Prioritizing Interrelated Road Projects Using Metaheuristics

Elham Shayanfar[1]; Arezoo Samimi Abianeh[2]; Paul Schonfeld[3]; and Lei Zhang[4]

Abstract: Projects are considered interrelated when their benefits or costs depend on which other projects are implemented. The timing of such projects may also complicate their analysis. Selection and scheduling of interrelated projects is a challenging optimization problem that has many applications in various fields, including economics, operations research, business, management, and transportation. The goal is to determine which projects should be selected and when they should be funded in order to minimize the total system cost over a planning horizon. Finding the optimal solution for such problems often requires extensive evaluation of possible solutions because of the complex nature and noisy surface of their solution space. This paper applies three metaheuristic algorithms including a genetic algorithm (GA), simulated annealing (SA), and Tabu search (TS) in seeking efficient and consistent solutions to the selection and scheduling problem. These approaches are applied to a special case of link capacity expansion projects to showcase their functionality and compare their performance. The paper's main contributions are to (1) compare three metaheuristics for this problem in terms of solution quality, computation time, and consistency; (2) consider explicitly the supplier costs as well as user costs in the formulated objective function; and (3) enhance some simplifying assumptions from previous studies by recognizing that candidate projects may not remain economically justifiable throughout the analyzed period. It is found that a GA yields the most consistent solution with the least total cost while SA and TS approaches excel in terms of computation time. **DOI: 10.1061/(ASCE)IS.1943-555X.0000293.**

Introduction

As traffic increases and links become congested, passenger and freight movements experience increasing travel times and delays. One obvious solution to this problem is to construct new lanes and create additional capacity on the highly congested links. Then it must be determined which links should be selected, in what order they should be implemented, and when they should be funded in order to minimize costs. One simple idea is to identify congested links and prioritize them according to their congestion level, i.e., volume/capacity ratio. However, even after adjusting for the relative costs of links, this approach does not yield the best solution because it disregards the interrelations among network links. In fact, changes in one link affect the flows on others, and removing bottlenecks from some links may shift them elsewhere in the network. Thus, in sequencing a set of improvement projects, their interrelations should be considered. The selection and scheduling of projects with consideration of their interrelations is a challenging optimization problem, but its solution is very valuable because it has applications in various fields, including economics, finance, operations research, development, industrial engineering, and business administration.

[1]Ph.D. Student, Dept. of Civil and Environmental Engineering, Univ. of Maryland, College Park, MD 20742. E-mail: eshayan@umd.edu

[2]Ph.D. Student, Dept. of Civil and Environmental Engineering, Univ. of Maryland, College Park, MD 20742. E-mail: asamimi@tamu.edu

[3]Professor, Dept. of Civil and Environmental Engineering, Univ. of Maryland, 1173 Glenn Martin Hall, College Park, MD 20742 (corresponding author). E-mail: pschon@umd.edu

[4]Associate Professor, Dept. of Civil and Environmental Engineering, Univ. of Maryland, 1173 Glenn Martin Hall, College Park, MD 20742. E-mail: lei@umd.edu

Note. This manuscript was submitted on December 31, 2014; approved on November 4, 2015; published online on January 25, 2016. Discussion period open until June 25, 2016; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Infrastructure Systems***, © ASCE, ISSN 1076-0342.

Conceptually, the first step of a project planning problem is the project evaluation, which identifies candidate projects and evaluates their merits, often in terms of their benefits and costs. In a second step, projects are selected from among the considered set for implementation. After evaluating and selecting a set of projects for improvement, a third step determines the order of projects; finally, a fourth step determines the deadline for completion under budget constraints (Wang and Schonfeld 2005). Project selection and scheduling easily constitute a large optimization problem whose feasible domain increases rapidly as the number of considered projects in the system grows. In considering a set of improvement projects for a given network, the objective is to find a project implementation sequence that minimizes the total system costs or maximizes the net benefits over the analyzed period. To date, several methods have been developed for scheduling interrelated projects. However, the number of studies on this topic is relatively low.

This study applies and compares three alternative metaheuristic algorithms for solving the problem of selecting and scheduling interrelated projects. These three algorithms are a genetic algorithm (GA), simulated annealing (SA), and Tabu search (TS). This study also demonstrates how a relatively simple method, namely, a traffic assignment model, can be efficiently used as the objective function for such an optimization problem and thereby compute the relevant interrelations among many projects that are implemented at various times. However, more complex methods for evaluating the objective functions, such as microscopic simulations, can also be combined with the same metaheuristic algorithms for optimizing the project selection and schedule.

The motivation of this line of research is to develop a general optimization framework for selecting, sequencing, and scheduling interrelated alternatives. The work presented in this paper advances this objective and contributes to the previous research in several ways. First, three metaheuristics are applied to explore and compare different solution approaches for the selection and scheduling of interrelated alternatives. Second, the objective function is reformulated to consider the total system cost, including supplier costs and user costs. Third, the algorithms' assumptions are further modified to account for the possibility that candidate projects may become economically unjustifiable after the implementation of previous projects. This may occur owing to project interrelations and the possibility that the cost savings from completing a project are affected by earlier project implementations.

The three metaheuristic algorithms are demonstrated through a case study and comparison of their performances in terms of solution quality, computation time, and consistency. The comparative analysis is useful in deciding which algorithm to use in different circumstances. Generally, the methodology presented in this paper should also be applicable to other prioritization and resource allocation problems with interrelated alternatives.

