ABSTRACT

Title of Document: PRIORITIZING AND SCHEDULING INTERRELATED ROAD PROJECTS USING METAHEURISTIC ALGORITHMS

Elham Shayanfar, Masters of Science, 2015

Directed By: Professor Paul Schonfeld
Department of Civil and Environmental Engineering

Projects are considered interrelated when their benefits or costs depend on which other projects are implemented. Selection and scheduling of interrelated projects is a challenging optimization problem which has 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 subject to a budget constraint. The budget is supplied by both external and internal sources from fuel tax revenues. This study then applies three meta-heuristic 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 in terms of solution quality, computation time and consistency.

PRIORITIZING AND SCHEDULING INTERRELATED ROAD PROJECTS USING METAHEURISTIC ALGORITHMS

By

Elham Shayanfar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Masters of Science
2015

Advisory Committee:
Professor Paul Schonfeld, Chair/Advisor
Associate Professor Lei Zhang
Associate Professor Cinzia Cirillo

Dedication

This work is dedicated to my parents for their endless love, support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Problem Statement

Evaluating transportation infrastructure projects and determining which of them at what time should be implemented requires several criteria. Common evaluation practices imply the linear aggregation of project impacts in the objective function which is later optimized. Nevertheless, these assumptions are inadequate since they disregard the interdependence due to non-linearly additive benefits, costs, budget constraints, constructability or operability requirements, and other possible factors. The selection and scheduling of projects with consideration of their interrelations is a challenging optimization problem, but its solution is very valuable as it has applications in various fields, including economics, finance, operations research, development, industrial engineering, and business administration. This research deals with road expansion projects as an example of interrelated projects, however, the introduced methods may be used generally to analyze interrelated alternatives.

As traffic increases and links become congested, passenger and freight movements experience increasing travel times and delays. One obvious solution to this problem is constructing new lanes and creating additional capacity on the highly congested links. Then we must determine which links should be selected, in what order they should be implemented and when they should be funded in order to minimize the present worth of cost. 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 as 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 we should consider their interrelations.

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. A second step selects which projects from among the considered set should be chosen for implementation. After evaluating and selecting a set of projects for improvement, a third step determines the order of projects and, finally, a fourth step determines the scheduled time for completion under budget limitations (Wang and Schonfeld 2005). Project selection and scheduling easily becomes a large optimization problem whose feasible region increases rapidly as the number of considered projects in the system grows. Considering a set of improvement projects for a given network, the objective is to find a project implementation sequence that minimizes the total system cost 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.

1.2 Sioux Falls Network

The Sioux Falls network is considered as a case study for this problem. Sioux Falls is the largest city in the U.S. state of South Dakota. It is the county seat of Minnehaha County, and also extends into Lincoln County to the south. Figure 0-1 shows the real map to the city. However, the network used in this research is partially different from the real network which has been used in many publications. This network is good for code debugging and also provides an opportunity to examine the data format. More detail about this network is presented in section 3.1.

