Genetic Algorithms for Selecting and Scheduling Waterway Projects

(Phase 3 Draft Report)

By Shiaaulir Wang, Ning Yang and Paul Schonfeld

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

June 15, 2007

Table of Contents

List of Tables

List of Figures

Abstract

A testbed waterway model (SIMOPT) that combines simulation and optimization has been developed for the Navigation, Economics Technologies (NETS) Program 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 presented in the first phase of GA optimization work completed in April 2006. In order to enhance both the search efficiency of the optimization model and the capabilities for imposing additional constraints, some improvements in investment optimization methods are implemented and tested on SIMOPT.

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. Some enhancements were completed in previous two phases. In this phase, the improvements in the investment model include (1) allowing multiple projects at the same location with different implementation times, (2) considering project construction times and capacity reductions during the construction period with elastic demand responding to the delays, (3) considering tradeoffs between construction time and cost, and (4) network-level and lock component-level maintenance planning and scheduling. With elastic demand, the optimization problem should be to maximize net benefits rather than minimize total costs. Additionally, in order to speed up the optimization process, the feasibility of applying parallel computing is investigated and tested.

Introduction

The U. S. Army Corps of Engineers (USACE) has considerable interest in the problem of selecting, sequencing and scheduling waterway improvement projects. When numerous projects are considered, a massive combinatorial optimization problem results. An investment optimization model based on genetic search algorithms is applied to solve this large and complex combinatorial problem in the SIMOPT, simulation–based optimization model.

In previous phases of this study (Wang and Schonfeld, 2006 NETS Reports Phase I and II), project construction time and capacity reductions during construction were introduced in the SIMOPT model. Mutually exclusive projects at locks are specified in that analysis. Constraints addressing lock precedence relations and regional budgets were also included in the search process. In order to reduce the computation time, the evaluated solutions were recorded to avoid re-simulating them during the genetic search.

The following sections focus on the allowing multiple projects with different implementation times at the same lock location, and considering possible capacity reductions and resulting demand changes during the project construction periods. When considering of capacity reduction during the construction period, the issue of elastic demand response delays (due to partial or full lock closure) arises. In the simulation model, elastic demand (or demand response) can be handled in the trip generation module. Tradeoffs between construction time and cost are also investigated. Since traffic demand and benefits may be significantly affected by the decisions being simulated, it is unreasonable to evaluate or optimize the system merely based on total costs. Benefits to waterway users should be estimated during simulation runs while accounting for the users' responses to lock closures which might significantly affect the travel times.

Enhanced Features in Project Scheduling Problems

According to the Scope of Work drafted for GA enhancement (see Appendix), several tasks are included in phase III, including specifying multiple projects with different implementation times at the same location, considering construction time and capacity reduction with demand elastically responding to delays, and analyzing the tradeoffs between construction times and costs. The option of applying parallel computing in GA optimization is also explored. With multiple processors working on simulation-based evaluations, the time required in the proposed GA search can be reduced in nearly inverse proportion to the number of processors used.

Multiple Projects with Different Implementation Times at the Same Lock

At any specific lock site, several improvement projects or expansion alternatives with discretely specified capacities may be considered. Those projects might be independent of each other, but might also be dependent with interrelated costs.

When considering different expansion projects, two cases project multiplicity may arise. In the case of mutually exclusive projects, only one project among those alternatives at each location could be selected. This one is straightforward since project costs for different alternatives are independent. As been has discussed in the first phase of GA enhancement work, the sequencing problem then determines project timing and size simultaneously.

With non-exclusive multiple projects, several alternatives could be selected for one site but implemented at different times over the planning period. Those alternatives include independent improvement projects as well as dependent expansion projects. There are no cost relations among those independent improvement projects. However, for different expansion projects, if implemented at different times, the project costs and sequence should be carefully defined. For example, expansion project A which increases capacity from a baseline based on capacity expansion ratio of 1.5 should be implemented (if ever) ahead of expansion project B which increases capacity based on capacity expansion ratio of 2. It makes no sense that project B is implemented ahead of project A. In addition, with project precedence, the cost of project B is not the construction cost of project B, but a conditional cost based on the implementation of project A. For example, if \$0.5 million is needed for project A, an extra cost of \$0.8 million is needed for project B after implementing project A.

For different expansion alternatives at different times, a precedence constraint can be applied to restrict the sequence of those expansion projects. Lock precedence constraints have been discussed in the second phase. 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. In this report, project precedence constraints are considered. At the same lock location, 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. That is, given an array of integers $\{x_i\}$ where *i = 1, 2, 3, …, n*, and *n* is the number of projects, each element in the array represents the scheduled order of one project. The precedence constraint can be formulated as $x_i < x_j$.

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 at great expense. 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,…, 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.

Construction Time and Capacity Reduction

Project construction times have been analyzed in the first phase report (Wang and Schonfeld, 2006) with a conservative assumption that the project construction starts when the funding required for the project is accumulated. However, construction can usually be started whenever there is available budget. Since projects are funded one at a time, their construction periods may actually overlap the funding periods of subsequent project, as shown in Figure 1.

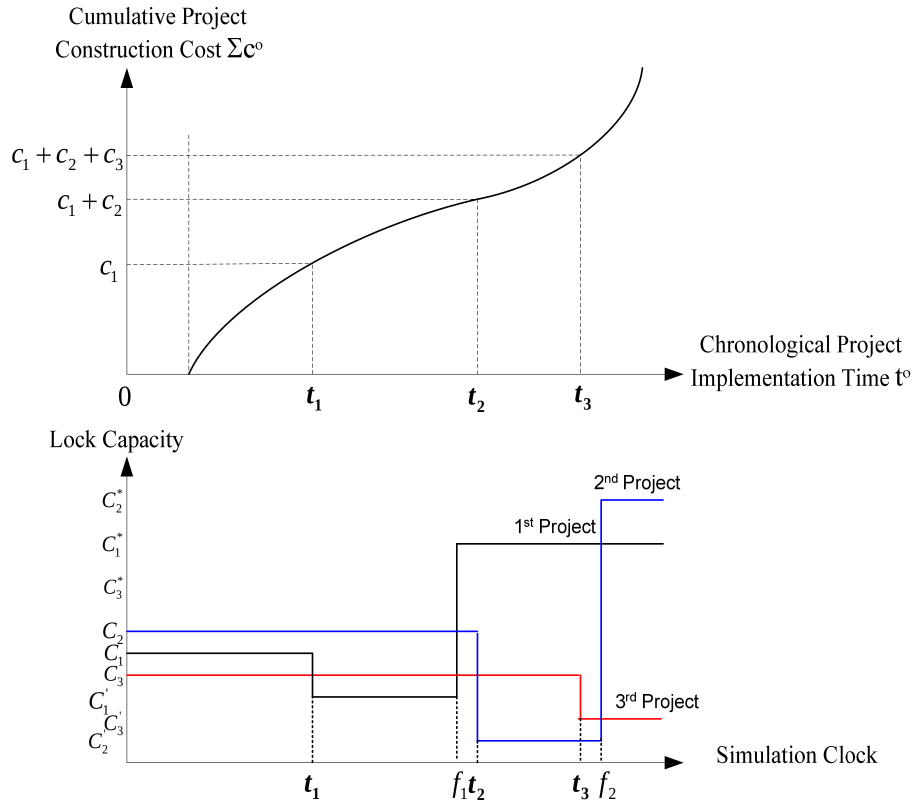


Figure 1 Project Funding Accumulation and Construction Time

However, in the real word, given a budget distributed over time that suffices for several projects, project timing is determined by available funding. Financially, it is desirable to avoid overlaps over time in funding different projects. The proposed model assumes the projects are completed chronologically at the time when sufficient budget is accumulated to cover their construction cost. Additionally, constructability concerns may result in construction overlap when project construction time exceeds than the time required for funding the project. Therefore, if construction overlaps exist, projects become operational after their construction is completed even if the project funding has been fully accumulated earlier. The closure time for construction should be as short as possible. If there are no construction overlaps, projects are started early enough to be completed by the time project funding is fully spent.

An example is shown in Figure 2. Three lock improvement projects are prioritized to increase lock capacities from C_1, C_2, and C_3 to C_1^*, C_2^*, and C_3^*, respectively. The project costs are c_1, c_2, and c_3; the implementation and completion times are t_1, t_2, t_3 and f_1, f_2, f_3, respectively. Figure 2 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 T_1, T_2, and T_3, respectively. After construction, the capacities are increased to the improved levels C_1^*, C_2^*, and C_3^*, respectively. The time between 0 to t_1 is the system warm-up time. If construction overlaps exists, the project operation time $f_i = \max\{t_i + T_i, t_{i+1}\}$; the project implementation time $t_i = \max\{t_{i+1} - T_i, t_i\}$. There are overlaps of construction times T_1, T_2, and T_3, but no overlaps of project funding times B_1, B_2, and B_3.

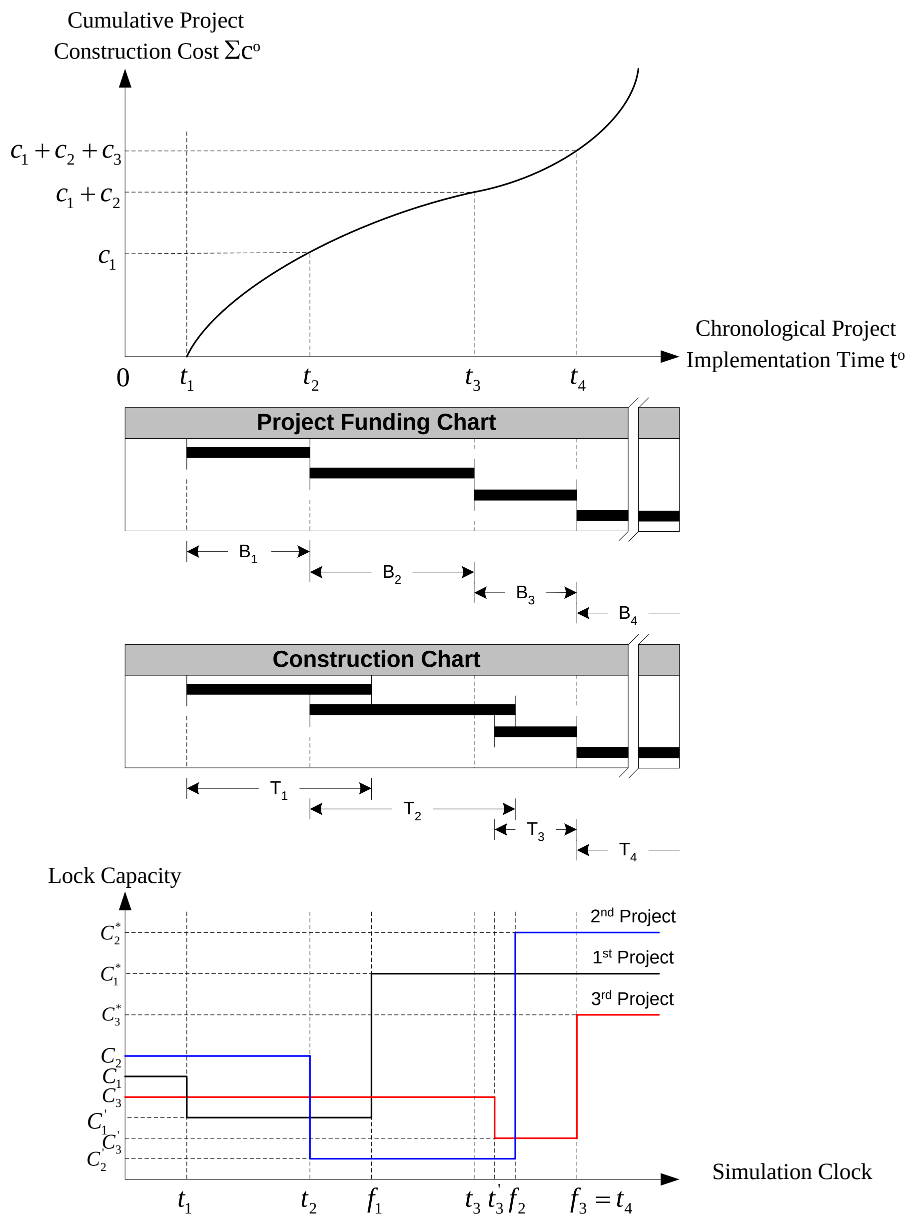


Figure 2 Relations of Project Schedule and Construction Time

Tradeoffs between Construction Times and Costs

For the general construction activities, there may be tradeoffs between construction time and costs. Such time-cost tradeoff relations should be identified and quantified. Various possible combinations of project duration and costs resulting from different procedures and/or resource combination should be considered. From Tien and Schonfeld (2006), the most efficient combinations, such as *a*, *b*, *c* or *d*, define an efficient frontier and are superior in time T or cost C (or both) to any point above that frontier, as shown in Figure 3. For a project with numerous component activities, the tradeoff frontier may be determined using heuristics and mathematical programming techniques, after analyzing resource relations with the critical path method. Thus, a time-cost curve could also be approximated, continuously or discretely, to indicate the minimum cost required to speed-up a project to some degree.

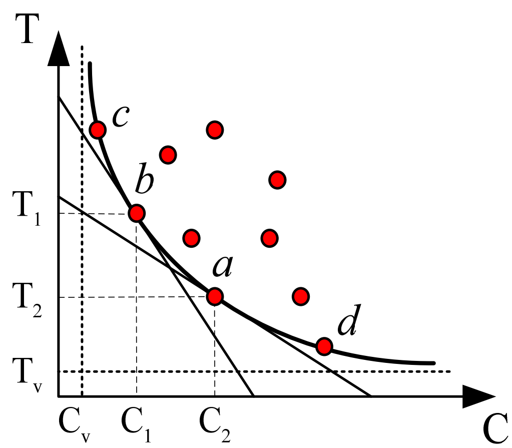


Figure 3 Time-Cost Tradeoff Relation

In a long-term investment planning problem, with a given budget distributed over time, the difference among nearby combination points of time and cost on the efficient frontier curve might not be very significant when an overall evaluation for a large network is pursued. In addition, investment planning tends to determine the best selection, sequence and schedule for candidate improvement projects. When considering a continuous relation between construction time and cost for each single project, it takes more efforts to represent the decision variables since any change between construction time and cost in one project may change the project sequence, and change the solution. Therefore, it is preferable to formulate the time-cost tradeoff relation discretely for several alternative time-cost combinations.

In order to provide alternative time-cost combinations for each project, a tradeoff constraint is considered. Since the tradeoff constraint intuitively provides one combination between two tradeoff variables, it has features of mutually exclusive constraints. That is, with mutually exclusive time-cost combinations for one project, i.e. if only one can be selected, we may consider the inclusion of construction time and cost decisions in the project scheduling problem. When combining construction time, cost, 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, m_i (i = 1,…, N) project alternatives, and l_j (j = 1,…, $\sum_i m_i$) time-cost combinations for each project, the total number of solution including all possible combinations and permutations would be $N! \cdot \prod_i m_i \cdot \prod_j l_j$. The tradeoff constraint must ensure that only one combination for each project is selected among all available alternatives. Let X_k be a binary variable. If $X_k = 1$, the time-cost combination alternative is selected; if $X_j = 0$, the time-cost combination is not selected. If k denotes the combination alternatives, then the tradeoff constraints for any project can be formulated as $\sum_{k} X_k \leq 1$.

As discussed in the first phase (Wang and Schonfeld, 2006), a chromosome used for mutually exclusive projects can be represented with a project ID and a tradeoff alternative ID (as shown in Figure 4). Each project has several discrete time-cost tradeoff alternatives (e.g., project #1, #2, #3…, etc. have 2, 3, 3…, etc. time-cost tradeoff alternatives, respectively.). With this chromosome representation, the proposed GA operators in SIMOPT could still be applied on the mutation and crossover processes without any modification to produce the offspring. As noted in Figure 4, the sequences with full list tradeoffs for all projects are not feasible solutions (as shown in the middle part of the figure). Since only one tradeoff alternative will be included in the implementation sequence, a "refining" scheme embedded to create the feasible solutions is required for simulation evaluation. Thus, 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 4).