Literature Review

In portfolio management, interrelations between choices (stocks) were identified and modeled as early as the 1950s in pioneering work by Markowitz (1952). This study proposed a quadratic program formulated as a multiobjective function to minimize the sum of purchase costs and interrelated risks. The consideration of project interdependence significantly complicated the model's structure because the combined costs and benefits for a set of projects were no longer equal to the sum of the costs and benefits, respectively, of individual projects. To deal with interdependencies among choices, a dependence matrix was introduced to capture the pairwise or n-way interrelations among alternatives. In portfolio optimization, the use of a dependence matrix was convenient in modeling interdependence between choices, and its variants are still used in recent works, for example, Durango-Cohen and Sarutipand (2007) and Bhattacharyya et al. (2011). However, estimation of a dependence matrix is difficult and its manipulation is computationally arduous when the project space grows (Disatnik and Benninga 2007). Moreover, the pairwise dependency between projects, as well as three-way and higher-order dependencies, is insufficient to model the complex relations among infrastructure development projects and difficult to estimate.

A different approach to analyzing interdependence alternatives is to adopt complete system models, such as queueing approximations (Jong and Schonfeld 2001), equilibrium assignment (Tao and Schonfeld 2005), microsimulation (Wang and Schonfeld 2008), and neural networks (Bagloee and Tavana 2012), to model the interrelations. These models are generally applicable for modeling truly complex systems and relations among infrastructure developments. The remainder of this section presents an overview of some recent studies applying these models.

Bouleimen and Lecocq (2003) developed a SA algorithm for the resource-constrained project scheduling problem. The objective of this model was to minimize total project duration. Tao and Schonfeld (2005) developed a Lagrangian heuristic for selecting interdependent projects under cost uncertainty. They developed a GA for solving the Lagrangian problem and applied equilibrium assignment to evaluate the objective function. Mika et al. (2005) proposed two local search metaheuristics, namely, SA and TS, to solve the multimode resource-constrained project scheduling problem with discounted cash flows. The objective was to maximize the net present value of all cash flows. Wang and Schonfeld (2005) developed a waterway simulation model for evaluating lock operations over long analysis periods and then solved the problem of selecting, sequencing, and scheduling interdependent projects with a GA. Milatovic and Badiru (2004) proposed a methodology for the mapping and scheduling of interdependent and multifunctional project resources. Their methodology included an activity scheduler and a resource mapper. The first procedure prioritized and scheduled activities based on their attributes, whereas the latter considered resource characteristics and mapped the available resource units to the scheduled activities.

Tao and Schonfeld (2007) developed island models, which are a variant of traditional GAs, to optimize the selection and scheduling of interrelated projects under resource constraints. Dueñas-Osorio et al. (2007) studied the interdependence response of network systems to internal or external disruptions. They established interdependencies among network elements based on geographical proximity. Their work indicated that responses that are detrimental to networks are greater when interdependencies are considered after disturbances. Szimba and Rothengatter (2012) extended the classical benefit–cost analysis by incorporating the interdependence among projects within an investment package. They addressed the interdependence problem by introducing a heuristic method to solve the large-scale problem with numerous projects. Bagloee and Tavana (2012) formulated the prioritization problem as a Traveling Salesman Problem (TSP) and incorporated a neural network (NN) to assess project interdependence. A heuristic algorithm with hybrid components was then used to search for the longest (most beneficial) path in the NN as a solution to the TSP. Li et al. (2013) proposed a hypergraph knapsack model to maximize the overall benefits for a subcollection of interdependent projects. Chen et al. (2015) proposed a surrogate-based optimization framework to simultaneously find both optimal capacity expansions of existing links and new link additions. The upper level aimed to minimize the network cost, while the lower level used a dynamic user-optimal condition that could be formulated as a variational inequality problem.

A literature review reveals a number of studies that adopt metaheuristics on the basis of evolution strategies, such as GAs, TS, or scatter search. Although metaheuristics have been widely used in network design modeling (Devika et al. 2014; Sicilia et al. 2015; Jeon et al. 2006; Xu et al. 2009), they are rarely investigated and compared for problems with interrelated alternatives to identify their strength and weaknesses. Such an exploration would be especially useful in deciding what algorithm is preferable in various circumstances.

Development of Evaluation Model

Traffic assignment models are simple methods for assessing the traffic-related attributes for unsaturated networks. These attributes include travel time, traffic flow, and volume-capacity ratio over all the links in a network. This information is important for estimating the cost savings from capacity improvements and therefore supports a proper evaluation method for the selection, sequencing, and scheduling of projects. The aforementioned cost savings mainly pertain to the value of travel time reduction for users, which is explained in detail in subsequent sections. These cost savings are obtained by rerunning the traffic assignment model at different stages of the metaheuristic algorithms to compute the objective function and compare solutions.

It should be noted that the use of traffic assignment in this paper provides a relatively simple and easy way to evaluate solutions, i.e., computing the values of their objective functions. More precise evaluations for such a road network capacity expansion problem may be obtained using a microscopic simulation model, but at considerably higher computation cost. Other problem-specific evaluation methods would be needed for other applications.

This paper employs the *convex combination algorithm* (Frank and Wolfe 1956) to evaluate link expansion projects upon their implementation in a network. This method is an iterative algorithm applicable to nonlinear programming problems with convex objective functions and linear constraints and is widely used for solving the traffic equilibrium problem (van Vliet 1987; Ahipasaoglu et al. 2008). Starting with an initial flow x, the search direction at each iteration is determined by solving a linear approximation of the objective function, determining the step size, and moving in that direction. The algorithm eventually stops when the convergence criterion, which is based on the similarity of two successive solutions, is satisfied.

Problem Formulation

The objective function for prioritizing transportation investments has a nonconvex surface. Moreover, the scope of the problem may be beyond the capability of typical mathematical optimization methods since the problem size grows very fast with the number of candidate projects n_p. The solution space for all possible sequences of projects is

$$\sum_{i=0}^{n_p} \frac{n_p!}{(n_p - i)!i!} \cdot i! = \sum_{i=0}^{n_p} \frac{n_p!}{(n_p - i)!} \qquad (1)$$

Consequently, heuristic methods have gained popularity among researchers for solving such complex problems. This paper explores three metaheuristic methods, including GAs, SA, and TS, which have often been found to be effective in finding near-optimal solutions for challenging optimization problems. The planning problem is to determine which links should be expanded in what order and when each project should be completed over the planning horizon T. The objective is to minimize the total cost, which consists of the total road user cost and the total supplier cost, subject to a budget constraint.