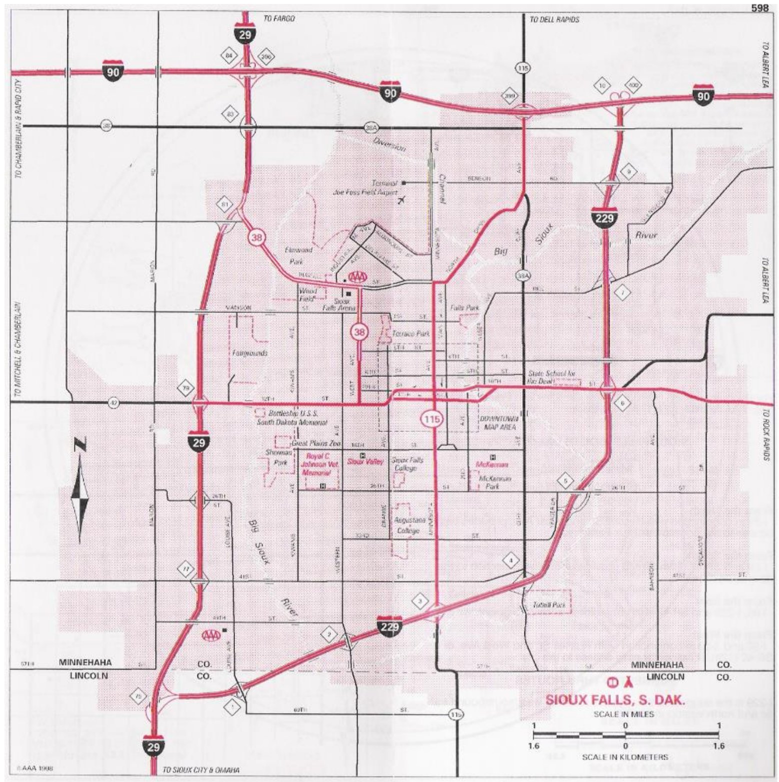


Figure 0-1 Sioux Falls Network Map

1.3 Research Objective and Contribution

One of the objectives of this research is to compare three alternative meta-heuristic 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). The GA results are furthermore investigated by sensitivity analysis and a statistical approach for optimality check. 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. In recent years, the meta-heuristics compared here have been widely used for finding optimal or near-optimal solutions. The work presented in this thesis contributes to the current research in several ways. First, we apply three meta-heuristics to explore and compare different approaches for solving the selection and scheduling problem. Second, we further modify algorithms' assumptions to account for the possibility that candidate projects may become economically unjustified after the implementation of previous projects, This may occur due to project interrelations and the possibility that the cost savings from completing a project are affected by earlier project implementations. Moreover, a multi-period analysis is incorporated in this study to distinguish between peak and off peak hour demands. Finally, we demonstrate these algorithms by conducting a case study and compare 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 work should also be applicable to other prioritization problems with interrelated alternatives.

1.4 Thesis Organization

Chapter 2 of the thesis overviews the existing literature on evaluating and prioritizing interrelated projects mostly focusing on recent works. Chapter 3 explains the methodology used for both evaluation and optimization process as well as the important assumptions made in this research. Chapter 4 provides the detailed algorithms for the three metaheuristic models (Simulated Annealing, Tabu Search and Genetic Algorithm) is provided. Chapter 5 describes the case study characteristics and network configuration. Chapter 6 presents the outcomes from the metaheuristic algorithms and compares their functionality from different aspects. Chapter 7 includes further analysis including sensitivity analysis and a statistical approach to check the goodness of the optimal solution exclusively for the GA results. Finally chapter 8 summarizes the methodology and results of the selecting and scheduling problem and provides future steps and recommendations.

Chapter 2: Literature Review

Studies in the existing literature mainly deal with the selection and scheduling of projects by assuming independence among them. Two approaches are commonly used for selecting and sequencing of independent projects. These are integer programming (Weingartner 1966; Cochran et al. 1971; Clark et al. 1984; Johnson et al. 1985) and dynamic programming (Weingartner 1966; Nemhauser and Ullman 1969; Morin and Esogbue 1971; Erlenkotter 1973; Morin 1974). The drawbacks of these approaches include the difficulty of capturing the interrelations among projects and their inefficiency or even the infeasibility for large problems (Jong and Schonfeld 2001).

In portfolio management, interrelations between choices (stocks) were identified and modelled as early as the 1950s in pioneering work by Markowitz (1952). The linear program is extended into a quadratic program with the inclusion of variances of returns for different stocks. The objective is to minimize the sum of purchase cost and interrelated risks. The consideration of project interdependence significantly complicates the model's structure because the combined costs and benefits for a set of projects are no longer equal to the sum of the costs and benefits, respectively, of individual projects. The dependence matrix, in the portfolio optimization, is convenient in modeling interdependence between choices. Its variants are still used in recent works, e.g., Durango-Cohen and Sarutipand (2007) and Bhattacharyya et al. (2011). However, the estimation of dependence matrix is difficult and its manipulation is computationally burdensome when the project space grows (Disatnik and Benninga 2007). Moreover, the pairwise dependency between the projects as well as three-way and higher order dependencies, are insufficient to model the complex relations among infrastructure development projects, as well as cumbersome to estimate. For example, roadway improvement projects are usually interrelated since delays at one link are affected by operations at other links, both upstream and downstream. Expansion at one location may shift delays and capacity bottlenecks elsewhere. Some complete system models, such as queueing approximations (Jong and Schonfeld 2001), equilibrium assignment (Tao and Schonfeld 2005), microsimulation (Wang et al. 2009) and neural networks (Bagloee and Tavana 2012), have been adopted to model the interrelations. The following section reviews some relevant literature on evaluating and prioritizing interdependent projects.

