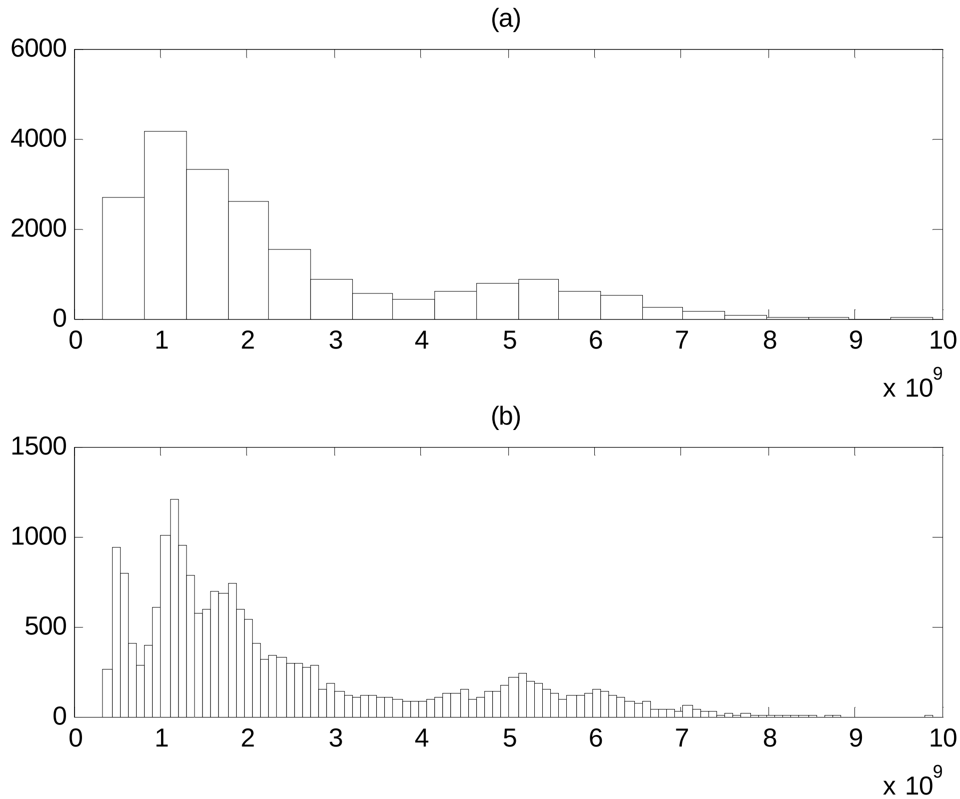


Figure 24 Histograms of Sampled Solutions

Figure 25 shows those 20,000 sample solutions fitted with gamma distribution, gamma (α, β), in which α and β are the shape and scale parameters, and lognormal distribution, LN (μ, σ^2), in which μ and σ^2 are sample mean variance. The values of α and β for the fitted gamma distribution are 2.0757 and 1.1451×10^9, and the values of μ and σ^2 for fitted lognormal distribution are 21.3292 and 0.7307, respectively. As can be seen, there is a large "spike" close to $x = 0$, which is covered better by the lognormal distribution.