Jong and Schonfeld (2001) formulated this problem by defining the decision variables as the completion time of projects. In this formulation the budget constraint is defined as follows:

$$\sum_{i=1}^{n_p} c_i x_i(t) \leq \int_0^t b(t)dt, \quad 0 \leq t \leq T \qquad (2)$$

$$x_i(t) = 0 \quad \text{if } t < t_i$$

$$x_i(t) = 1 \quad \text{if } t > t_i$$

where t_i = time when project i is completed; and $x_i(t)$ = binary variable specifying whether project i is completed by time t. It should be noted that the set of all t_i eventually determines the schedule of projects. This occurs because under a limited budget, which is continuously distributed over time, it is reasonable to fund and finish each project one at a time knowing that there are always some justifiable projects awaiting funding, and the system gains an immediate benefit as soon as a project is completed. In other words, funding multiple projects simultaneously delays their completion time and, hence, delays the cost savings of capacity improvements. Thus, under limited budget flow it is desirable to fund and complete one project at a time and avoid funding overlaps (although not necessarily construction time overlaps). As a result, the schedule of each project is easily determined from a given sequence by considering the budget flow. The idea is that each project is funded immediately after the preceding one is finished and is completed as soon as the available cumulative budget reaches the project cost. To date, other studies have have haveassumed that all candidate projects that are initially justified economically remain so until the end of the studied period. However, owing to project interdependencies, the cost savings from completing a project may change over time. It is also possible that initially unjustifiable projects turn out to be desirable later. To tackle this problem, projects with unacceptable marginal benefit–cost ratios are removed temporarily from the sequence list during the evaluation stages and replaced by other justifiable projects. At later stages, the removed projects may reenter the sequence after their marginal benefits outweigh their marginal costs.

As stated earlier, the objective function minimizes the total supplier cost and user cost over the planning horizon subject to a budget constraint. The user cost is defined as the system delay multiplied the by value of time, and the supplier cost is the present value of all project costs. Unlike in some previous studies, in which the cumulative costs of projects are implicitly determined by the budget constraint, the cost of projects must be included in the objective function used here since not all selected projects are guaranteed to be implemented during the analyzed period. Therefore, the objective function that is first introduced in Jong and Schonfeld (2001) is modified as follows:

$$\min Z = \sum_{j=1}^{T} \left\{ \frac{v}{(1+r)^j} \sum_{i=1}^{n_l} w_{ij} \right\} + \sum_{i=1}^{n_p} \frac{c_i x_i(t)}{(1+r)^t} \qquad (3)$$

where w_{ij} = waiting time over link i in year j; c_i = present value of cost of project i; n_p, n_l, and v = number of projects implemented, total number of links, and value of time, respectively; and r is the interest rate.

Development of Optimization Models

The main objective of this paper is to compare the performance of three metaheuristic methods (GA, SA, TS) in solving the selection, sequencing, and scheduling of interrelated projects. The common elements of the three approaches are as follows. First, the solutions are represented by the sequence of projects in which projects are implemented. Second, the objective function with all three approaches minimizes the present worth of the total user and system costs subject to a cumulative budget constraint, defined in the previous section. Third, all three algorithms incorporate a solution feasibility test to check the justification of adding a new project to the project list. This is done by estimating the marginal benefit and the marginal cost of adding a new project to the sequence and calculating the resulting benefit–cost ratio. Any unjustified project is discarded before the next project in the list is similarly considered in order to maintain the feasibility of solutions. Furthermore, the implementation time is checked so as not to exceed the planning horizon, and the projects scheduled beyond the horizon are deleted from the accepted sequence. This makes intuitive sense because in real-world applications, there are usually more desirable projects than the available budget allows for during a planning time period, and one must choose from among a subset of candidate projects and discard the rest. If justified projects are always available, then the budget constraints are binding and optimal sequencing decisions also determine the optimal timing of projects. Fourth, two stopping criteria, namely, the number of iterations and running time, are tested for all the algorithms.

Genetic Algorithm

A GA is a metaheuristic method that mimics the process of natural selection and is a successful optimization method in a wide range of fields. GAs obtain a set of possible solutions called the population. Each individual in the population is specified by a string of encoded genes called a chromosome. In this process some individuals are selected to reproduce offspring, and since each individual has a probability of selection according to its fitness value, better ("fitter") solutions have a higher probability of being selected. Then the selected solutions are processed through a series of crossover and mutation operators that create offspring and change their attributes while maintaining the diversity of the population. Designing an appropriate GA can lead to an optimal or near-optimal solution.

In general, solutions of GAs are mostly represented by binary digits, and the initial population is generated randomly. In this paper, each individual in a population is defined by a string, including a sequence of numbers, each corresponding to a specific project. In addition to random-order solutions, two other methods—greedy-order solutions and bottleneck-order solutions—are used to create the initial population (Jong and Schonfeld 2001). In greedy-order solutions, projects are selected based on their benefit–cost ratio, regardless of their interrelations. In bottleneck-order solutions, projects are ranked based on the link volume–capacity ratio, which describes the congestion severity over a link. This assumes that more congested links should have higher priority for implementation.