Solving the interrelation problem is first attempted by Weingartner (1966) with an integer programming approach. Nemhauser and Ulmann (1969) tried to re-solve Weingartner's objective function by incorporating pairwise interactions. Afterwards, several studies, namely Gear (1980), Fox (1984), and Janson (1988) aimed to broaden the second-order interaction to third-order and fourth-order but failed to reflect all possible interactions. Martinelli (1993) proposed a heuristic method for selecting and scheduling interdependent waterway investment projects by comparing various combinations of projects over an analysis period. This method began by establishing an initial sequence based on an independent evaluation. Then the initial solution is adjusted with a heuristic performing pairwise swaps as long as system costs were improved, according to evaluation functions estimated from simulation results. Wei and Schonfeld (1993) developed an algorithm which combined an artificial neural network and a branch and bound algorithm to find a a near-optimal solution for scheduling interdependent projects. They proposed a multi-period network design model for selecting the best combination of improvement projects and schedules. Then they utilized the neural network approach to estimate total travel times corresponding to different project selection and scheduling. They applied their model to the Calvert County highway system in southern Maryland to check its performance. Martinelli and Schonfeld (1995) developed an approximation to microsimulation model to evaluate lock improvements with consideration of their interrelation. Jong and Schonfeld (2001) developed a genetic algorithm and a simple approximation to solve project investment planning problem. They showed that GAs are very effective at searching for minimum cost highway alignments.

Bouleiman and Lecocq (2002) developed a simulated annealing algorithm for the resource constrained project scheduling problem. The objective of this model is to minimize total project duration. A new design is substituted the conventional SA search scheme which considered the specificity of the solution space of project scheduling problems. Tao and Schonfeld (2005) developed a Lagrangian heuristic to solve the selection of interdependent projects under cost uncertainty. In this paper a genetic algorithm is developed to solve the Lagrangian problem, and an equilibrium assignment is applied to evaluate the objective function. Mika et.al (2005) proposed two local search meta-heuristics, simulated annealing and tabu search to solve the multi-mode resource constrained project scheduling problem with discounted cash flows. The objective is set to maximize the net present value of all cash flows. Four payment models were considered in this study: lump-sum payment at the completion of the project, payments at activity completion times, payments at equal time intervals, and progress payments. They evaluated their model on a set of instance switches that were based on some standard test problems constructed by the ProGen project generator. 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 genetic algorithm. Milatovic and Badiru (2004) proposed a methodology for mapping and scheduling of interdependent and multifunctional project resources. Their methodology performed two alternative procedures, namely activity scheduler and resource mapper. The first procedure prioritized and scheduled activities based on their attributes and the latter considered resource characteristics and mapped the available resource units to the scheduled activities.

Durango-Cohen and Sarutipand (2007) developed a quadratic programming formulation for optimizing maintenance and repair (M&R) policies with consideration of link interdependencies in a network. The quadratic objective of their work captures the pairwise economic dependencies which reflect both the costs and benefits of improving adjacent facilities. Tao and Schonfeld (2007) developed variation of traditional genetic algorithms called island models 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 larger when interdependencies are considered after disturbances. Zhang et.al (2008) proposed an agent-based approach to evaluate price completion, capacity choice, and product differentiation on congested networks. This approach is specifically designed to analyze the distributional effects of network management and financing policies, and evaluate policy scenarios with consideration of the spatial and temporal effects of these policies. Zhang and Levinson (2008) study the effect of investment rules on hierarchical structure of roads, and their vulnerability to natural disasters, congestion and accidents. In this research a set of Monte Carlo simulation runs are used to evaluate the equilibrium road network under two policy scenarios: 1. Investment based on benefit-cost ratio; 2.investments based on bottle-neck removal. Zhang and Levinson (2009) explore the economic impact of alternative ownership structures on transportation system performance, social welfare, and regulatory needs. An agent-based evolutionary model is used to capture Road pricing, investment, and ownership decisions in a large network. Results suggest that a completely privatized transportation network could achieve net social benefits close to the theoretical optimum.