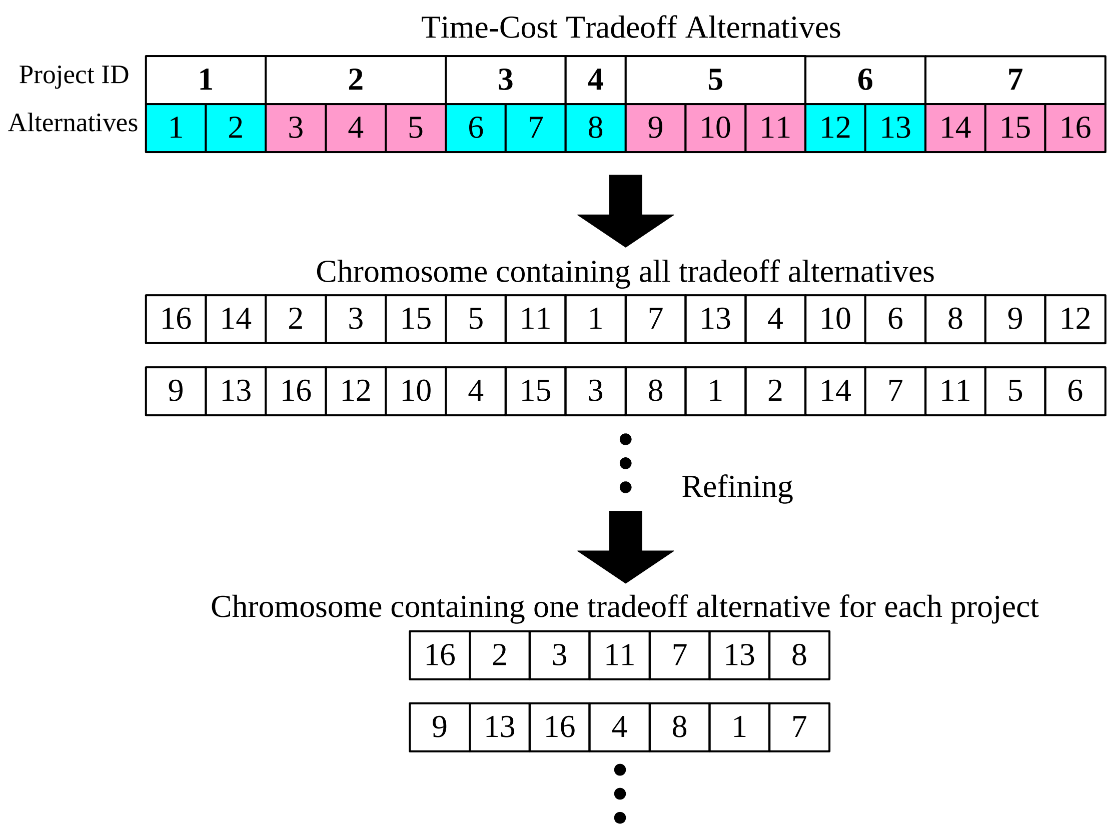


Figure 4 Paired Representation of Chromosome for Mutually Exclusive Projects

A similar "refining" technique (discussed in Wang and Schonfeld, 2006) is applied to discard the other tradeoff alternatives for the same project. As known, all the mutation and crossover operators are applied on the full-list chromosomes, rather than the refined chromosomes. Before starting any simulation evaluation, chromosome refining processes are performed on all offspring produced from any mutation or crossover operations.

While considering mutually exclusive time-cost tradeoff alternatives, more information is added into the chromosome definition. As shown in the top of Figure 5, any gene in a sequence chromosome includes alternative ID, project ID, and lock ID (seen as A.ID, P.ID, L.ID, respectively). Among the IDs, only the alternative ID used here is unique for genes (as shown at the bottom of Figure 5).

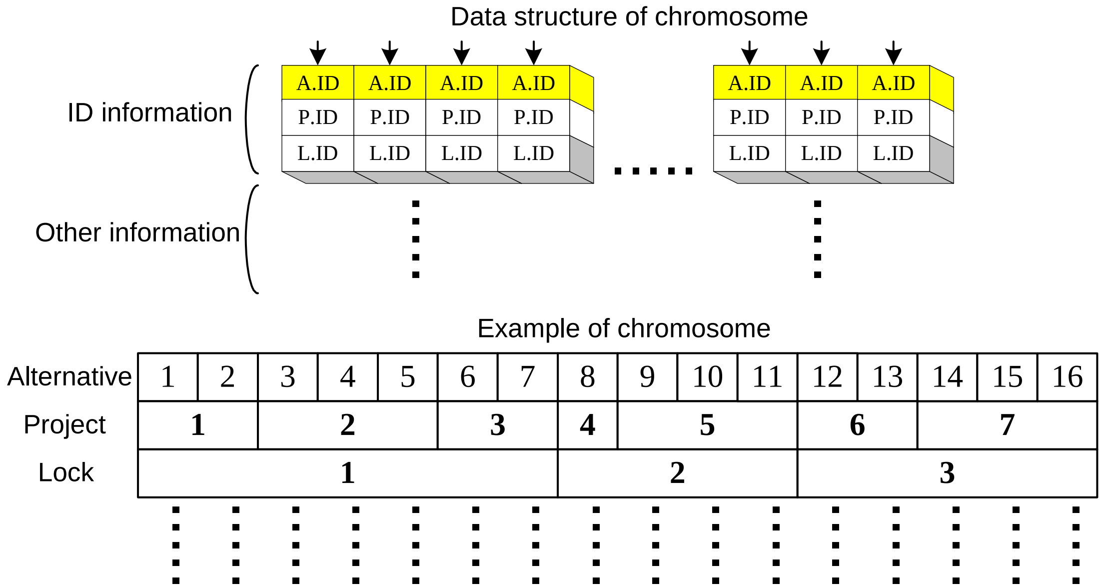
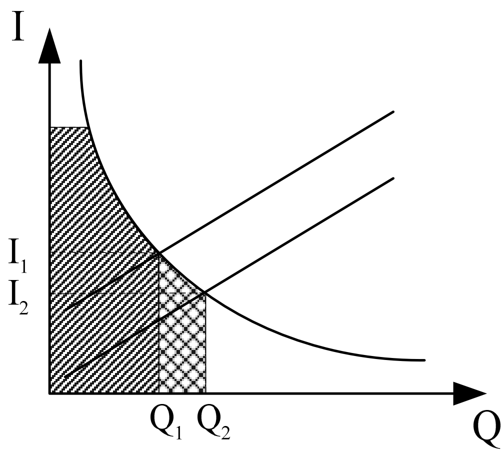


Figure 5 Modified Structure of Chromosome for Mutually Exclusive Projects

Demand Elasticity and Benefit Measurement in Simulation Model

In general, it is assumed that a given demand at the start of simulation is already consistent with network equilibrium and determined in response to congestion levels in regional routing analysis. In the long term, a secular traffic increase may occur due to demographic and economic growth. The demand should also be sensitive to the impedance, i.e., generalized trip cost, for any O/D pair. Usually, travel time is a major determinant of service in a waterway network and thus constitutes an impedance factor for estimating demand. Therefore, demand can be formulated as a function of with growth rate and impedance. According to economic theory, user benefits B_u are measured as the shaded area under the demand curve up to the actual demand Q_1 (as shown in Figure 6):



$$B_u = \int_0^{Q_1} I \cdot dQ$$

Figure 6 Traffic Demand vs. Impedance Factor

The demand may be limited by various factors determining system capacity, such as chamber dimensions and lockage times. Thus, demand increases as service times and, hence, capacity improves. When impedance decreases from I_1 to I_2 and traffic volume increases from Q_1 to Q_2, the area of user benefit increases. Similarly, a capacity reduction increases impedance and decreases user benefits.

Demand Function

In Wang and Schonfeld (2007), when lock closures (partial or total) are explicitly considered, the local volume to capacity (V/C) ratios may become very critical, and possibly far above 1.0 for long periods. In Figure 7(a), partial closures are scheduled at times t_1, t_2, and t_3 , which greatly reduce lock capacities in periods t_1 to f_1, t_2 to f_2, and t_3 to f_3 , respectively. If the V/C ratios exceed 1.0 and the demand level λ stays unchanged over time, the number of tows accumulating in queues is represented by the shaded area in Figure 7(b). In Figure 7(c), the queue length increases at different rates when inflows exceed lock capacities and decreases when there capacity exceeds inflows (i.e., $c - \lambda$). We must then consider how demand should be adjusted during closure times, i.e., in response to the reduced capacity and resulting delays.

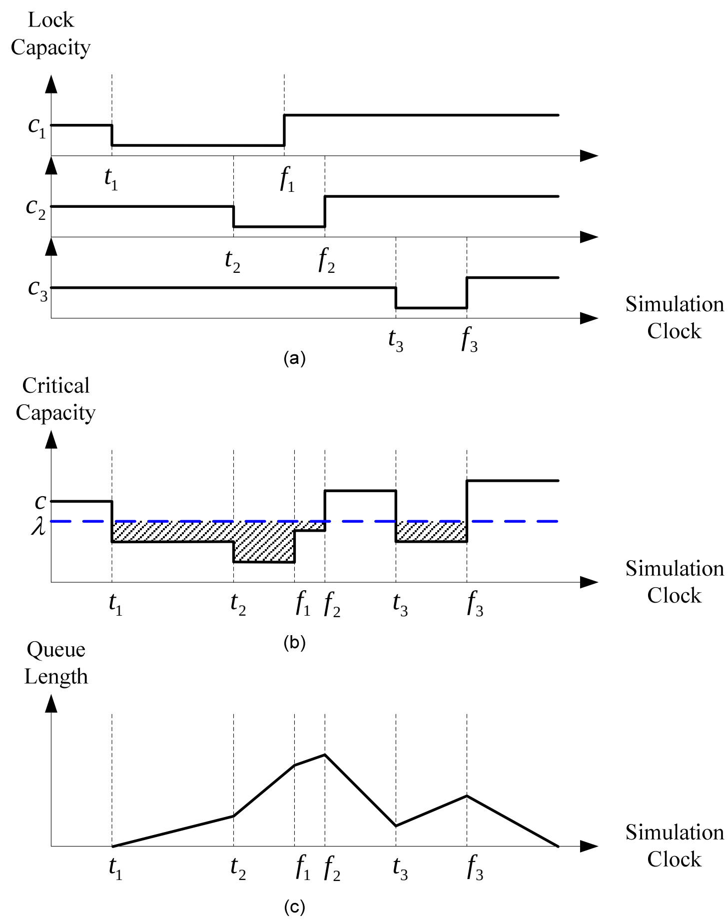


Figure 7 Capacity Change during the Lock Closure

It is possible that delays may rise rapidly when a local capacity is reduced to zero or near zero during a closure time, if demand cannot respond to the resulting delay. In order to avoid infinite queues, an elastic demand model is used here to account for traffic sensitivity to the total travel time, which is mainly affected by lock service times. For each O/D pair, the demand function is assumed to be a hyperbolic curve, which has a constant elasticity with respect to an impedance factor. In order to normalize the factor of total travel time, the ratio of real travel time (*z*) to baseline expected travel time (*y*), z/y, is used as the impedance factor.

When we also consider secular growth in traffic, we let λ_{ij} denote the generation rate in a particular interval, r_{ij} denote the annual growth rate and k_{ij} denote the demand elasticity for each O_i / D_j pair. Then the demand function for each simulation period t can be expressed as

$$(\lambda_{ij})_t = (\lambda_{ij})_{t-1} \cdot (1 + r_{ij})^{t_p} \cdot [\frac{(z_{ij})_{t-1}}{y_{ij}}]^{k_{ij}} \qquad (1)$$

where t_p duration of simulation interval

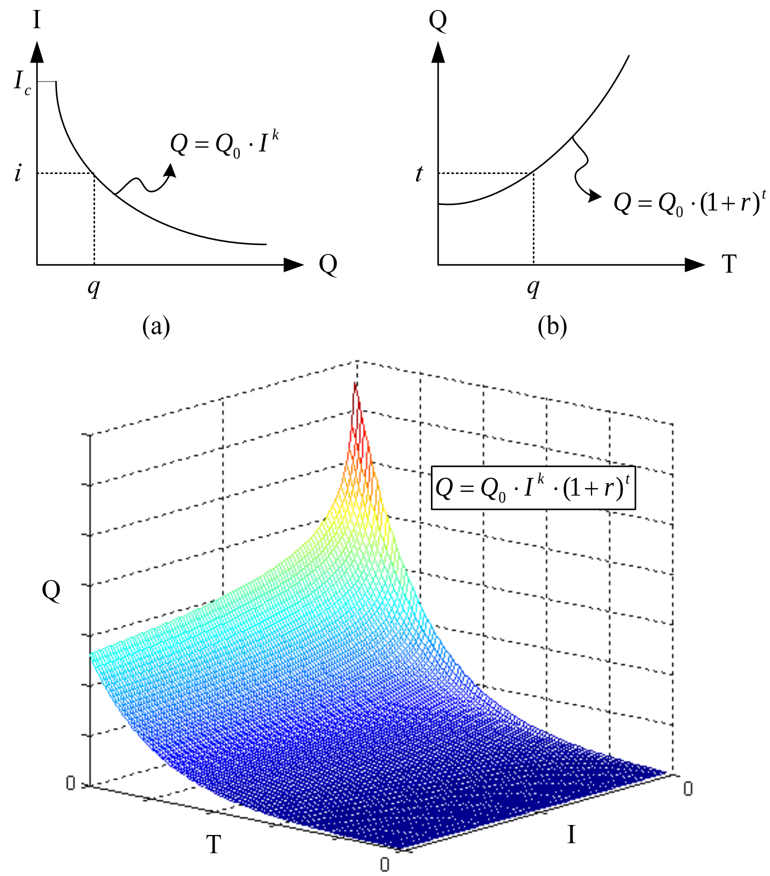
z_{ij} simulated travel time for interval t

y_{ij} expected travel time in the base case simulation

Measurement of User Benefit

If the demand was fixed, i.e., having zero elasticity, then a total cost function (including construction, maintenance, vessel operations and user costs) would suffice to compare scenarios or drive an optimization process. However, if the demand can be affected by simulated decisions, we should maximize a net benefit function rather than minimize total cost. (Otherwise, the optimization might favor decisions that drive traffic, and hence costs, towards zero.) The net present worth (NPW) should be the present worth of total benefits minus the present worth of total costs, with user benefits estimated from the demand functions.

As discussed above, the proposed demand for each O/D pair is a function of continuous impedance and time variables. As shown in Figure 8, (a) and (b) are the projections of (c), an outward hyperbolic demand surface. At any time *t*, demand function Q_t can be expressed as $Q_t = Q_0 \cdot (I_t)^k \cdot (1+r)^t$.



(c)

Figure 8 Proposed Waterway Demand Function

In order to model demand effects in the simulation model, a total simulation period is divided into *n* equal time intervals of duration T. As shown in Figure 9, the total user benefit, in any time interval *i*, is defined as the area under the demand (=marginal user benefit) curve for that interval, integrated from *0* to Q_i and truncated at an impedance value I_c, where the waterway cost reaches the cost of the rail alternative (Wang et al (6)). Due to discontinuities, assuming the demand elasticity k is constant over time, demand curves projected from Figure 8 are shown in the top-right corner of Figure 9 for successive time intervals *i* ($t_1 < t_2 < t_3 < t_4 < t_5$). Thus, the total user benefits in the i^{th} interval are obtained by integrating the demand function, $\int_0^{q_i} I \cdot dQ$, where q_i is the total volume in the i^{th} simulation interval. Since q_i may fluctuate in different simulation intervals, the overall undiscounted user benefit for the entire simulated period is $\sum_{i=1}^{n}\left(\int_0^{D_i} I \cdot dQ\right)$.