The fitness function is considered equivalent to the value of the objective function [net present value (NPV) of total cost] and is computed through the traffic assignment model. The selection probability is generally based on the value of the objective function in maximization problems. Therefore, in minimization problems, the selection probability varies inversely with the objective function value. However, to prevent some undesirable properties of prematurity, a ranking method is applied instead (Wang 2001). In this method, the population is sorted with nonlinear ranking from best to worst. Then the selection probability of each chromosome is assigned according to its exponential ranking value considering a lowest fitness value equal to one (Michalewicz 1995). Let q be the selective pressure $\in [0, 1]$; the selection probability is defined as follows:

$$P_i = c \times q(1-q)^{i-1}, \qquad c = 1/[1-(1-q)^{\text{PopSize}}] \tag{4}$$

Next, a roulette-wheel approach is incorporated to select appropriate parents based on their selection probabilities (Michalewicz 1995). Then a crossover and a mutation operator are applied to reproduce offspring and create a new population. Common methods of mutation and crossover are not very efficient for sequencing problems since they construct many infeasible solutions with repetitive project numbers in one sequence. To avoid producing such solutions, some other genetic operators are employed to solve the project sequencing problem. These crossover and mutation operators consist of partial mapped crossover, position-based crossover, order crossover, insertion mutation, and swap mutation (Goldberg 1989; Gen and Cheng 1997). The reproducing process randomly selects one operator and applies it to the selected parents.

Simulated Annealing

Simulated annealing is a probabilistic metaheuristic method for the global optimization of an objective function, which may possess several local optima. The algorithm that was introduced independently by Kirkpatrick et al. (1983) and Černý (1985) was inspired by a process that involves the heating and gradual cooling of a material to reach a minimum energy configuration. Starting from an initial solution (S), the value of the objective function is calculated for the new solution in the neighborhood $f(S')$ where $f()$ denotes the objective function value for a solution. Then the algorithm attempts to move to a neighborhood solution (S') based on specified criteria. In minimization problems, a transition to a new solution is immediately allowed when $\Delta = f(S') - f(S) < 0$. However, a transition to the new solution is also permitted based on the probability function $\exp(-\Delta/\text{Tmp})$, where Tmp (temperature) is a control parameter. Allowing for such transitions guarantees the diversification of the solutions and enables SA to escape a local optimum in a search for the global optimum. After each iteration, the parameter Tmp decreases within a *cooling function* ($\text{Tmp} = \text{Tmp} \times \alpha$), where α is a constant parameter by which the temperature decreases after each iteration. The algorithm finally stops when the stopping criterion is satisfied. In the developed SA, the neighbor solutions are produced by using the *Project Shift* operator in which project j is randomly selected from the project list and project i is randomly selected from the first predecessor and successor of project j. The two selected projects switch positions, and the new solution is evaluated for possible transition.

One of the most important steps in SA is to set an appropriate initial temperature. In this paper, a recursive formula proposed in Ben-Ameur (2004) is used to assess an initial value for the temperature Tmp as follows:

$$\text{Tmp}_{n+1} = \text{Tmp}_n \left\{ \frac{\ln[\hat{\chi}(\text{Tmp}_n)]}{\ln(\chi_0)} \right\}^{1/\rho} \tag{5}$$

where χ_0 = desired acceptance probability; ρ = real number ≥ 1; and $\hat{\chi}(\text{Tmp}_n)$ is determined by generating a set of positive transitions P (a transition in which the objective function increases), storing the corresponding objective functions $[f(S'), f(S)]$, and using the following equation:

$$\hat{\chi}(\text{Tmp}_n) = \frac{\sum_{p\in P} \exp\left[-\frac{f(S')_p}{\text{Tmp}}\right]}{\sum_{p\in P} \exp\left[-\frac{f(S)_p}{\text{Tmp}}\right]} \tag{6}$$

The iteration stops as $\hat{\chi}(\text{Tmp}_n)$ becomes sufficiently close to χ_0 and the value of Tmp_n can be used as a good approximation for the initial temperature.

Tabu Search

Tabu Search is a metaheuristic originated by Glover (1986) that employs neighborhood search and enhances it by using a memory structure that avoids visiting previously investigated solutions. To achieve this goal, the method records recent *moves* and stores them in a *tabu list*, preventing the algorithm from retracing these moves. This insures that new regions of the solution space will be explored in the search for the global optimal solution.

Similarly to SA, the neighbors of current solutions are generated by swapping the position of projects in the project sequence. A move is defined as the position number in the project list selected for swapping. After a move is made, its reverse enters the tabu list, while the oldest existing move exits the list. All moves that exist in the list remain tabu for a specified number of iterations, called *tabu tenure*. However, it is possible that a tabu move will reach a nonvisited solution. To avoid the possibility of overlooking a better solution, an *aspiration criterion* authorizes a tabu move only if this move leads to a solution with the best objective value visited so far.

Case Study

The Sioux Falls network illustrated in Fig. 1 is selected for demonstrating the performance of the proposed algorithms. This is not considered a realistic network since it mainly includes the city's major arterial roads and omits many characteristics of its transportation