Szimba and Rothengatter (2012) extended the classical benefit-cost analysis by integrating the occurrence of interdependence among the projects within an investment package. They addressed the interdependence problem by introducing a heuristic method to solve the large-scale problem with numerous projects. In this approach, the number of projects and their interrelations are reduced step by step in order to reduce the number of interdependence cases. Bagloee and Tavana (2012) formulated the prioritization problem as a Traveling Salesman Problem, and incorporated a Neural Network to deal with project interdependence. A heuristic algorithm with hybrid components is then used to search for the longest path (most benefit) in the NN as a solution to the TSP. Zhang and Yusufzyanova (2012) evaluate the pricing and capacity decisions for private toll roads against existing public roads using an agent-based model. The purpose of this study is to evaluate regulation policy packages and illustrate the effect of public and private network hierarchy on network growth patterns. Li et.al (2013) proposed a hypergraph knapsack model to maximize the overall benefits for a sub collection of interdependent projects. For this purpose, a multi-commodity minimum cost network (MMCN) was developed to obtain traffic volume and speed to estimate benefits using a life cycle cost analysis method. Mollanejad et.al (2014) proposed a model to prioritize transportation investments for a megaregion. They maximized the total production of the megaregion subject to a budget and environmental constraints. In this study a simple all-or-nothing traffic assignment model used to evaluate transportation investments. Another study by Mollanejad and Zhang (2014) attempts to prioritize road improvements by accounting for equity issues into the interurban road network design problem. This is done by minimizing the total inaccessibility in the region by solving a mixed integer program. Chen et.al (2015) reformulates the mixed network design problem (MNDP) to simultaneously find both optimal capacity expansions of existing links and new link additions. The upper level aims to minimize the network cost in terms of the average travel time via the expansion of existing links and the addition of new candidate links. The lower level is a dynamic user-optimal condition that can be formulated as a variational inequality problem. A surrogate based optimization framework is proposed to solve the MNDP.

The literature review indicates that most application methods for selection and prioritization of projects as part of investment packages are based on classical BCA

(Benefit-Cost Analysis) in which projects are regarded as mutually exclusive components. On the other hand, the state-of-the art shows both insufficient studies on the matter and lack of comprehensive applicable methods for real world problems. The literature also indicates that earlier studies addressed the problem by customizing it to specific cases, without generalizing it to real world problems. Many studies address the problem of interrelation by estimating and using the marginal pairwise or n-way interrelations which are rarely adequate. More complete system models, for example simulation models, are desirable for evaluating systems with project interrelations. Furthermore, existing methods are computationally expensive and may be overwhelmed when the numbers of scenarios are increased.

This study solves the project selection and scheduling problem where the objective function is not implicit but can be evaluated with a user equilibrium model. The study also takes into account several uncertainties by conducting a sensitivity analysis of important parameters. The most important components of the proposed methodology that extend the current methodologies offered in the literature include the following: First, the budget constraint is reformulated to accommodate internal budget supply by assessing fuel tax revenues before each project implementation. Second, three meta-heuristic algorithms are used to solve the optimization problem whose objective function is evaluated with user equilibrium model that accounts for all possible interactions beyond the conventional pairwise interaction. Third, these algorithms are modified to account for the possibility that candidate projects may become economically unjustified after the implementation of previous projects. This task is fulfilled by calculating the marginal benefits and costs of candidate projects before including them in the project sequence. Finally, a roadway network example is designed to test the performance of the proposed methodology.

Chapter 3: Research Methodology

3.1 Development of Evaluation Model

This research incorporates the *convex combination algorithm* (Frank and Wolfe 1956) to evaluate link expansion projects upon their implementation in the network. This method is an iterative algorithm applicable for nonlinear programming problems with convex objective functions and linear constraints. 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.

Given a current travel time for link a, t_a^{n-1} the *n*th iteration of the convex combination algorithm is summarized as follows:

1. *Initialization*: all or nothing assignment assuming t_a^{n-1} which yields x_a^n.
2. *Updating travel time*: using a BPR function $t_a^n = t_a(x_a^n) = t_0(1 + 0.15\,(\frac{v}{c})^4)$.
3. *Direction finding*: all or nothing assignment considering t_a^n which yields auxiliary flow y_a^n.
4. *Line search*: find α that solves $min \sum_a \int_0^{x_a^n + \alpha(y_a^n - x_a^n)} t_a(\omega)d\omega$.
5. *Move:* set $x_a^{n+1} = x_a^n + \alpha_n(y_a^n$- $x_a^n)$, $\forall\alpha$.
6. *Convergence test*: If a convergence criterion met, stop. Otherwise set n=n+1 and go to step 1.