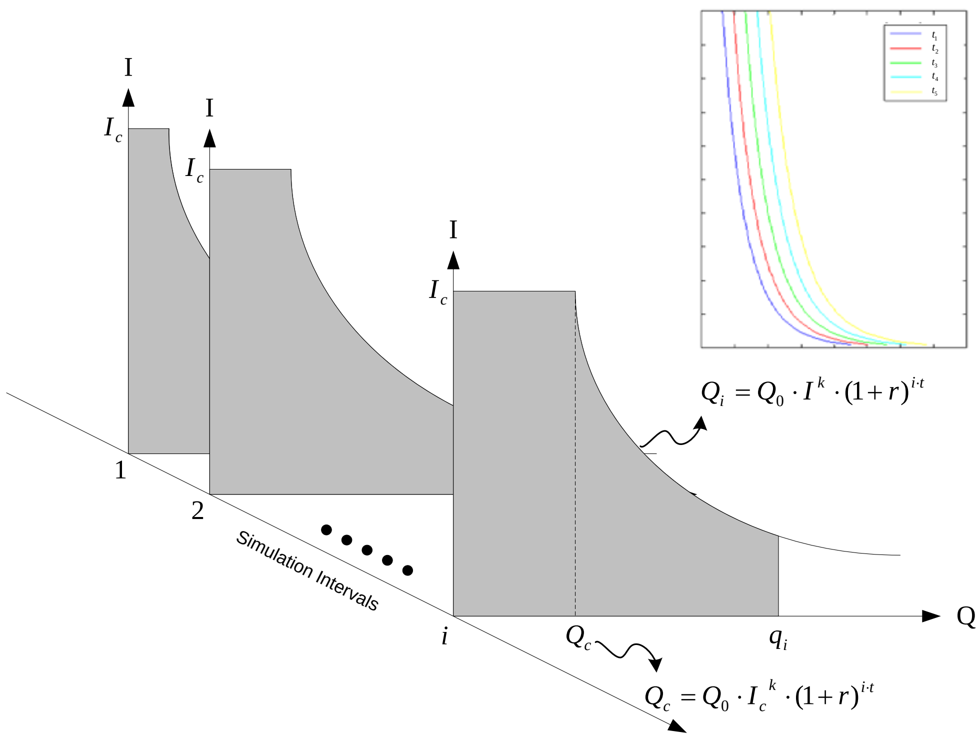


Figure 9 Total User Benefit

Therefore, with traffic changes in different intervals, the present worth of overall user benefit for the entire analysis period (*n* equal time intervals) is:

$$B_u = \sum_{i=1}^{n} \frac{v}{(1+R)^i}\left(\int_0^{Q_i} I \cdot dQ\right) \tag{2}$$

where

v	time value ($/tow hour)
B_u	present worth of user benefits ($)

R interest rate (per time interval)

Objective Function

The objective function is the present worth of net system benefit *NB* which is total benefit *TB* minus total cost *TC*. To simplify this analysis, only construction costs, user costs and user benefits are considered in the objective function. Operation costs and benefits on the supplier side are not considered here. In this study, user benefits are estimated from the demand function and user costs are estimated from the simulated total travel times, which include cruising times, lock service times, and delay times. Several costs, e.g., for fuel, crew, vessel depreciation, time value of cargo, are combined into an hourly cost v (in \$/tow-hour). This cost also represents the users unit time value, and when multiplied by impedance *I*, it transforms the impedance and user benefit measures from hours to \$. Therefore, let c_i denote the total travel time in interval *i*, the net present worth (*NPW*) is simplified here as the present worth of total user benefit minus total user cost:

$$\begin{aligned} NPW &= B_u - C_u \\ &= \frac{v}{(1+R)^i}\left(\sum_{i=1}^{n}\left(\int_0^{Q_i} I \cdot dQ\right) - \sum_{i=1}^{n}(c_i)\right) \end{aligned} \tag{3}$$

If project selection is considered (especially for mutually exclusive projects), the supplier's construction cost C_c cannot be factored out and should also be subtracted from the measured benefit. Let b_i denote the budget spent in the interval *i* and NPW is found by subtracting the project construction cost.

$$\begin{aligned} NPW &= B_u - C_u - C_c \\ &= \frac{v}{(1+R)^i}\left(\sum_{i=1}^{n}\left(\int_0^{Q_i} I \cdot dQ\right) - \sum_{i=1}^{n}(c_i) - \sum_{i=1}^{n}(b_i)\right) \end{aligned} \tag{4}$$

Improvements to Genetic Algorithms

In our previous work, most GA characteristics are fairly standard, including the creation of solutions, the genetic operators and the computation method. Some improvements are proposed to enhance GAs search performance by creating weighted sequences, developing smarter problem-specific genetic operators and applying parallel computing techniques.

Generation of "Weighted" Sequences

After the first generation of solutions, GAs create further solutions by applying designed genetic operators. However, sometimes we may still need to generate solutions without using any operators, such as in the steps of creating initial population and selecting possible parents. It is clear that all individuals in the initial population are generated without applying any genetic operators. Although all the candidate parents are selected from the current population based on their fitness values or ranks, any specific individual may only be selected a limited number of times in order to prevent "super individuals" from dominating the population. Thus, if an individual has been already selected as candidate parent for a fixed number of times (e.g., two times), whenever it is selected again, a brand new solution should be generated.

Besides, based on the concept of evolution, stronger parents in current generations are more competitive and have a higher chance to produce stronger offspring in the next generations. With different problem characteristics, sequences generated from random selection might be less promising and may then reduce search efficiency. Therefore, it is preferable to have weighted sequences which contain information on problem characteristics in addition to ordinary random sequences.

In the proposed GA, there are two categories of generated solutions: random-order solutions and weighted-order solutions. In random-order solutions, each project in any one of the solution sequences has the same probability to have a particular implementation priority. That is, the sampling process randomly selects a project and leaves the other projects in a sampling space for the next sampling processes. All the projects have the same chance of being selected in any position of a solution sequence. Thus such randomly generated solutions can explore the search space by providing as many variations as possible. On the other hand, weighted-order sequences will favor projects at locks which have special traffic, cost, or benefit characteristics. By considering this prior knowledge, those solutions can help speed up the convergence process.

Different ways may be used to weight the order of projects. The most common consideration for project implementation sequence is the current lock congestion level. Bottleneck-order solutions then include the information about possibly optimal sequences, in which the projects are implemented in the order of the severity of total delays at individual locks. That is, based on individual lock delays, those projects at most congested locks have higher chances to be selected first in a solution sequence. In order to include the senses of delay severity, a baseline simulation run is pursued. In the baseline simulation, the network is evaluated with current traffic and system conditions. No traffic growth or lock improvement projects appear during the simulation.

Development of "Smart" and Problem-Specific Genetic Operator

In the previous section, weighted sequences are generated for the initial population and in the process of selecting candidate parents. A "smart" operator is then used to create more promising solutions in the reproduction process. Those problem-specific operators should be able to produce offspring oriented toward project features such as benefits and locations. Since those considerations are hard to implement in a "crossover" way, two "smart" mutation operators are proposed, namely project mutation (PM) and geometry mutation (GM).

The PM operator works similarly to the EM (reciprocal exchange mutation, Wang and Schonfeld 2006) operator, but with more than two-point swapping. The number of swapping points depends on the size of problem (e.g., number of projects). Figure 10(a) shows a 3-point PM operator. It swaps the projects in those randomly generated positions with the order of their capacity expansion ratios. Thus, a project with a higher capacity expansion ratio will be shifted foward to an earlier implementation time.

The GM operator considers the network geometry. The number of groups depends on the size of network. It first randomly selects numbers of positions on the chromosome. The adjacent projects are grouped based on the project location shown on the gene. Figure 10(b) shows a 2-group GM operator. It groups adjacent upstream and downstream projects with the selected project.

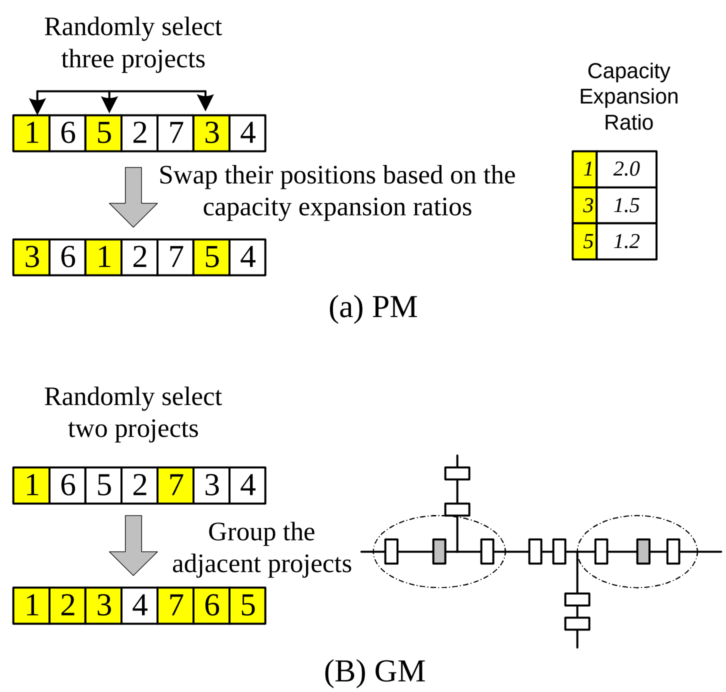


Figure 10 "Smart" Mutation Operators

Parallel Computing with GAs

Adapting the optimization model to parallel computing is a promising approach for speeding up the optimization process. In simulation-based optimization, the evaluation process is the most time-consuming task. If that evaluation process can be distributed among parallel processors, the optimization search time may be reduced almost in inverse proportion to the number of processors, with some additional time for "communication" between processors.

Parallel GAs (PGAs) are relatively easy to implement compared to other parallel computing algorithms. To improve the efficiency of running SIMOPT, we seek to adjust this model to parallel computing without changing the basic structure of the GA applied in it.

According to Yang and Schonfeld (2007), among several conceptual models of the major PGA paradigms, a hierarchical model (also called in the literature a “master-slave” model) is easy to visualize and it is relatively simple to implement. The master processor handles parameters necessary for the objective function evaluation to the slave processors; the slave processors receive the messages and perform the evaluation; objective function values are then returned to the master processor. Figure 11 shows the PGA procedure with the hierarchical model.

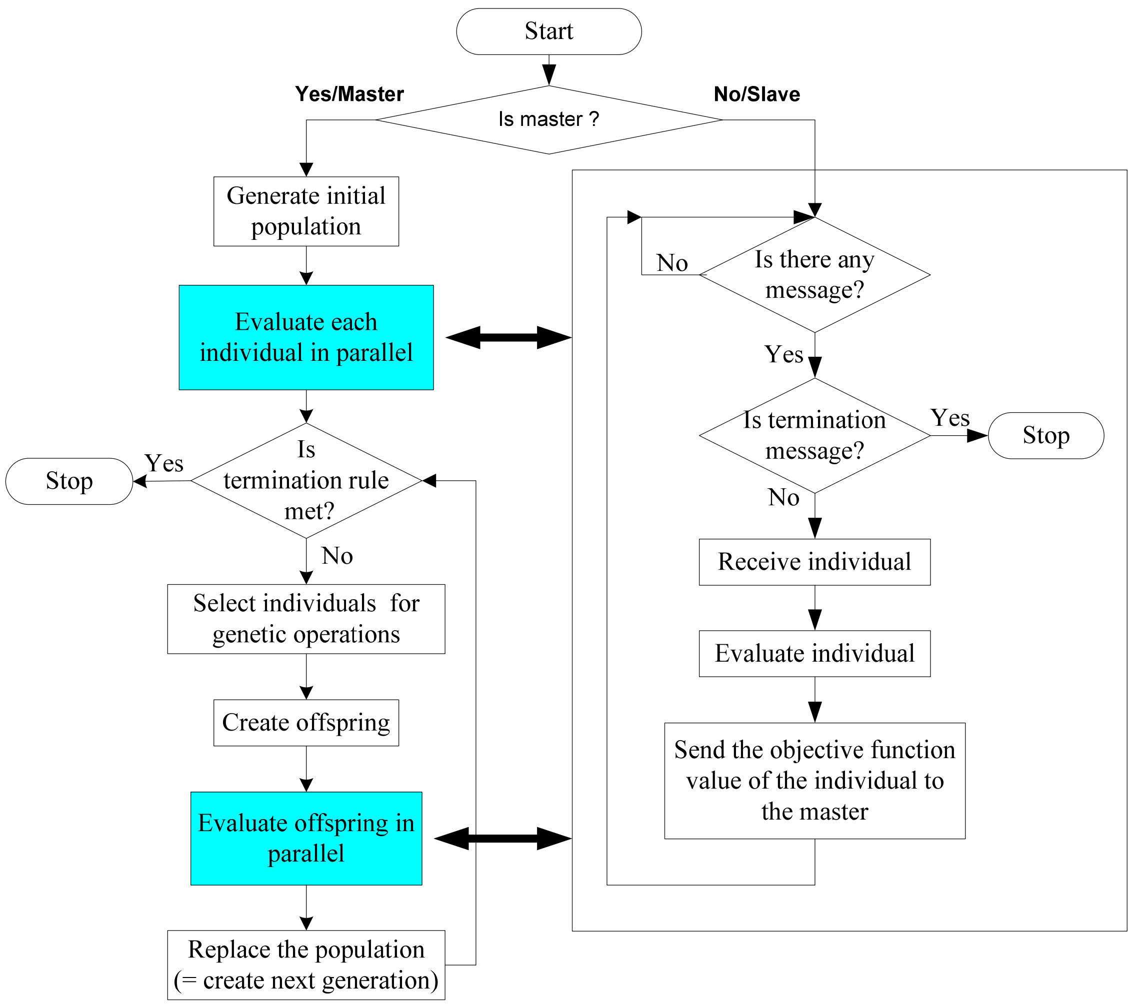


Figure 11 The PGA Procedure with the Hierarchical Model

As shown in Figure 11, in each PGA evaluation step the master processor assigns a subgroup of individuals to each slave processor and receives back their objective function values. Since the fitness evaluations are independent of one another, slave processors only need to communicate with the master processor, without interacting with one another. To reduce idle time and minimize redundant or wasted effort, it is important to design an appropriate task (individual) distribution method which can result in a balanced division of work between contributing processors.

Model Test (Enhanced SIMOPT)

The proposed demand model is incorporated into a generalized waterway simulation model developed by Wang and Schonfeld (2002). The simulation model in SIMOPT is a discrete-time, event-based microscopic simulation model which is developed to analyze trip behavior, lock operations and demand variation, and to evaluate the system performance, and is used here for demonstrating two aspects of demand sensitivity.

Incorporating Dynamic Demand into Simulation Model

In the basic SIMOPT simulation model, there are five operational modules: (1) the *port generation module* where a tow is generated at a port; (2) The *port arrival module* where a tow arrives at a port; (3) the *lock arrival module* where a tow arrives a lock; (4) the *lock departure module* where a tow leaves a lock; (5) the *port exiting module* where a tow ends its trip at a port. All events are invoked by the timing control module, preceded by updating the timing average, and followed by a statistical counting process. The logical organization among its basic event modules is shown in Figure 12. Starting with the timing control scheme and ending with the statistical module, all events (three port events for tows being generated, ending their trips and passing by, plus two lock events for tows being served and leaving) occur at network nodes during the simulation time.