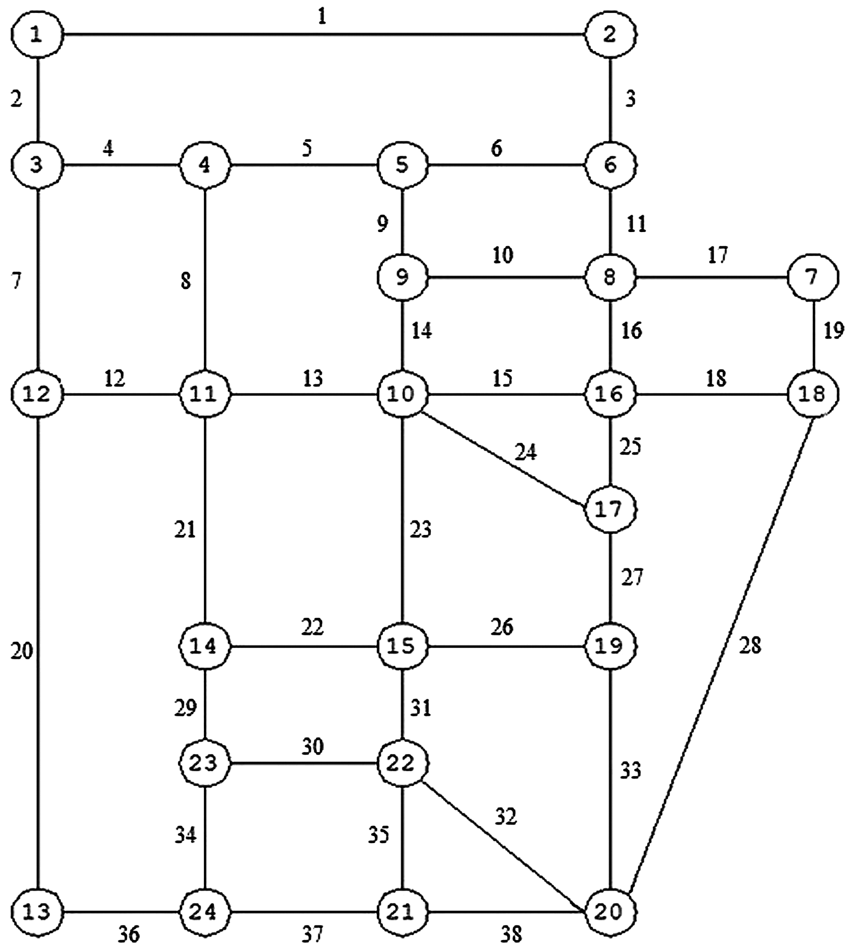

Fig. 1. Sioux Falls network

system. However, it has been used widely to examine and compare studies on networks (LeBlanc et al. 1975). Sioux Falls city has a total area of 190 km^2 and is located in eastern South Dakota, approximately 24 km west of the Minnesota border. This network consists of 24 nodes and 38 links. The links in Fig. 1 are numbered from 1 to 38, each one representing traffic in both directions. Since the demand matrix used in this study is symmetric for all origin and destination nodes, each link expansion improvement is assumed to be implemented in both directions between the two connected nodes. This assumption is also justified economically because it saves costs thanks to the joint use of resources and construction equipment. The network inputs include the O/D demand table, link capacity, link length, and free flow travel time. These are documented in Shayanfar and Schonfeld (2015). After running the traffic assignment model, the critical lanes with high volume–capacity ratios were identified as an initial set of candidate projects. After identifying an initial collection of candidates, all projects are further investigated through a benefit–cost analysis to identify and rank economically beneficial projects. Eventually, the finalized set of candidate projects includes links with the most severe congestion level (i.e., highest volume–capacity ratio) and benefit–cost ratios greater than one: {2, 3, 4, 8, 9, 11, 16, 14, 13, 15, 12, 21, 36, 22, 25, 27, 35, 37, 30, 34}.

It is assumed that each project improvement adds one lane equivalent to 700 vehicle/h additional capacity to each link, while the equivalent annual cost of each lane expansion is assumed to be \$1,800,000/km. Moreover, the value of time is set at \$15/h, and the interest rate is assumed to be 10%. It should be noted that this study does not account for delays during lane closures. The main cost saving of link expansion projects is the reduced travel time for all the users. These travel time reductions can be computed using the traffic assignment model by comparing the total system travel time before and after project implementation. Table 1 shows a sample of the travel time savings from separate implementations of projects in the network. The second column presents the initial link travel times prior to project implementation, while Columns 3–7 present the travel time reductions for single projects. Positive values indicate travel time reductions, while negative values show increases in travel time due to network interdependencies. (Conceptually, if the capacity increases in one link in the network, the congestion and average travel times tend to increase in unchanged links that are "in series" with it while decreasing in its "parallel" links.) The bolded numbers indicate the travel time changes in the location of the expanded links. These numbers are relatively higher since the expanded links gain direct benefits following project implementation. Notably, the sum of all the cells in one column is not equal to the travel time changes on the links that are being expanded. This, in effect, confirms the interrelation among links and the possible shifting of bottlenecks to nearby links. The last column shows the reductions in the cost of travel times from implementing two projects together. This column shows that the total system delay savings (25.216 min/veh) differs from the sum of cost savings for two individual projects ($9.753 + 15.656 = 24.409$), emphasizing that the cost savings of multiple projects are not linear summations of their individual savings.

Table 1. Travel Time Reduction due to Link Expansion

Link	Link travel time without projects	Link travel time reduction (min/veh)					
		Expanding 2	Expanding 3	Expanding 4	Expanding 8	Expanding 9	Expanding 2 and 3
1	3.6	0.012	−0.036	0.016	0	−0.008	−0.046
2	10.042	**2.23**	1.44	−0.816	−0.124	0.062	**3.318**
3	20.712	4.676	**11.424**	6.936	1.316	−0.684	**14.48**
4	9.1	−0.692	1.14	**1.854**	−0.538	−0.288	1.472
5	3.258	−0.186	0.332	−0.12	0.03	−0.412	0.328
6	6.002	−0.688	0.68	−1.052	−0.544	−1.702	0.734
7	5.236	0.032	0.082	0.064	0.042	0.082	0.104
8	14.748	−0.076	−8.442	−8.796	**5.606**	4.102	−4.616
9	14.78	0.106	2.428	1.866	1.766	**4.784**	2.026
…	…	…	…	…	…	…	…
37	6.574	0.18	0.378	0.328	−0.38	−0.346	0.278
38	2.861	−0.006	0.082	−0.173	0.100	0.067	0.158
Total travel time savings		9.753	15.656	8.054	13.632	18.037	25.216

Note: Bold numbers indicate the travel time changes in the location of the expanded links.

Results

This section analyzes the results obtained from the GA, SA, and TS in terms of (1) the quality of the final results, (2) the computational speed, and (3) the consistency of the optimized solutions. Table 2 describes the parameter values for each algorithm. The paper further compares each algorithm in the aforementioned categories.