3.2 Problem Formulation

The objective function for prioritizing transportation investments has a non-convex 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:

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

As a result, heuristic methods have gained popularity among researchers for solving such complex problems. This research explores three meta-heuristic methods, including Genetic Algorithm, Simulated Annealing and Tabu Search, which have often been found effective in finding near-optimal solutions. 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 *(i)* total road user cost and *(ii)* total supplier cost, subject to a budget constraint. In this problem, an internal source of budget is considered for funding the future projects. More specifically, the collected fuel taxes from users is added to an external budget, constituting the overall budget for future projects. This assumption concurs with the reality as fuel taxes and toll collections contribute substantially to the highway improvement budget. The following equation is used to estimate the internal budget:

$$Internal\ Budget = VMT * fuel\ consumption * fuel\ cost * tax \quad (2)$$

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:

$$\sum_{i=1}^{n_p} c_i x_i(t) \leq \int_0^t b(t)dt, \quad 0 \leq t \leq T \qquad (3)$$

$$\begin{cases} x_i(t) = 0 & if\ t < t_i \\ x_i(t) = 1 & if\ t > t_i \end{cases}$$

where t_i is the time when project *i* is finished and $x_i(t)$ is a binary variable specifying whether project *i* is finished by time *t*. It should be noted that the set of all t_is eventually determines the schedule of projects. This occurs because under the 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 immediate benefit as soon as a project is completed. In other words, funding multiple projects simultaneously increases the completion time meaning that the cost savings of capacity improvements are delayed. 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 by considering the budget flow. The idea is that each project is funded immediately after the predecessor one is finished and is completed as soon as the available cumulative budget reaches the project cost. To date, other studies assumed that all candidate projects remain justified until the end of the studied period. However, due to project interdependence the cost savings of completing a project may change over time. In order to tackle this problem, we developed algorithms that account for the possibility that projects may become economically unjustified after some other projects are implemented.

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 by value of time, and the supplier cost is describes as the present value of all project costs. Unlike in some previous studies, the cost of projects has to be included in the objective function since not all the selected projects are guaranteed to be implemented during the analyzed period. In fact, some projects may be discarded from the sequence as they may not be financially justified at some point during the analysis. The present worth of total cost Z to be minimized is:

$$min\,Z = \sum_{j=1}^{T} \frac{v}{(1+r)^j} \sum_{i=1}^{n_l} w_{ij} + \sum_{j=1}^{T} \frac{1}{(1+r)^j} \sum_{i=1}^{n_p} c_i x_i(t) \qquad (4)$$

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

3.3 Development of Optimization Models

One of the goals of this research is to compare the performance of three meta-heuristic methods (GA, SA, TS) in solving the selection, sequencing and scheduling of interrelated projects. The common elements of the four approaches are as follows.

3.3.1 Solution representation

The solutions are represented by the sequence of projects in which projects are implemented. In this setting, each project has to occur after all its predecessors and after all its successors. Figure 3-1 represents an example of a feasible solution.

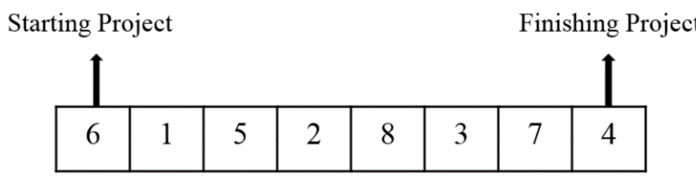


Figure 3-1 Example of a Feasible Solution

3.3.2 Objective function

The objective function with all three approaches minimizes the present worth of the total user and system cost subject to a budget constraint which was defined in the previous section.

3.3.3 Solution feasibility

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 not to exceed the planning horizon and the projects scheduled beyond the horizon are deleted from the accepted sequence. This makes intuitive sense as in real world application there are usually more desirable projects than the budget available during a planning time and one must choose 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 optimal timing of projects: Spending on the next project starts immediately after spending on the preceding project is completed.

3.3.4 Stopping criterion

Two stopping criteria, including number of iterations and running time, are tested for all the algorithms. In the first case, the search stops after a specific number of iterations is completed, and the second criterion terminates the search after a specified amount of computation time.

3.3.5 Project Selection, Sequencing and Scheduling

The framework of the general proposed method for selecting, sequencing and scheduling interdependent road projects is presented in Figure 3-2. The proposed combination of traffic assignment and metaheuristic algorithms may be used to evaluate any sequence of projects and discover the near optimal solution. Assuming that each project is funded immediately after the predecessor one is finished and is completed as soon as the available cumulative budget reaches the project cost, the schedule of projects is automatically derived from the sequence of projects. It should be noted that an internal source of budget is considered for funding the future projects. More specifically, the fuel tax revenues from users is added to an external budget, constituting the overall budget for future projects.

As a result, the schedule of each project is easily determined by considering the budget flow.