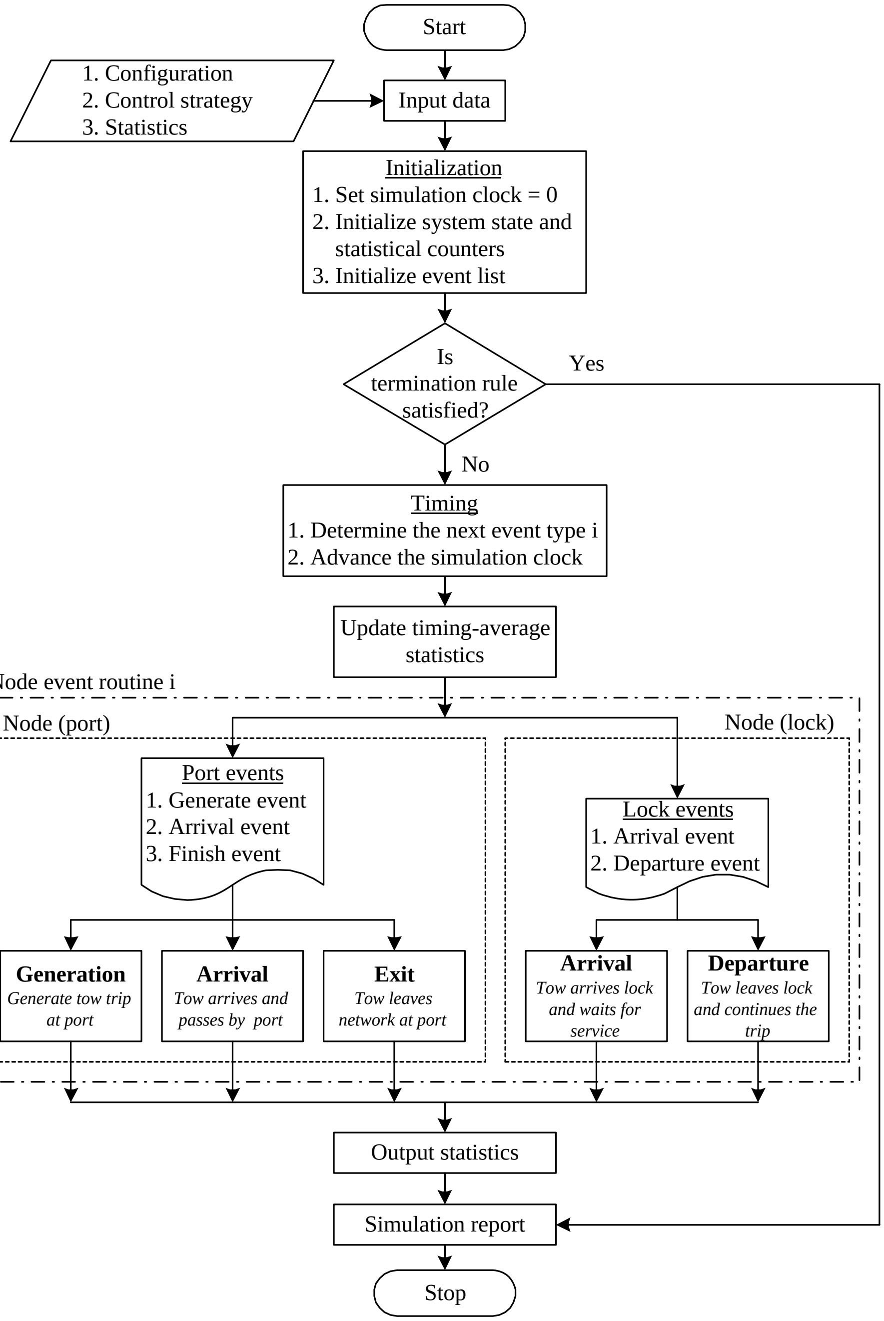


Figure 12 Overall Framework of Basic Waterway Simulation Model

The proposed simulation model is used to evaluate any sequence of projects. In a project scheduling problem, whenever a project sequence is generated by GA, the simulation model is called to evaluate the performance of generated project sequence. With the implementation schedule calculated from the cumulative budgets and project costs, projects are chronologically introduced into the simulation program and implemented immediately. Thus each simulation is run at least for the duration of the planning horizon *T*. As shown in Figure 13, except for the five timing events at ports or locks (defined in basic simulation framework), there are project implementation events which bring the project implementation timetable into the simulation and update the system by increasing capacity as well as reducing the service time during the simulation. By extension, if project construction times and capacity reductions are considered, two project implementation events will be included, namely the start of the project and the end of the project. When starting projects, the system is updated by decreasing capacity and increasing service time; when finishing projects, capacity is increased and service time is decreased. During the construction period, demand responds to service level and is elastically changed.

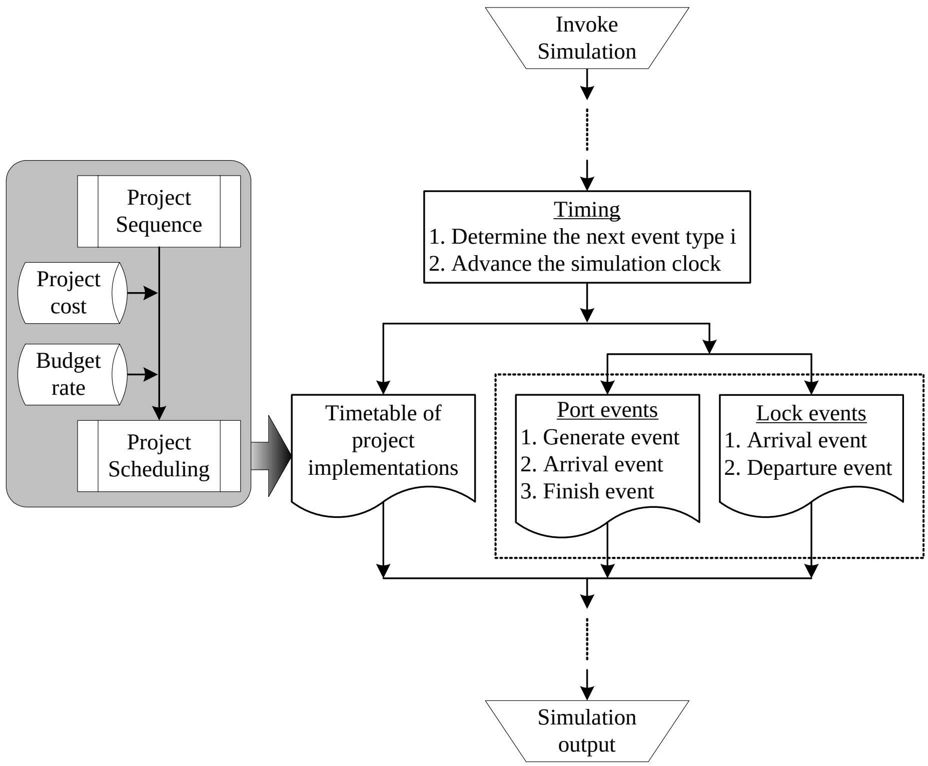


Figure 13 Simulation Model with Project Implementation

If demand is steady, or even seasonal, with unchanged trip rates (e.g., number of trips per day, month, season, or year) along the time axis, all trips can be generated in advance as a complete shipment list from which trips are then fed into the simulation model one by one based on their departure times. However, if demand grows annually or responds elastically to travel delays, as might result from project construction and improvements, a pre-generated shipment list is no longer suitable for the simulation model. In order to incorporate dynamic demand into simulation model, a generation event during the simulation becomes necessary. The generation event is called whenever the simulation clock runs to the instant when a new trip is ready to be generated. Only one trip is generated per generation event. After generating one trip, the next generation event is scheduled based on the most updated traffic level, including O/Ds and travel delays.

Figure 14 presents a flowchart for the generation event. While invoking a generation event, timing control has determined the origin port. Based on the O/D matrix, the tow trip is stochastically assigned a destination port. Meanwhile, the tow's size and travel speed are generated as part of that tow's data structure. In order to avoid generating extreme speed values, truncations of minimum and maximum speeds must be specified. After associating all the characteristics to the newly generated tow, the generation module sends the tow into the network link, determines its arrival time at the next lock and schedules next generation event based on the updated O/D matrix.

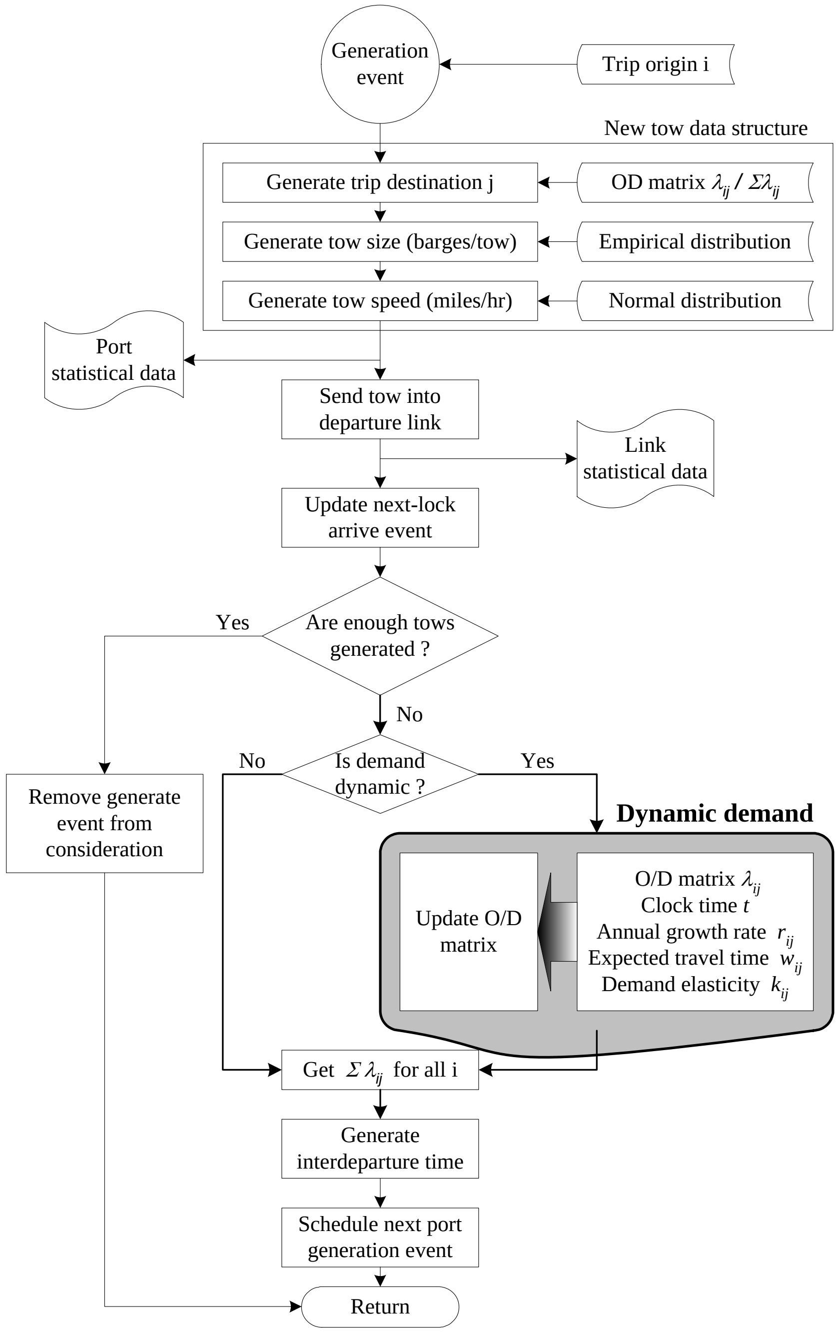


Figure 14 Port Generation Module

During the simulation run, the OD matrix, i.e. the trip generation rate, changes over time. The process of dynamically updating the existing O/D matrix is illustrated in Figure 15. An annual traffic growth rate (*r)* is first included with an exponential factor of the time interval (*t)* between successively generated events, i.e., $(1+r)^t$. Then the effects of demand elasticity depend on the uncongested O/D travel time from the baseline simulation and on periodic information collected within simulation runs.

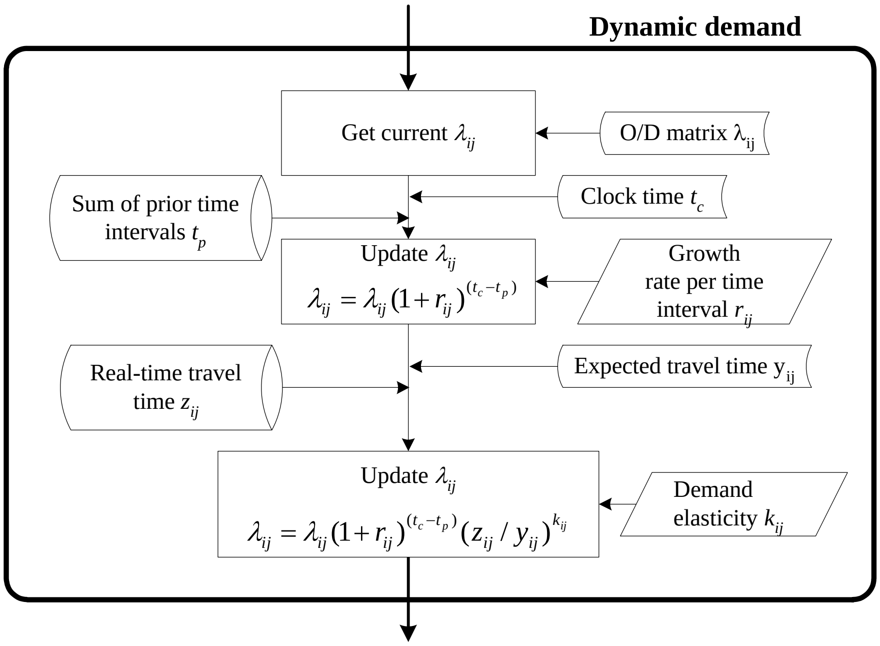


Figure 15 Dynamic Demand Module

Expected O/D travel time is simply acquired from a simulation base case, where a free-flow steady demand is considered over time and the O/D travel time is collected and averaged from all O_i/Dj trips. However, while gathering real-time information, it is possible to not have enough direct O/D travel time data for each O/D pair within a specified time period. Thus, in order to avoid the shortage of statistical data of O/D trips, the information on real O/D travel times is calculated indirectly from the summations of link travel times and lock processing times. Lock processing time includes regular lock service time and delay time in queue, which is the main factor influencing the launching decision for the next tow trip. Through normalization, defined as the ratio of real travel time (*z*) and base-case expected travel time (*w*), z/w, the traffic rates (or, equivalently, the time intervals between generated tows) can be updated by the normalized factor $(z/w)^k$, where *k* is the demand elasticity.

Based on predicted traffic growth, the O/D matrix values increase continuously at annual growth rates which may differ for various O/D pairs. Furthermore, if updated O/D travel times are provided to users during the simulation, the O/D matrix is changed based on the newly updated travel times and applicable elasticities.

Test Network

A simple test network is used here for testing proposed simulation-based optimization model (as shown in Figure 16). There are 3 rivers, 5 ports, and 7 locks (4 single-chamber locks and 3 double-chamber locks). Locks are numbered with ID's 0, 1, 2, 3, 4, 6, 7. Locks #5 and #8 are dummy locks. (The 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.