Quality

Each metaheuristic is tested for 50 replications, each encompassing 150 iterations, which is considered a reasonable number of iterations for comparison purposes since all 3 algorithms reach a stable convergence within 150 iterations. The best results out of 50 replications in terms of the final value for the objective function (minimum total cost) are extracted and plotted in Fig. 2, which presents the performances of the GA, SA, and TS. The results suggest that after all the algorithms are run for a long enough time to obtain stable convergence, the GA performs better in terms of finding solutions with lower objective functions and TS performs better than SA. In this case, the present values of total system costs are as follows: GA = $15,009 million, SA = $15,028 million, and TS = $15,016 million, which are remarkably close. Furthermore, the resulting selection, sequencing, and scheduling of projects are presented in Table 3, which also presents a comparison between the metaheuristic solutions and the solution ranked according to congestion severities. The severity-ranked solution has a total cost of $15,605 million, while the solutions obtained from the metaheuristics have lower total costs, highlighting the significance of project interrelations and the importance of estimating their effects. In fact, the present worth of the total costs is reduced by $596 million, $577 million, and $589 million when applying GA, SA, and TS, respectively, compared to the severity-ranked order.

Table 2. Parameter Values for GA, SA, and TS

Algorithm	Parameter	Value
GA	Neighborhood size	100
	Number of samples for initial temperature	0.5
	Cooling ratio	0.8
	Trial count	20
	Move method	Swap
SA	Population size	20
	Mutation rate	0.5
	Crossover rate	0.5
	Selective pressure	0.1
	Sampling mechanism	Roulette Wheel
TS	Neighborhood size	100
	Tabu tenure	3
	Trial count	20
	Move method	Swap

Computation Time

The metaheuristic results may also be compared in terms of computation time. For this purpose, the average running time per iteration is computed for all algorithms as follows: GA = 87.5 s, SA = 19.3 s, and TS = 37.7 s. The results indicate that the GA has the most and the SA the least computation time. This is due to the relative complexity and multiple operators incorporated in the GA. However, as discussed in the previous section, if the running time is sufficiently large for all algorithms to reach convergence, then GA yields slightly better solutions than SA and TS.

In this section, the sensitivity of computation time to the problem size (i.e., number of candidate projects) and planning horizon for the three algorithms is also evaluated. According to Fig. 3, the computation time of all algorithms grows logarithmically as the planning horizon increases, whereas Fig. 4 shows an exponential growth of computation time as the problem size grows. These

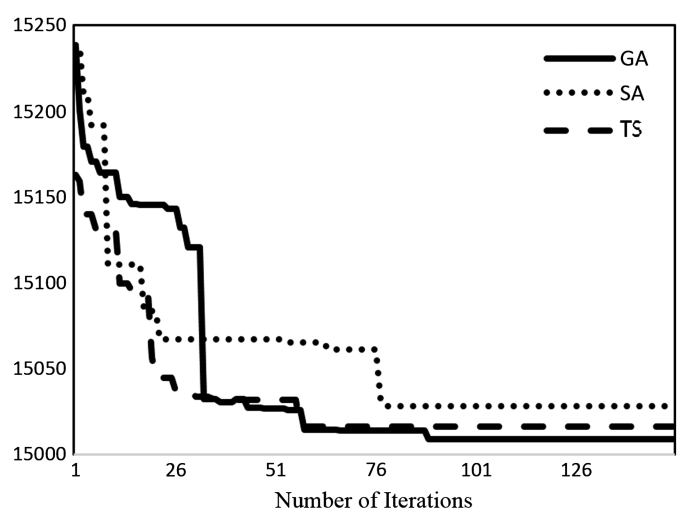

Fig. 2. Performances of GA, SA, and TS for 150 iterations

Table 3. Selection, Sequencing, and Scheduling of Projects

Bottleneck order	GA		SA		TS	
	Sequence	Scheduled completion year	Sequence	Scheduled completion year	Sequence	Scheduled completion year
11	11	1.8	11	1.8	11	1.8
36	3	6	36	6	3	6
34	36	9	12	9	36	9
14	12	13	9	9.8	12	13
9	9	15.2	25	12	9	15.2
27	15	17.4	9	14	22	17.4
35	14	18.2	37	17.6	14	18.2
12	21	20	4	19.8	8	21.2
15	13	21.6	8	22.8	27	22.4
21	2	26.2	14	23.6	15	24.6
3	22	28.4	21	25.4	4	28.2
13	—	—	27	27.6	37	29.1
30	—	—	—	—	—	—
NPV = 15,605	NPV = 15,009		NPV = 15,028		NPV = 15,016	

Note: NPV = net present value of total cost (million $).

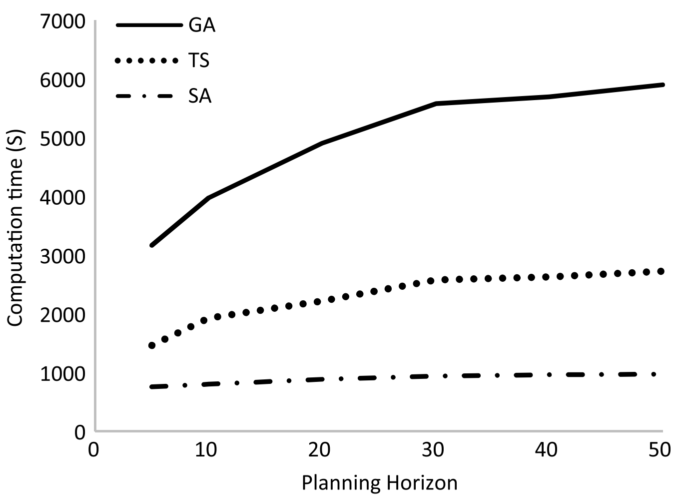

Fig. 3. Sensitivity analysis for GA, SA, and TS (computation time versus planning horizon)