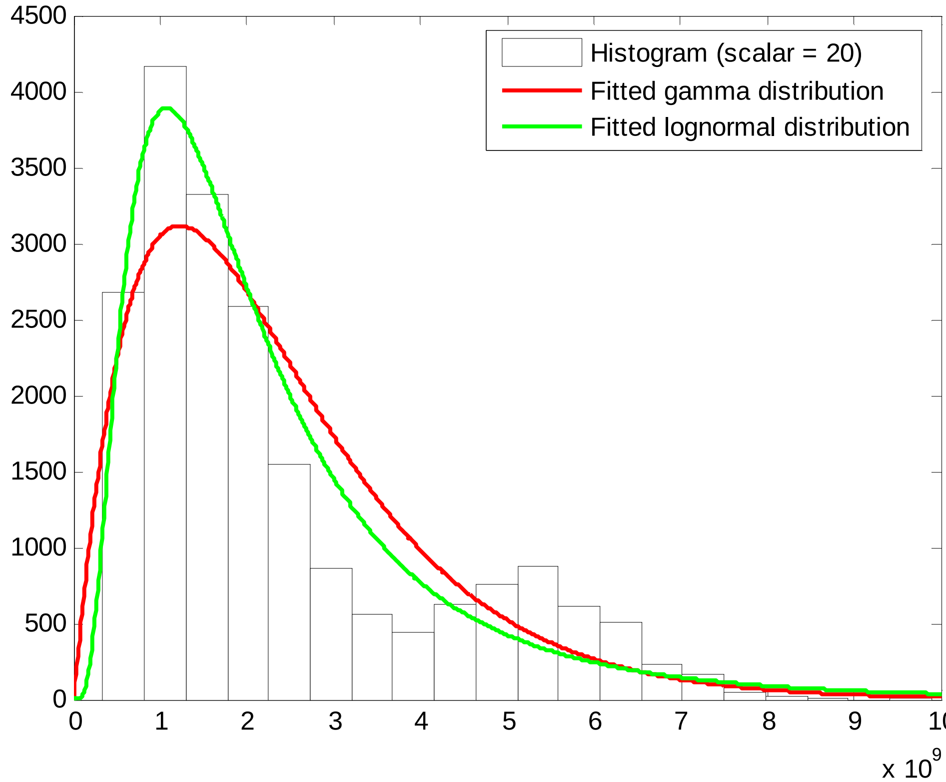


Figure 25 Fitted Gamma and Lognormal Distributions

Compared with the optimized solutions found in case 2 (best fitness value = 0.3449×10^9) and case 3 (best fitness value = 0.3420×10^9), the best solution with fitness value of 0.34521×10^9 is approximately 0.1% higher (i.e. worse) than the GA search results. That is, the optimized solution found in proposed GA search still 0.1% less than the best solution found from the random search experiment. Those optimized solutions, though not necessarily optimal, are still very good when compared with other random solutions in the solution space. That practically shows the reliability and validity of the proposed search algorithm.

Conclusions and Future Work

Summary and Conclusions

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. Thus, using the SIMOPT testbed, enhancements of the simulation-based optimization models are developed and tested.

When considering project construction time and capacity reduction during the construction, the "events" of starting and completing the projects are defined to update the system capacity during the simulation. The simulation model also considers the possibility of queue "explosion" if lock capacity decreases significantly during construction periods. Traffic demand is thus designed to be sensitive to the service level and adjusted automatically during the trip generation. For the optimization model, extra project information related to construction is added into the GA chromosome. Results show how the construction time and associated capacity reduction significantly affect the optimized sequence.

When considering mutually exclusive projects, the GA chromosome definition should be modified. In order to apply the same genetic operators developed in SIMOPT, the newly defined chromosome contains a full list of mutually exclusive projects. However, solutions with full lists of projects are not feasible when we allow at most one project per lock. Therefore, a "refining" technique is applied to create feasible solutions with lists of projects having at most one project per lock. The modified SIMOPT is able to solve 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. By avoiding duplicated simulation runs, the test case shows that the optimization search time is reduced by approximately 20% over 21 generations. Even larger percentage reductions are expected if the number of generations is increased.

At the end, a verification process is conducted to show the validity and reliability of the proposed GA search algorithm. Random solutions are generated from a sampling process and fitted with gamma or lognormal distributions. Compared with the those generated solutions, the optimized results found by the proposed GA search algorithm are still 0.1% better than the minimum value found from the random search experiment.

Future Work

A key component of NaSS is the investment optimization module, which is currently tested with Genetic Algorithm (GA) optimization. This investment optimization module is used to identify project modifications that are worthy of implementation, their order of implementation, and optimal implementation timing. There are good reasons for choosing GA instead of other optimization algorithms. First, GA's provide great flexibility for creative ideas, for example in the selection method, mutation/crossover rules, problem specific operators, and immigration and replacement between generations. Secondly, GA's are naturally suitable for running on parallel processors. With parallel computing, the optimization time could be significantly reduced. Also, the GA's developed for network-level optimization also seem adaptable for optimizing lock-level enhancement projects.

Additional ways of enhancing the GA optimization algorithms are available. The scope of work for the next phase includes additional development of genetic algorithms, and their application to project selection, sequencing and scheduling.

In GA's, search performance could possibly be affected by the mix of different genetic operators. To exploit the problem structure, some "smart" operators might be created specifically for waterway project scheduling. Some prescreening rules could also be developed to avoid simulating solutions that are unpromising or violate constraints.

Since the optimization model can be developed separately from the network simulation model, it is possible to integrate them with a simple "evaluator" to save the time in running simulation-based optimization. The simple evaluator could be any approximate simulator or even an algebraic function.

In the problem of project selection, sequencing and scheduling, additional complexities may arise, such as multiple alternatives at the same location which may be implemented at different times, project precedence relations, further budget constraints (e.g. regional limits, new construction vs. maintenance), budgets related to taxes on traffic levels found during simulations, and tradeoffs between construction times and costs. Such complexities could all be addressed in future model developments.

Appendix

GA Phase 1 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

Task 2 Evaluation / Simulation model

> 2.1 Store results and prescreen alternatives to avoid repeated simulation near previous searches

Task 3 Project selection / sequencing / scheduling

> 3.1 Include construction time during simulation
>
> 3.2 Consider capacity reduction during construction period
>
> 3.3 Consider multiple alternatives at the same location / mutually exclusive projects
>
> 3.4 Consider optimal timing for projects absent budget constraints

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.

References

Wang, S. “Simulation and Optimization of Interdependent Waterway Improvement Projects”, PhD Dissertation, University of Maryland, 2001.

Wang, S. and Schonfeld, P. “Development of Waterway Simulation Model”, Annual TRB Meeting, Jan. 2002 (02-2194 on Conference CD-ROM).

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

Wang, S. and Schonfeld, P. “SIMOPT”, presented in July 2005 at Fort Belvoir, VA.

Wang, S. and Schonfeld, P. “Modeling Shipper Response to Scheduled Waterway Lock Closures”, Annual TRB Meeting Jan. 2006 (06-1833 on CD-ROM).

Wang, S. and Schonfeld, P., “Genetic Algorithms for Selecting and Scheduling Waterway Projects”, presented at NETS SYMPOSIUM, Salt Lake City, Jan. 2006.