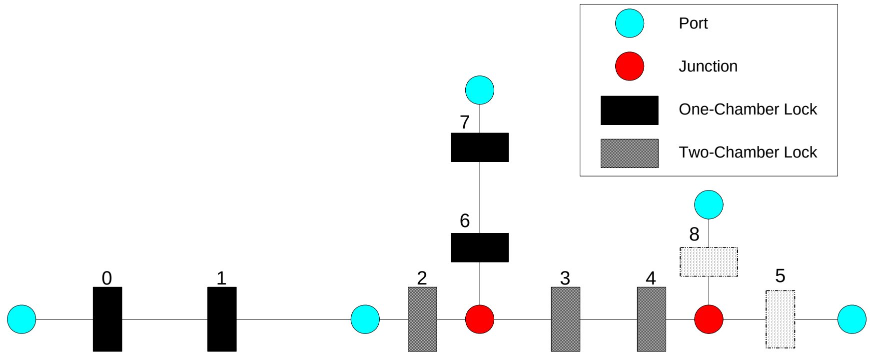


Figure 16 Test Network for SIMOPT Extension

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 basic project-relevant inputs include regional budget rate, project ID, costs, capacity expansion/residual ratios, and precedence relations (as shown in Table 1). The regional budget constraints limit the project funds in each region: \$40 $\times 10^6$, \$70 $\times 10^6$, \$40 $\times 10^6$ annually for regions 1, 2, and 3, respectively. The budgets are uniformly distributed within each year. For example, the alternative projects at locks #7, #2, and #6 are funded annually by the 2nd regional budget, 70 $\times 10^6$.

There are two kinds of precedence constraints. One limits the sequence of locks receiving improvement projects, and the other restricts the order of projects at the same lock. For example, all the projects at lock #6 should be funded before all the projects at lock #2 and #3. Projects #4, #7 and #12 must follow the completion of project #2, #6 and #11, respectively.

Table 1 Project Information

Project ID	Lock ID	Region Code	Capacity Expansion Ratio	Cost ($\times 10^6$)	Lock Precedence Relations	Project Precedence Relations
1	7	2	1.2	17		
2	7	2	1.5	20		2→4
3	7	2	1.8	23		

4	7	2	2.0	27		2→4
5	1	1	1.2	16		
6	1	1	1.5	20		6→7
7	1	1	2.0	26		6→7
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		11→12
12	0	1	2.0	29		11→12
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 order to accelerate the analysis, a high budget flow is specified. 10 replications are used here to complete one simulation evaluation of any candidate solution (i.e., generated project sequence and resulting schedule). The GA population size per generation in this test is set at 500 or 1000 based on the problem size. An interest rate of 4% and an average time value of $450/tow-hour are assumed. In the evolution process, if the generated sequence violates a constraint, its fitness value is assigned a large cost (or zero benefit) and is unlikely to be selected as a parent for next generation.

The termination rule for GA search is set when the optimized solutions stays unchanged for 50 generations. Mutation and crossover rates are 0.7 and 0.3, respectively. Since parallel computing is tested in this phase, all the tests are run in parallel on several Pentium IV processors with 3.2 or 3.6 GHz CPU, and 1 or 2 GB memory, depending on the availability of computer resources.

Test Results

Adding Constraints (Multiple Projects with Different Implementation Time)

In this test, multiple projects are considered at some lock locations, and all projects will be funded at different times. Therefore a full implementation sequence with 18 projects is sought in this test: 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. Without mutual exclusivity constraints for projects at some locks, the solution space is 18! = 6,402,373,705,728,000. However, with 2 lock precedence constraints and 3 project precedence constraints, the solution space is further reduced to ($C_7^{18} \times 7! \times 3! \times 2!$) = 1,924,715,520.

Due to increased problem size, restricted memory and search time within one computer are of concern after searching through several generations. Thus PGAs are used in these tests in order to alleviate the memory loads for each computer and also reduce the search time for a near optimal solution.

In this test, each feasible solution is simulated 10 times with different random seeds. A fitness value is calculated by averaging those 10 evaluations. Penalties (e.g. large costs) are directly assigned to the infeasible solutions (i.e. those that violate constraints) before any simulation is used to evaluate them. For any generation, parents are selected based on the ranks of their fitness values. Individuals with large cost are less likely to be selected to produce offspring. For GA, since the solution space is huge, population size for each generation starts with 500 and the overall search is ended when optimized solution stays unchanged for 100 generations.

With those pre-specified parameters, the optimized results of 20 GA search processes with different random seeds are shown in Table 2. The inherited stochastic features in GA result in various values in the search process, such as number of generations required to locate optimized solution, number of total generated sequences in any search, and number of evaluated sequences which go through simulations. The optimized results from different GA searches are also slightly different (approximately 1% difference between \$647,830,771, \$655,890,781, \$655,902,872, and \$655,916,715) but should be very close to the lower-cost tail of the distribution of solutions. The lowest cost found as \$647,830,771 is found from 5 GA searches. The optimal sequence for these 5 GA searches is 8 → 5 → 2 → 16 → 11 → 4 → 6 → 9 → 18 → 14 → 7 → 15 → 13 → 3 → 12 → 17 → 1 → 10.

Approximately 3.7 ~ 3.8 hours are required for completing one generation on 6 parallel computers (Pentium IV 3.2 or 3.6 GHz, 1 or 2 GB memory). Total search times vary for different GA searches and are based on the number of generations required to find their optimized solutions. Significant improvements in the time needed to obtain results from simulation-based optimization process result from applying parallel computing. Without PGAs, more than a week might be required for each GA search for this problem. The larger the problem, the more valuable the PGAs become.

Table 2 Optimized Results (Minimization Problem)

GA Search	# of Gen.	# of Generated Sequences	# of Evaluated Sequences	Optimal Total Cost (\$)	Search Time (sec)
1	132	126,463	4,265	655,916,715	75,714
2	226	216,368	6,370	647,830,771	118,622
3	136	130,245	4,382	655,902,872	77,704
4	141	135,242	3,676	647,830,771	50,332
5	169	161,628	4,731	647,830,771	61,476
6	111	106,343	3,389	664,294,640	51,666
7	138	132,360	4,313	655,902,872	64,808
8	134	128,106	4,330	657,931,927	64,566

9	147	140,863	4,105	653,778,575	60,811
10	140	133,988	3,964	655,902,872	60,041
11	158	151,415	4,028	647,830,771	60,510
12	120	114,987	3,766	655,902,872	56,039
13	159	152,350	4,830	657,931,927	72,601
14	223	213,243	7,569	655,916,715	113,584
15	218	208,326	6,663	655,890,781	99,610
16	126	120,793	3,650	672,879,681	55,085
17	148	141,731	4,291	647,830,771	64,208
18	202	193,169	6,038	655,890,781	91,451
19	150	143,759	5,026	655,916,715	75,019
20	239	228,611	6,968	664,294,640	104,802

The evolution of objective values from 4 GA searches (searches #3, #14, #15 and #18) which have more than 200 generations is plotted in Figure 17. Though four searches converge with slightly different optimized solutions (approximately 1% difference), the optimized solutions improve relatively quickly in early generations and converge at the end of genetic search.

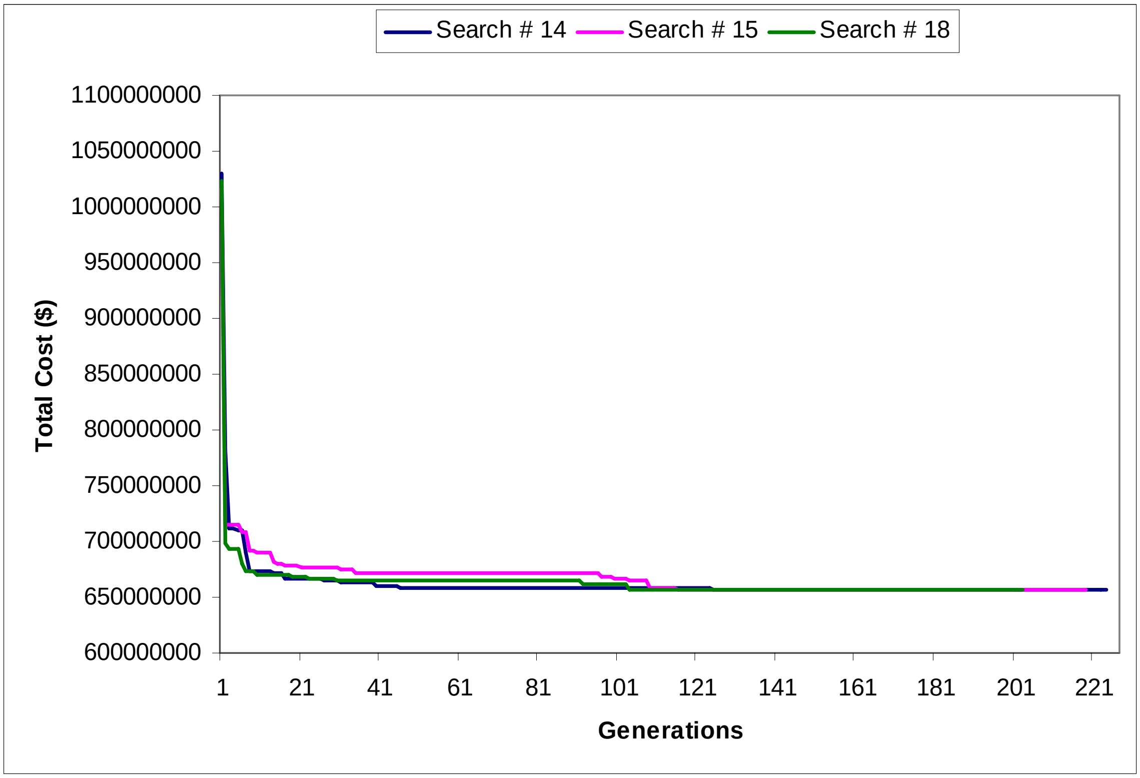
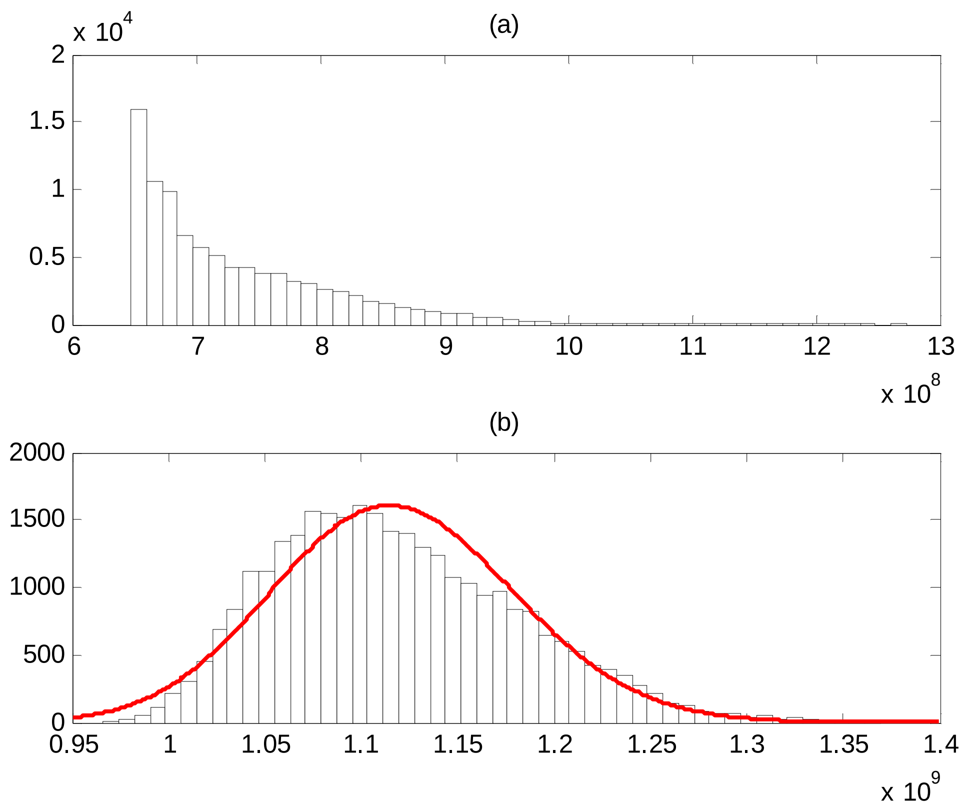


Figure 17 GA Search Performance (Minimization Problem)

Figure 18(a) shows the solution distributions (i.e. histogram of solutions) throughout the GA search process. A total of 93,774 feasible solutions are generated from 20 GA searches with a mean of 7.3122×10^8 and standard deviation of 7.7979×10^7. In each GA procedure, in addition to solutions created in the initial population, new solutions are all created as offspring from selected parents from previous generations and specified mutation/crossover operators. Thus, except for the solutions in initial population which are generated randomly, the newly generated solutions from reproduction process are no longer created randomly in the solution space. The created offspring are highly related to the selected parents who have better fitness values and higher chances to survive in the evolution process. Compared with the solution distribution (shown in Figure 18(b)) based on 30,547 randomly generated feasible solutions (out of 1,000,000 random solutions), the GA changes the random search to a smarter search which search solutions in the domain which contain solutions with better fitness values. Thus, the GA process should have already directed toward on the search more efficiently to optimal solution.

Figure 18(c) further jointly shows density functions for two solution distributions. The histograms on the left and right sides are for the solutions produced in GA search and in random search, respectively. As can be seen, the GA process directs the search into the domain with lower cost with a mean of 7.3122×10^8. Very conservatively, based on the fitted normal distribution with a mean of 1.1182×10^9 and standard deviation of 6.303×10^7, the mean of GA solution set is located in the tail far away from the mean of random solution set (more than 6 standard deviations). The optimal solutions found from different GA searches are further away in the tail. It is estimated that the probability of finding a solution better than optimal solutions found by the GA is extremely low, i.e., very close to zero.



(c)

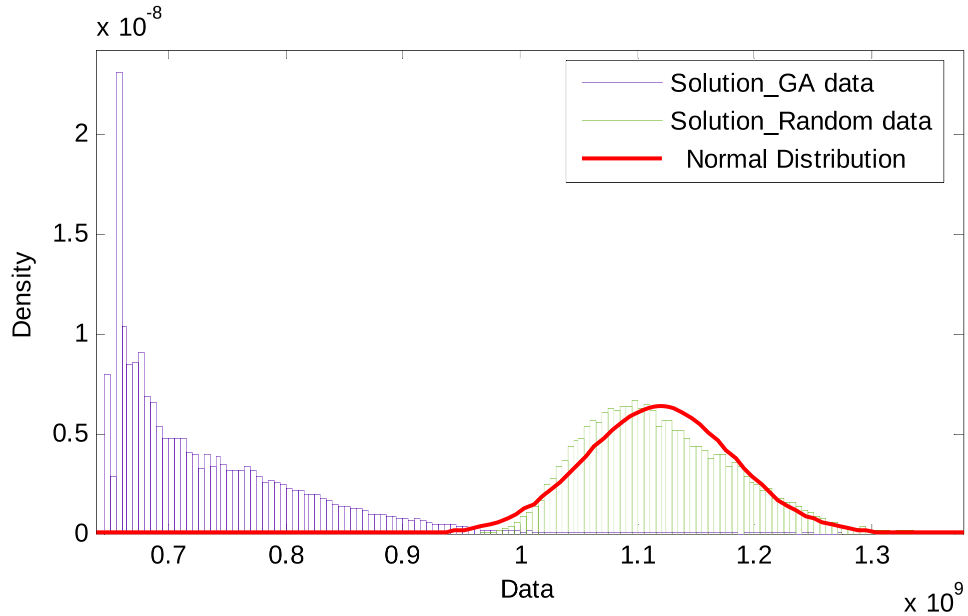


Figure 18 Solution Distributions

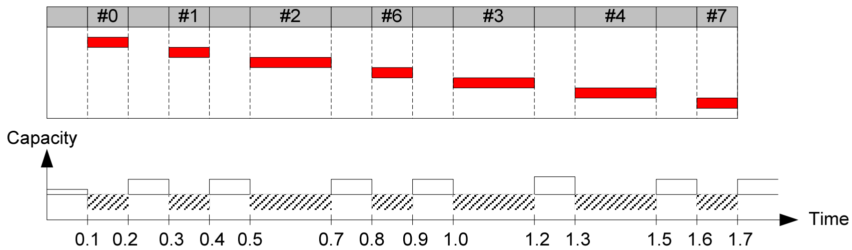
Measuring Net Present Worth with Dynamic Demand

Table 3 shows the information on scheduled work (such as construction projects or maintenance tasks which require partial lock closures) at locks, including work schedule, closure duration, residual capacity ratio during the closure period, and improved capacity ratio after the work completion. Lockage rates vary and lockage times vary in SIMOPT inversely with the residual ratio. Thus, if the residual capacity ratio is 0.5, the service time will double. In this case, closures for different projects occur at different times. A two-year period is simulated, after running a one-year warm-up period to populate the network with traffic and approach a steady-state. With a given schedule of lock closures, O/D traffic responds to the simulated service level based on the given elasticity coefficients. Thus, the total user benefit and total costs are computed from simulation outputs. Throughout the simulation (for each weekly period), user travel times (including delays) and user benefits (integrated from demand functions) are recorded so that aggregate performance measures can be computed at the end of the simulated period by summing individual movements. By tracking individual movements, total costs are summed at the end of simulation.