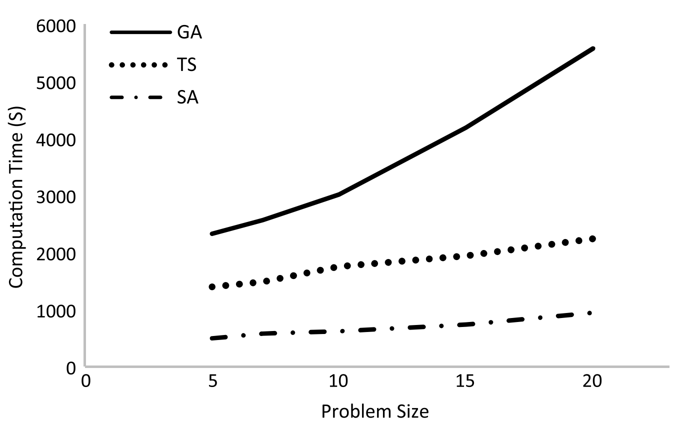

Fig. 4. Sensitivity analysis for GA, SA, and TS (computation time versus problem size)

results indicate that the computation time is more sensitive to the problem size for the GA than for SA and TS.

Consistency

When running replications of the metaheuristics, one can find how similar the results are among replications after different numbers of iterations. In other words, how consistent are the outcomes after running a specific number of iterations and at what point do they reach steady state? To address these questions, after running 50 replications the coefficient of variation (CV) of the objective function is estimated for each number of iterations. Fig. 5 shows the CV value for each algorithm as the number of iterations increases. It indicates that the variation in results is relatively low initially since the set of initial solutions is quite similar, then it increases during the process, and finally drops after the 80th iteration, converging to 0.07% for the GA. This means that running different replications of the GA method yields almost similar results after the 80th iteration. Similarly for TS and SA, the CV value fluctuates with the number of iterations and finally converges to 0.13% and 0.22%, respectively. In the case analyzed here, the GA is the most consistent algorithm, followed by TS and SA.

Conclusions

The selection and scheduling of interrelated projects is an interesting problem for policymakers and researchers in various fields, including economics, operation research, business, management, and transportation. Although it is crucial to consider the interrelations among projects when evaluating and prioritizing them, the problem is not sufficiently addressed in the literature. This paper combines a simple traffic assignment model for evaluating the objective function with three metaheuristic algorithms, namely, GA, SA, and TS, for optimizing the sequence and schedule of interrelated expansion projects. In particular, the optimized schedule is directly determined by the sequence of selected projects. More specifically, under a limited budget, which is continuously distributed over time, it is reasonable to fund and finish each project one at a time and gain benefits from each project as soon as it is completed.

To apply the proposed algorithms and demonstrate the numerical results, a sample network is examined through the evaluation

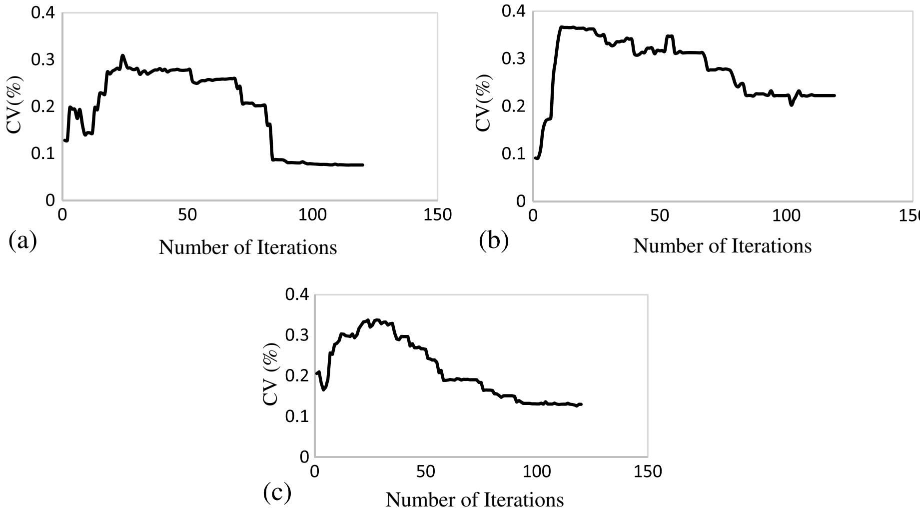

Fig. 5. (a) Coefficient of variation of objective function for GA; (b) coefficient of variation of objective function for SA; (c) coefficient of variation of objective function for TS

and optimization process. The outcomes are further used for a multilateral comparison in the last section of the paper. After finding the optimum sequence and schedule of the projects, a comparative analysis indicates that the GA, SA, and TS decrease the present worth of the total cost by \$596 million, \$577 million, and \$589 million, respectively, compared to a congestion-ranked solution, indicating that the GA yields a better solution with lower total costs than the other two. However, the SA and TS yield better solutions in the earlier stages of the search and thus seem preferable if computation budgets are limited, although the latter case is unlikely in the long-term planning and scheduling of significant investments. The results also indicate that the GA yields the most consistent solutions with a 0.07% coefficient of variation for the 150th iteration, implying that different replications of the GA yield almost equal final solutions after a sufficient number of iterations.

Acknowledgments

This research was partly funded by the Maryland State Highway Administration and the USDOT University Transportation Center Program through the National Center for Strategic Transportation Policies, Investments, and Decisions at the University of Maryland. The opinions in this paper do not necessarily reflect the views of the funding agencies. The authors are solely responsible for all statements in the paper.

Supplemental Data

The TSC Report 2015-04, "Characteristics of the Sioux Falls Test Network," is available online in the ASCE Library (www.ascelibrary.org).

References

Ahipasaoglu, D. S., Sun, P., and Todd, M. J. (2008). "Linear convergence of a modified Frank-Wolfe algorithm for computing minimum-volume enclosing ellipsoids." *Optim. Method. Software*, 23(1), 5–19.

Bagloee, S. A., and Tavana, M. (2012). "An efficient hybrid heuristic method for prioritising large transportation projects with interdependent activities." *Int. J. Logist. Syst. Manage.*, 11(1), 114–142.