Table 3 Work Information and Schedule

Lock ID	Work Schedule (yr)	Work Duration (yr)	Residual Capacity Ratio	Improved Capacity Ratio
0	0.1	0.1	0.5	1.2
1	0.3	0.1	0.5	1.2
2	0.5	0.2	0.7	1.5
6	0.8	0.1	0.5	1.2
3	1.0	0.2	0.7	1.5

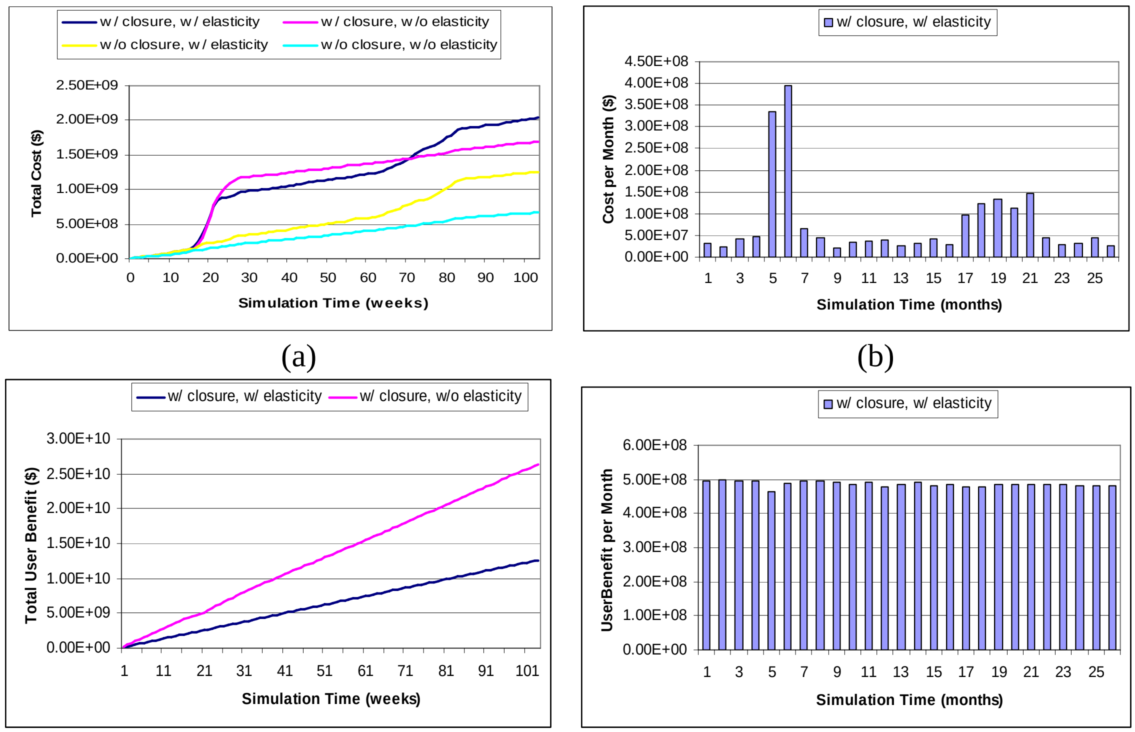
4	1.3	0.2	0.7	1.5
7	1.6	0.1	0.5	1.2



Four designed scenarios are tested: with or without capacity reductions (i.e., closures) during construction work, and with or without demand elasticity. The capacity reductions during closures are expected to greatly increase costs. During such periods, service rates should greatly decrease and queuing delays should greatly increase. Projects that increase lock capacity would reduce delays and, if demand is elastic, increase subsequent traffic.

Figure 19 shows the simulated cost and benefit outputs from four different scenarios with a summary table on the top. The elasticity coefficient *k* is -0.1 for all for scenarios that consider elasticities. Figure 19(a) shows the cumulative costs over the simulation time for the four scenarios. Scenarios with closures have much higher total costs than those without. Since work projects reduce the impedance to traffic, scenarios with elasticity have greater traffic after the work and hence greater total costs (but also greater benefits). Figure 19(b) shows monthly changes in average cost based on the first scenario, considering closure and demand elasticity. Early in the simulation period, user cost may increase due to closures. As more completed projects add capacity at locks, traffic experiences less impedance and incurs less cost. Figure 19(c) compares the cumulative benefits of scenarios with and without demand elasticity. Those with elasticity yield lower benefits by having traffic respond to travel times. Along the simulation time, traffic grows as more works are completed. Figure 19(d) shows the monthly benefits over the simulation time based on the first scenario.

Scenarios	**Simulation Outputs**		
	Total Tows in System	**Total Cost ($)**	**Total Benefit ($)**
(1) w/ closure, w/ elasticity	804,529	2.03×10^{9}	1.26×10^{10}
(2) w/ closure, w/o elasticity	732,561	1.69×10^{9}	2.64×10^{10}
(3) w/o closure, w/ elasticity	672,947	1.25×10^{9}	1.27×10^{10}
(4) w/o closure, w/o elasticity	554,299	6.66×10^{8}	2.64×10^{10}



(a) (b)

(c) (d)

Figure 19 Simulation Outputs w/ and w/o Demand Elasticity

Including Construction Time and Capacity Reduction

Construction time and residual capacity ratio are included in the project information shown in Table 4. Some construction requires total or partial lock closures. That is, during the construction period, capacities at some locks decrease based on the relevant residual capacity ratios. Afterwards, the capacities increase based on capacity expansion ratios when improvement projects are completed. The tests are still subjected to all the project precedence and lock precedence constraints shown in Table 1.

Table 4 Project Information (continued from Table 1)

Project ID	Lock ID	Capacity Expansion Ratio	Cost ($\times 10^6$)	Construction Time (year)	Residual Capacity Ratio
1	7	1.2	17	0.09	0.8
2	7	1.5	20	0.10	1.0
3	7	1.8	23	0.13	0.8
4	7	2.0	27	0.17	0.2
5	1	1.2	16	0.04	0.8
6	1	1.5	20	0.05	1.0
7	1	2.0	26	0.09	0.5
8	6	1.5	27	0.10	0.8
9	6	2.0	33	0.12	1.0

10	0	1.2	20	0.09	0.8
11	0	1.5	12	0.10	1.0
12	0	2.0	29	0.11	0.5
13	2	1.1	32	0.03	0.8
14	2	1.2	35	0.05	0.8
15	4	1.1	25	0.01	1.0
16	4	1.2	27	0.05	0.5
17	4	1.3	31	0.09	0.2
18	3	1.1	35	0.04	0.5

As discussed above, since more traffic delays are expected when lock capacity is reduced, dynamic demand should be estimated in the simulation model to avoid infinite queues during the construction periods. Therefore, in this scenario, a net benefit maximization approach is formulated to solve this optimization problem. Fitness values measured from simulation model are the net benefits (as shown in previous section), rather than total costs. In this analysis, the construction costs need not be explicitly subtracted from the benefits (in order to obtain net benefits) because the same construction costs as limited by the budget constraints, are spent over a given analysis period, regardless of the project sequence that is evaluated.

In a minimization problem, large penalty values are assigned to the infeasible solutions which violate any of constraints. In a maximization problem, zero benefits are assigned to those infeasible solutions. Based on the ranks of fitness values, individuals with zero benefits still have very slight chances of being selected as parents to produce offspring for next generation.

The optimized results and evolution processes from 5 searches are shown in Table 5 and Figure 20. The optimal sequence shown in first GA search, 2 → 6 → 15 → 8 → 11 → 9 → 16 → 3 → 7 → 4 → 17 → 14 → 12 → 13 → 18 → 10 → 1 → 5, is the best one among these 5 searches with largest net benefits.

Table 5 Optimized results (Maximization Problem)

GA Search	# of Gen.	# of Generated Sequences	# of Evaluated Sequences	Optimal Total Net Benefit ($)
1	165	158,067	4,358	9,445,148,835,840
2	290	277,410	6,267	9,428,852,275,200
3	254	242,991	5,264	9,450,640,465,920
4	161	154,075	3,685	9,443,086,110,720
5	147	140,805	3,137	9,427,986,063,360

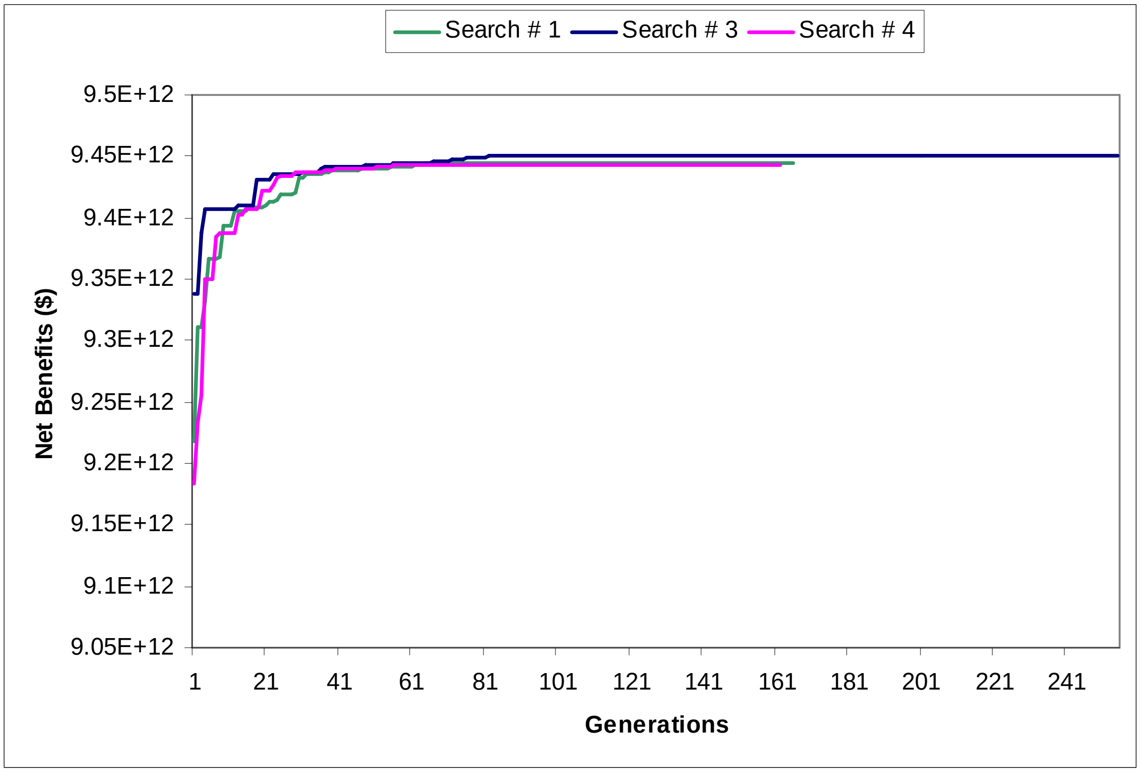


Figure 20 GA Search Performance (Maximization Problem)

The new optimal sequence is entirely different from that found in the previous scenario that does not consider construction closure and capacity reduction during the construction periods, as shown in Table 6. It is important that the implementation decision for improvement projects is strongly based on the total benefits as well as the total costs. The shippers' response to any possible service interruption has significant effects on overall network performance and resulting optimal project sequence.

Table 6 Comparison of Optimal Project Sequences

CT & CR	Optimized Sequences
w/o	8→5→2→16→11→4→6→9→18→14→7→15→13→3→12→17→1→10
w/	2→6→15→8→11→9→16→3→7→4→17→14→12→13→18→10→1→ 5

* CT: construction time; CR: capacity reduction

Tradeoffs between Construction Times and Costs

If the tradeoffs between construction times and costs are considered, more project information is required, such as detailed ID information (including alternative ID, project ID and lock ID), and tradeoff alternatives among construction costs and times (as shown in Table 7). When construction times are considered, the effects of partial lock closures with residual capacity ratios must also be considered. Shorter closure times would usually require higher construction costs.

As discussed above, tradeoff alternatives between construction times and costs can be considered as mutually exclusive alternatives for any project (for example, project #1 has 2 construction cost/time alternatives, project #2 has 3 construction cost/time alternatives, etc). That is, only one alternative is selected for any single project. In this case, 18 out of 42 alternatives are selected for 18 projects. After prescreening the generated solutions based on the precedence constraints, the simulation model is used to evaluate the feasible solutions and zero benefits are directly assigned to infeasible solutions.

Table 7 Project Information (continued from Tables 1 and 3)

Project ID	Alternative ID	Cost ($\times10^6$)	Time (years)		Project ID	Alternative ID	Cost ($\times10^6$)	Time (years)
1	1	17	0.09		10	23	20	0.09
1	2	15	0.10		10	24	30	0.06
2	3	20	0.10		11	25	25	0.10
2	4	15	0.13		11	26	20	0.13
2	5	10	0.20		11	27	30	0.08
3	6	23	0.13		12	28	29	0.11
3	7	20	0.15		12	29	35	0.09
4	8	27	0.17		12	30	40	0.08
4	9	20	0.23		13	31	32	0.03
4	10	30	0.15		13	32	48	0.02
5	11	16	0.04		14	33	35	0.05
5	12	12	0.05		14	34	44	0.04
6	13	20	0.05		15	35	25	0.01
6	14	25	0.04		15	36	20	0.02
6	15	30	0.03		16	37	27	0.05
7	16	26	0.09		16	38	35	0.04
7	17	20	0.12		17	39	31	0.09
7	18	30	0.08		17	40	27	0.10
8	19	27	0.10		18	41	35	0.04
8	20	30	0.09		18	42	42	0.03
9	21	33	0.12					
9	22	40	0.10					

In this test, there are two categories of solutions, random-order solutions and weighted-order solutions. That is, in the initial population, one half of the sequences are generated with random order and the other half are generated based on projects' priorities which are defined as lock congestion levels. Projects at the same lock are assumed to have the same priorities. In the selection process, new solutions, if necessary, are also generated with the projects' priorities. If multiple projects are considered at the same lock, the same priorities are assigned to those projects. Additionally, two "smart" mutation operators, PM and GM, designed for this specific problem (as discussed in previous section) are added to the process of creating offspring, to further refine some of the offspring based on specified probabilities. It is assumed that the prioritized individuals and "smart" operators help expedite the search process.

The optimized results from one single search (i.e., one random seed) are shown in Table 8 based on "smart" GA and "standard" GA, respectively. The resulting optimal sequences and schedules are presented with listed alternative IDs and their relevant project IDs and lock locations. As can be seen from this single search, "smart" GA by adapting weighted sequences and problem-specific operators can find better solution, with higher net benefits, than "standard" GA does. Again, the optimal sequence found in this scenario is entirely different from that found in the previous scenario without tradeoff alternatives. This indicates that the implementation decision changes if available resources (e.g. time and cost) change.

Table 8 Optimized Results

	Number of Generation	Optimized Total Net Benefit ($)
"Smart" GA	272	9,521,229,404,160
"Standard" GA	273	9,506,448,046,080

"Smart" GA				"Standard" GA		
Alternative ID	**Project ID**	**Lock ID**		**Alternative ID**	**Project ID**	**Lock ID**
15	6	1		24	10	0
2	1	7		12	5	1
20	8	6		20	8	6
28	12	0		5	2	7
5	2	7		13	6	1
12	5	1		21	9	6
21	9	6		39	17	4
26	11	0		26	11	0
42	18	3		10	4	7
6	3	7		36	15	4
17	7	1		16	7	1
36	15	4		34	14	2
8	4	7		32	13	2
32	13	2		41	18	3
33	14	2		28	12	0
38	16	4		6	3	7
40	17	4		1	1	7
24	10	0		37	16	4

Other GA Applications for Waterway Operations

The optimization based on evaluating objective functions with simulation is becoming feasible but computation time is crucial. With the advantage of advanced computer resources (e.g. faster CPU) and techniques (e.g. parallel computing), other applications of simulation-based optimization for waterway operations becoming feasible, such as network-level or lock component-level maintenance planning, in addition to the capital investment discussed earlier

In order to apply the current methodology on waterway maintenance planning, modifications in the objective function and cost structure are required. Again, since the optimization method can be fully separated from the simulation model, the development efforts for these two processes can proceed concurrently. Both of them should be able to handle specific characteristics of maintenance problem.

Network Level Maintenance Planning and Scheduling

Ideally, a general objective of the maintenance planning process should be maximizing overall net benefits, including the costs and benefits associated with system failures and the performance of maintenance and repairs. Due to lock deterioration in waterway network, in addition to have capital investment on improvement projects, it is also critical to have timely maintenance to preserve navigability and safety with good lockage service. Lock stalls (e.g. downtimes) affect the waterway traffic by reducing lock capacity, increasing operation costs and interfering with lockage services. In order to lower the traffic impact over time, it is important to have scheduled maintenance to keep the condition above the threshold and provide the minimum acceptable level of service. If budgets are constrained, it is necessary to optimize maintenance scheduling over a multiyear planning period.

With scheduled maintenance, the change of lock condition over time is shown in Figure 21(a). The cyclic lock condition shows that lock deteriorating continuously over time at increasing deterioration rates. The lock condition is recovered after the maintenance is carried out. A minimum allowable condition, i.e., threshold condition, is set as the lowest tolerable lock condition, under which the lockage service is still acceptable for waterway users. If a lock's condition is below the threshold, that lock might lose its functionality and maintenance cost might possibly exceed the replacement cost. If the condition reaches 0, end of the life cycle, further maintenance will not help increase the condition to serviceable level but require reconstruction or rehabilitation. Figure 21(b) shows the change of lock condition if the maintenance durations are considered.

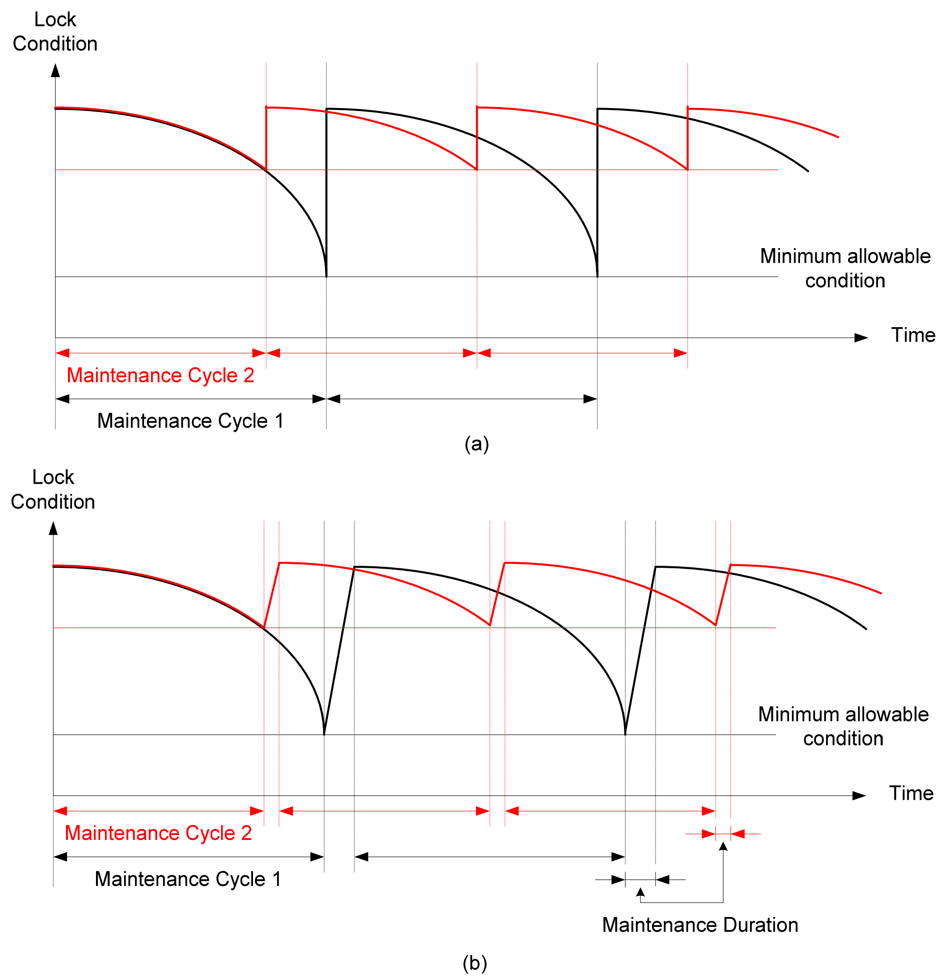


Figure 21 Lock Condition Change with Scheduled Preventive Maintenance

There may be different maintenance cycles associated with different threshold conditions. With a shorter cycle, a shorter maintenance duration is needed due to the lower deterioration, thus resulting in less traffic impact. However, it is undesirable to have very frequent maintenance, that is, with very short maintenance cycle, since there is a fixed cost associated with each maintenance project.

Since the intent of preventive maintenance is to improve the locks fromt a lower condition to higher one, the cost for preventive maintenance, that is, routine maintenance, increases over time due to the severity of deterioration at locks. As shown in Figure 22(a), scheduled preventive maintenance cost generally consists of an initial cost and a variable cost which may be proportional to the recovery level from the current lock condition to original lock condition. The variable cost increases with increasing slopes. Since we have a maintenance budget flow ($ per year), the maintenance schedule for each single lock may be determined based on the expected maintenance cost and the cumulative maintenance budget. As shown in Figure 22(b), for any single lock, its maintenance schedule is determined when the cumulative maintenance budget is available for this lock and equals the maintenance cost at that time.

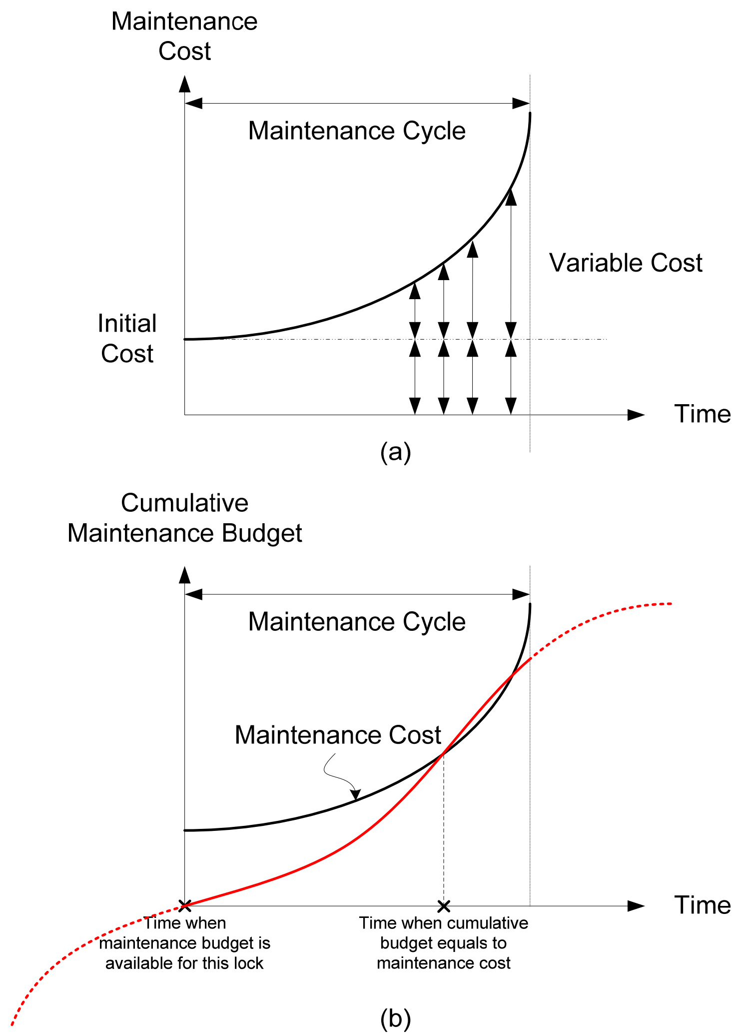


Figure 22 Maintenance Cost and Maintenance Schedule

If the i^{th} maintenance cost is function of time, $f_i(t)$, plus its initial cost $(C_{IC})_i$, for a given maintenance sequence, the time at which each maintenance task is finished can be obtained by comparing the available maintenance budgets and required maintenance costs if budget constraints are binding . Then let o_i denote the i^{th} maintenance task 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, $(C_{IC})_i + f_i(t) = \int_{t_{i-1}^o}^{t_i^o} b(t)dt$, to get time schedule t. The maintenance cost $f_i(t)$ is then determined by t. Although there may be more than one point where the two curves cross, the first intersection is selected for the maintenance schedule when budget constraints are binding.

In addition to normal deterioration over time due to natural wear and tear at locks, such as decreasing channel depths, rusting elements, crumbling guide walls, etc., it is possible to have operational failures due to mechanical or electrical failures on opening/closing valves, gates, pumping water, etc. Those random failures are related to the reliability problems. There is a tradeoff between scheduled preventive maintenance and unscheduled corrective maintenance, e.g., recovery from random failures. With earlier or more frequent preventive maintenance, the lock failures are less likely. Thus, the failure frequency and expected restoration effort are related to the lock condition. As shown in Figure 23(a) and (b), failure probability and expected failure recovery cost increases over time during the maintenance cycle. When the lock condition decreases to 0, the failure probability is 1 and it is time for reconstruction or rehabilitation. The values and curves of failure probability and expected restoration cost will be reset for each maintenance cycle.

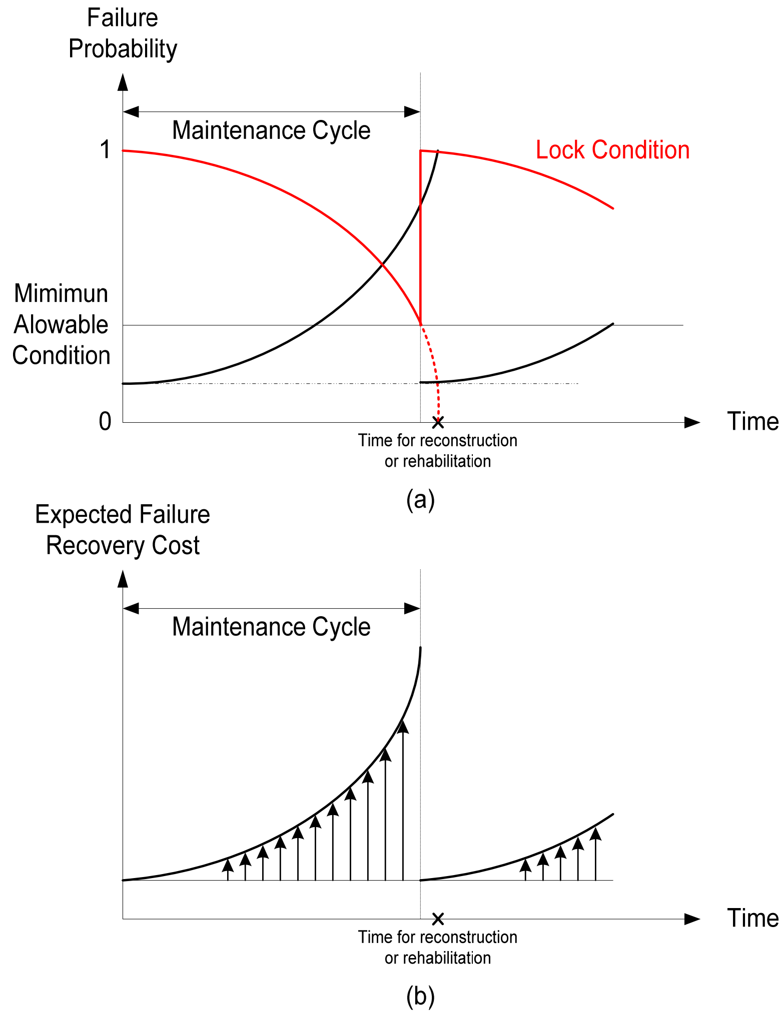


Figure 23 Lock Failure Probability and Recovery Cost

Since a simulation model is used to evaluate the maintenance plan in this study, some modifications in SIMOPT are necessary. First, the simulation model used for network-level lock maintenance should be able to handle the increasing service times due to the condition deterioration at locks, e.g. decreasing capacity. Besides, it is expected that locks/chambers must close for a certain period, i.e., lock/chamber capacity decreases to zero, if there are scheduled maintenance tasks. Since capacity changes at locks affect service quality, impacts on traffic demand should be included in the simulation model.

Finally, since there are risks of facility failures which result from insufficient preventive maintenance, the simulation should model random failures and the relevant traffic impacts. The required restoration cost will definitely take a fraction of the regular maintenance budget and change the maintenance plan. In order to simplify the maintenance application at the current stage, random failures related to the lack of maintenance are not yet considered.

A simple test is provided here to demonstrate how the optimized maintenance schedule is determined with SIMOPT. Usually network-level maintenance scheduling is performed for a specified period. In this scenario, the smallest unit considered for maintenance work is "chamber". Locks with parallel chambers require two separate maintenance tasks, respectively. Table 9 shows the information provided for network maintenance. The proposed maintenance planning covers one maintenance task for each individual chamber. All chambers have their initial conditions at the beginning of planning period. With the depreciation function, chamber acquires maintenance work before its condition reaches threshold value. Thus, threshold conditions are set as constraints which enforce the maintenance to be completed before reaching the minimum operational stage.