Ben-Ameur, W. (2004). "Computing the initial temperature of simulated annealing." *Comput. Optim. Appl.*, 29(3), 369–385.

Bhattacharyya, R., Kumar, P., and Kar, S. (2011). "Fuzzy R&D portfolio selection of interdependent projects." *Comput. Math. Appl.*, 62(10), 3857–3870.

Bouleimen, K., and Lecocq, H. (2003). "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version." *Eur. J. Oper. Res.*, 149(2), 268–281.

Černý, V. (1985). "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm." *J. Optim. Theory Appl.*, 45(1), 41–51.

Chen, X., Zhu, Z., He, X., and Zhang, L. (2015). "Surrogate-based optimization for solving mixed integer network design problem." *Transp. Res. Rec*, 15-4556, in press.

Devika, K., Jafarian, A., and Nourbakhsh, V. (2014). "Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques." *Eur. J. Oper. Res.*, 235(3), 594–615.

Disatnik, D. J., and Benninga, S. (2007). "Shrinking the covariance matrix." *J. Portfolio Manage.*, 33(4), 55–63.

Dueñas-Osorio, L., Craig, J. I., Goodno, B. J., and Bostrom, A. (2007). "Interdependent response of networked systems." *J. Infrastruct. Syst.*, 10.1061/(ASCE)1076-0342(2007)13:3(185), 185–194.

Durango-Cohen, P. L., and Sarutipand, P. (2007). "Capturing interdependencies and heterogeneity in the management of multifacility transportation infrastructure systems." *J. Infrastruct. Syst.*, 10.1061/(ASCE)1076-0342(2007)13:2(115), 115–123.

Frank, M., and Wolfe, P. (1956). "An algorithm for quadratic programming." *Nav. Res. Logist. Q.*, 3(1–2), 95–110.

Gen, M., and Cheng, R. (1997). *Genetic algorithms and engineering design*, Wiley, New York.

Glover, F. (1986). "Future paths for integer programming and links to artificial intelligence." *Comput. Oper. Res.*, 13(5), 533–549.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, New York.

Jeon, K., Lee, J., Ukkusuri, S., and Waller, S. (2006). "Selectorecombinative genetic algorithm to relax computational complexity of discrete network design problem." *Transp. Res. Rec.*, 1964, 91–103.

Jong, J. C., and Schonfeld, P. (2001). "Genetic algorithm for selecting and scheduling interdependent projects." *J. Waterw. Port, Coastal, Ocean Eng.*, 10.1061/(ASCE)0733-950X(2001)127:1(45), 45–52.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). "Optimization by simulated annealing." *Science*, 220(4598), 671–680.

LeBlanc, L. J., Morlok, E. K., and Pierskalla, W. P. (1975). “An efficient approach to solving the road network equilibrium traffic assignment problem.” *Transport. Res.*, 9(5), 309–318.

Li, Z., Roshandeh, A. M., Zhou, B., and Lee, S. H. (2013). “Optimal decision making of interdependent tollway capital investments incorporating risk and uncertainty.” *J. Transp. Eng.*, 10.1061/(ASCE)TE.1943-5436.0000540, 686–696.

Markowitz, H. (1952). “Portfolio selection.” *J. Financ.*, 7(1), 77–79.

Michalewicz, Z. (1995). *Genetic algorithms + data Structure = evolution programs*, Springer, Berlin.

Mika, M., Waligóra, G., and Węglarz, J. (2005). “Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models.” *Eur. J. Oper. Res.*, 164(3), 639–668.

Milatovic, M., and Badiru, A. B. (2004). “Applied mathematics modeling of intelligent mapping and scheduling of interdependent and multi-functional project resources.” *Appl. Math. Comput.*, 149(3), 703–721.

Shayanfar, E., and Schonfeld, P. (2015). “Characteristics of the sioux falls test network.” *TSC Rep. 2015-04*, Univ. of Maryland, College Park, MD.

Sicilia, J. A., Quemada, C., Royo, B., and Escuín, D. (2015). “An optimization algorithm for solving the rich vehicle routing problem based on variable neighborhood search and Tabu search metaheuristics.” *J. Comput. Appl. Math*, in press.

Szimba, E., and Rothengatter, W. (2012). “Spending scarce funds more efficiently-including the pattern of interdependence in cost-benefit analysis.” *J. Infrastruct. Syst.*, 10.1061/(ASCE)IS.1943-555X.0000102, 242–251.

Tao, X., and Schonfeld, P. (2005). “Lagrangian relaxation heuristic for selecting interdependent transportation projects under cost uncertainty.” *Transp. Res. Rec.*, 1931(1), 74–80.

Tao, X., and Schonfeld, P. (2007). “Island models for a stochastic problem of transportation project selection and scheduling.” *Transp. Res. Rec.*, 2039, 16–23.

Van Vliet, D. (1987). “The Frank-Wolfe algorithm for equilibrium traffic assignment viewed as a variational inequality.” *Trans. Res. B-Method*, 21(1), 87–89.

Wang, S. L. (2001). “Simulation and optimization of interdependent waterway improvement projects.” Ph.D. dissertation, Univ. of Maryland, College Park, MD.

Wang, S. L., and Schonfeld, P. (2005). “Scheduling interdependent waterway projects through simulation and genetic optimization.” *J. Waterw. Port, Coastal, Ocean Eng.*, 10.1061/(ASCE)0733-950X(2005)131:3(89), 89–97.

Wang, S. L., and Schonfeld, P. (2008). “Scheduling of waterway projects with complex interrelations.” *Transp. Res. Rec.*, 2062, 59–65.

Xu, T., Wei, H., and Hu, G. (2009). “Study on continuous network design problem using simulated annealing and genetic algorithm.” *Expert. Syst. Appl.*, 36(2), 1322–1328.