Table 9 Network-Level Lock Maintenance Information

Maintenance ID	Lock ID	Chamber ID	Initial Condition	Threshold Condition	Restored Capacity Ratio	Life Cycle (year)
1	0	0	0.9	0.2	1.0	15
2	1	0	1.0	0.2	1.0	15
3	2	0	0.7	0.2	1.0	10
4	2	1	0.8	0.2	1.0	10
5	3	0	0.7	0.2	1.0	10
6	3	1	1.0	0.2	1.0	10
7	4	0	0.8	0.2	1.0	10
8	4	1	0.9	0.2	1.0	10
9	6	0	0.9	0.2	1.0	12
10	7	0	0.7	0.2	1.0	12

Maintenance ID	Lock ID	Chamber ID	Initial Cost ($\times 10^6$)	Maintenance Duration (year)	Residual Capacity Ratio
1	0	0	0.9	0.02	0
2	1	0	0.7	0.02	0
3	2	0	1.0	0.02	0
4	2	1	0.8	0.02	0
5	3	0	1.0	0.02	0
6	3	1	0.8	0.02	0
7	4	0	1.0	0.02	0
8	4	1	0.8	0.02	0
9	6	0	0.9	0.02	0
10	7	0	0.7	0.02	0

Chamber closures are required for almost all the maintenance tasks. For single-chamber locks, chamber closure results in total lock closure and there is no lockage service during the closure period. For double-chamber locks, a lock is still operational even one chamber is closed for maintenance. Since the maintenance is done one by one, two chambers will not be closed at the same time. During the closure period, traffic demand is elastically responding to the delays, as discussed in previous section.

A polynomial deterioration function, $c = 1 - t^3 / L^3$, is used determine chamber condition in this test, where c is current condition, t is time lag from the first installment, L is life cycle (as shown in the upper part of Figure 24). The deterioration rate is increasing over time and chambers are completed deteriorated at the end of their life cycles. Chamber service time then varies inversely with the chamber condition. That is, along the simulation time, lock service time increases as chamber condition decreases. Whenever a lockage is performed, the required service time reflects the chamber condition at that time.

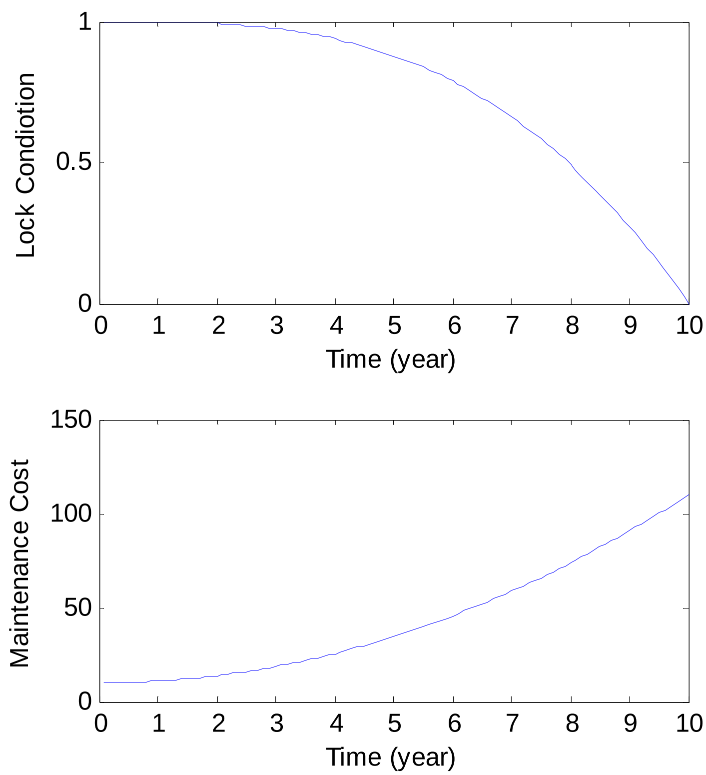


Figure 24Lock Deterioration and Maintenance Cost Functions

A parabolic function, $C = C_{IC} + t^2$, is used for maintenance cost, where C is total maintenance cost, C_{IC} is initial cost and t is the time lag (as shown in the lower part of Figure 24). The initial cost is required for performing any maintenance task, and the marginal maintenance cost is increasing over time.

In the proposed genetic algorithm, generated sequences are the candidate solutions for maintenance planning. Whenever a sequence is generated, the maintenance schedule is then determined chronologically when the available budget for the current chamber covers its required maintenance cost. As shown in Figure 25, with binding budget, maintenance schedule for any single lock is determined when maintenance cost curve first intersect with cumulative budget curve. That is, the cumulative budget is able to cover the maintenance work at that time.

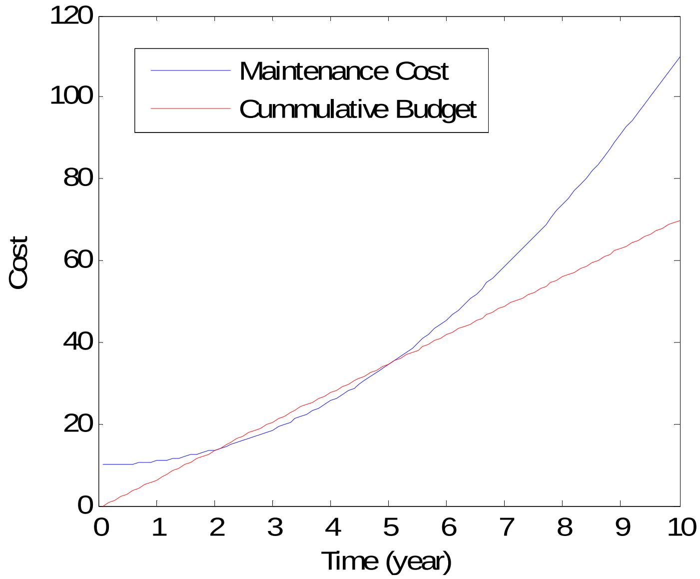


Figure 25 Maintenance Cost and Schedule

In addition, threshold constraints for each chamber are used to check a solution's feasibility. With a determined schedule, chamber conditions can be calculated based on their deterioration functions which are the functions of elapse times and initial conditions. Solutions violate the threshold constraints if any one of the chamber conditions is below its threshold condition when maintenance work is performed. Zero benefits are then assigned to all the infeasible solutions.

The optimized results from one single search are shown in Table 10. The resulting optimal sequence and schedule are presented with maintenance IDs and their relevant lock IDs and chamber IDs. Chamber conditions are described with I.C. (initial condition), C.C. (current condition), and T.C. (threshold condition). Based on the proposed maintenance schedule, required maintenance cost and current deterioration condition can be determined. As can be seen, when maintenance is scheduled, current conditions for all chambers are still above threshold conditions.

Table 10 Optimized Results for Network-Level Maintenance Planning

# of Gen.	# of Generated Sequences	# of Evaluated Sequences	Optimal Total Net Benefit ($)
134	470	1763	1,237,038,336,000

Maintenance	Lock	Chamber	I.C	Schedule	Cost	C.C	T.C

ID	ID	ID		(year)	($)		
4	2	1	0.8	0.38	0.9444	0.7584	0.2
10	7	0	0.7	0.70	1.19	0.6145	0.2
3	2	0	0.7	1.20	2.44	0.5080	0.2
7	4	0	0.8	1.70	3.89	0.5700	0.2
5	3	0	0.7	2.20	5.84	0.2964	0.2
8	4	1	0.9	2.57	7.4049	0.6249	0.2
6	3	1	1.0	2.95	9.5025	0.9743	0.2
1	0	0	0.9	3.39	12.3921	0.6713	0.2
9	6	0	0.9	3.82	15.4924	0.5209	0.2
2	1	0	1.0	4.15	17.9225	0.9788	0.2

Lock Component Level Maintenance Planning and Scheduling

Unlike network-level maintenance planning for locks/chambers, component-level maintenance planning is conducted for single locks in order to keep the functional and operational infrastructure elements of locks, such as gates, valves, and walls. Similarly, components have their life cycle and require periodic maintenance to sustain the level of service. Thus the relevant maintenance concepts about deterioration functions, initial/threshold conditions, and restoration costs applied in network-level maintenance planning can be employed here. Maintenance requires chamber closures for some components, but not others.

A simple test for a single lock is provided here to demonstrate how the optimized maintenance schedule is determined with SIMOPT. Table 11 shows the information provided for single lock maintenance. For any single lock, 10 components are assumed to be scheduled for maintenance.

Table 11 Component-Level Lock Maintenance Information

Component ID	Initial Condition	Threshold Condition	Restored Capacity Ratio	Initial Cost ($\times 10^6$)	Maintenance Duration (year)	Residual Capacity Ratio
1	1.0	0.5	1.0	0.9	0.005	0
2	1.0	0.3	1.0	0.7	0.007	0
3	1.0	0.4	1.0	1.0	0.003	1
4	1.0	0.5	1.0	0.8	0.004	1
5	1.0	0.2	1.0	1.0	0.005	0
6	1.0	0.3	1.0	0.8	0.002	1
7	1.0	0.4	1.0	1.0	0.008	0
8	1.0	0.5	1.0	0.8	0.007	1
9	1.0	0.2	1.0	0.9	0.003	1
10	1.0	0.3	1.0	0.7	0.005	0

At the opening of a lock, initial conditions for all components are 1.0, perfectly new conditions. The threshold conditions, maintenance cost and maintenance durations vary among different components. Some components, such as gates and valves, play key roles in lockage process, thus requiring lock closures for maintenance work. Some components may partially or not affect lockage service and resulting lock capacity. In this test, component deterioration functions and maintenance cost functions are similar to those used in previous section.

The optimized results from 10 searches for component-level maintenance planning are shown in Table 12. All sequenced are ended with maximum net benefits of $1,443,639,744,000. Since some component maintenance does not require any service interruption, their positions in scheduled sequence do not affect the overall evaluated system performance. In addition, if traffic is low, some very short closure for component maintenance will not affect the travel delay and benefit measurement as well. Therefore the positions of some short-term closures in a sequence are not significantly important.

Table 12 Optimized Results for Component-Level Maintenance Planning (Low Traffic)

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
2	2	2	2	2	2	2	2	2	2
10	10	10	10	10	10	10	10	10	10
5	8	6	9	1	6	5	6	8	9
3	1	5	5	8	5	6	8	9	6
4	4	8	3	4	8	4	7	7	7
8	3	9	1	3	3	3	9	3	1
9	5	7	7	9	7	1	3	4	5
6	7	1	6	6	4	8	5	6	8
1	9	3	8	5	9	9	4	1	3
7	6	4	4	7	1	7	1	5	4

A higher traffic is applied to retest this scenario. Table 13 shows the optimized results from 10 searches for component-level maintenance planning under higher traffic. All sequences are ended with maximum net benefits of $1,443,639,744,000. As discussed above, since no service interruption is required for some component maintenance, their positions in scheduled sequence do not affect the overall evaluated system performance. Short-term closures do not affect sequences as well.

Table 13 Optimized Results for Component-Level Maintenance Planning (High Traffic)

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
2	2	2	2	2	2	2	2	2	2
10	10	10	10	10	10	10	10	10	10
5	9	6	6	8	6	3	9	3	6
3	6	5	5	1	5	5	7	7	7
4	3	8	3	6	8	6	4	1	3
8	7	9	7	3	3	8	1	6	8
9	4	7	4	9	7	7	5	9	9
6	8	1	9	5	4	1	6	5	5
1	5	3	8	7	9	9	3	4	1
7	1	4	1	4	1	4	8	8	4

Conclusions

The development of NETS methods for evaluating, prioritizing and scheduling waterway projects is continuing at the University of Maryland. A testbed waterway model (SIMOPT) that combines simulation and optimization has been developed. It employs simulation to evaluate project implementation schedules found through an evolutionary search process by Genetic Algorithms. Thus it solves the problems of evaluating, selection, sequencing and scheduling waterway improvement projects, and 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.

In order to enhance the search efficiency of the optimization model and consider additional constraints, some improvements in investment optimization methods are tested in SIMOPT. The improved optimization models are intended to work with the next generation NaSS waterway simulation model which is developed under USACE’s NETS program. As a testbed, SIMOPT was first (in Phase I) modified to consider project construction time and capacity reductions during construction, to avoid duplicate simulations of similar or identical solutions (“solutions” consist here of project implementation schedules) and to consider mutually exclusive projects within any locks. Secondly in Phase II, additional constraints on project precedence and regional budgets can now be imposed. A simple evaluator was proposed to replace complex and time consuming simulation model while investigating search efficiencies among different genetic operators and their combinations.

In the current phase (Phase III), pre-screening rules are also used to avoid expensive simulation of unpromising or infeasible solutions. Recent improvements allow the investment model to consider multiple projects at the same location with different implementation times as well as consider project construction time and capacity reduction during the construction period, with demand elastically responding to the service quality. With elastic demand, the optimization problem is changed to maximize net system benefits rather than minimize total cost. In addition, tradeoffs between construction time and cost are considered while sequencing and scheduling project alternatives.

When considering multiple projects, which could be independent improvement projects as well as dependent expansion projects, at the same location with different implementation times, the cost relations among dependent projects should be considered. In order to cope with dependent projects, project precedence constraints, similar to lock precedence constraints, are applied to restrict the sequence of those expansion projects. With two kinds of precedence constraints, project and lock precedence constraints, which define an order of succession among projects, there are large fractions of infeasible solutions which are prescreened and discarded before being simulated.

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 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. If demand is fixed, a total cost function would suffice to compare scenarios or drive an optimization process. However, if the demand can be affected by simulated decisions, an objective function of maximizing a net benefit rather than minimizing total cost is used. Different results show how shippers' response to any possible service interruption due to capacity reduction during the construction time has significant effects on overall network performance and resulting optimal project sequence.

When considering tradeoffs between construction time and cost, mutually exclusive constraints for combination of construction time and cost are developed. That is, only one combination of construction time and cost can be selected if there are mutually exclusive tradeoffs for the same project. Thus the newly defined chromosome contains a full list of mutually exclusive tradeoffs. Solutions with full lists of projects are not feasible when we allow at most one combination per project. Therefore, a "refining" technique is applied to create feasible solutions with lists of tradeoffs having at most one tradeoff per project. The modified SIMOPT is able to solve the problem of sequencing and scheduling project with tradeoff consideration among construction time and cost.

The simulation-based optimization method is also being applied to analyze other waterway problems, such as network-level and component-level maintenance planning. With appropriately defined project costs and performance changes resulting from projects, the modified SIMOPT is now able to optimize network or lock maintenance schedule.

GAs search performance may be improved by creating weighted sequences, developing smarter problem-specific genetic operators and applying parallel computing techniques. The feasibility of applying parallel computing to speed up the optimization process has been tested on various scenarios for scheduling waterway improvement projects. The value of parallel computing in simulation-based optimization increases significantly as the problem size and complexity increase. It greatly reduces the time needed to obtain solutions as well as the memory load for any individual computer.

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 Phase I, April 2006

Wang, S and Schonfeld, P., "Genetic Algorithms for Selecting and Scheduling Waterway Projects (Phase 2)", NETS Report Phase II, November 2006

Yang, N and Schonfeld, P., "Parallel Computing in Waterway Optimization", TSC Report, University of Maryland

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

Wang, S. and Schonfeld, P. (2007), "Demand Elasticity and Benefit Measurement in a Waterway Simulation Model", Transportation Research Board 2007 Annual Meeting (CD-ROM 07-2579)

Tien, S. and Schonfeld, P. (2006), "Duration versus Costs Tradeoffs in Work Zone Optimization", Transportation Research Board 2006 Annual Meeting (CD-ROM 06-2579)

Appendix GA Phase 3 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. Two extra subtasks (Task 1.3 and Task 2.2) were added during the contract period.

Task 1 Genetic algorithm

1.1 Create “smart” operators specific for NaSS problem

1.2 Explore parallel processing options

1.3 Create “weighted” sequences during search process

Task 2 Evaluation / Simulation model

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

2.2 Estimate total use benefit from simulation model

Task 3 Project selection / sequencing / scheduling

3.1 Consider multiple alternatives at the same location which may be implemented at different times

3.2 Consider the tradeoffs between construction time and cost

3.3 Consider construction times and capacity reductions during construction periods

Task 4 Other applications

4.1 Adapt GA for network level maintenance planning and scheduling

4.2 Adapt GA for lock component maintenance planning and scheduling

Task 5 Continued participation on NaSS team

5.1 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. (Phases 2.2 and 3)

5.2 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. (Phases 2.2 and 3)