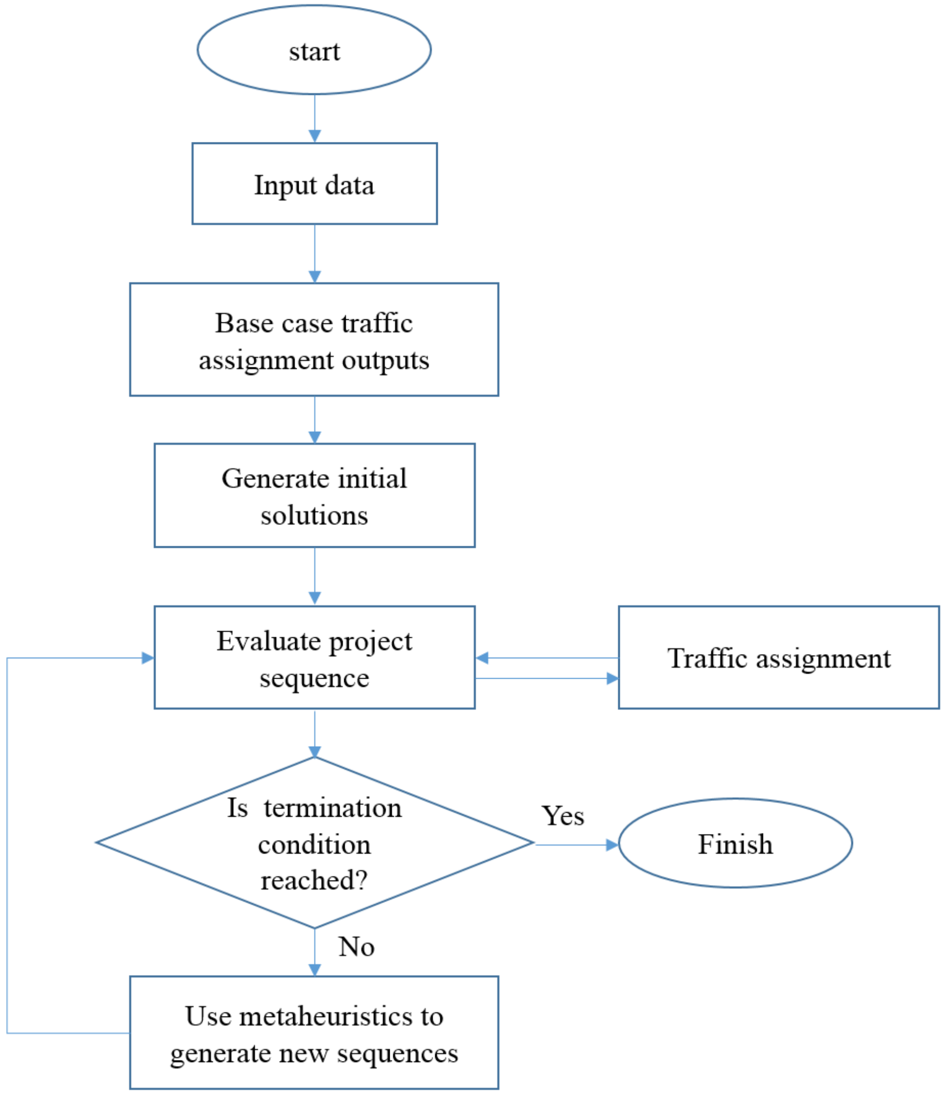


Figure 3-2 Overall Framework of Optimization Process

Chapter 4: Meta-Heuristic Algorithms

3.4 Genetic Algorithm

A Genetic Algorithm (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 get a set of possible solutions called the population. Each individual in the population is specified by a string of encoded genes which is called a chromosome. In this process some individuals are selected to reproduce off springs and since each individual has a probability of selection according to its fitness value, better ("fitter") solutions have a higher opportunity of being selected. Then the selected solutions are processed through a series of crossover and mutation operators which 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.

3.4.1 Initial population

In general, solutions of GAs are mostly represented by binary digits and the initial population is generated randomly. In this research, 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 comprising greedy-order solutions and bottleneck-order solutions are used to create the initial population (Jong 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 that link. This assumes that more congested links should have higher priority for being implemented.

3.4.2 Fitness function and parent selection

The fitness function is considered equivalent to the value of the objective function (NPV of total cost) and it 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, for preventing some undesirable properties of prematurity, a ranking method is applied instead (Wang 2001). In this method, the population is sorted with nonlinear ranking from the best to the worst. Then the selection probability of each chromosome is assigned according to its exponential ranking value considering the lowest fitness value equal to one (Michalewicz 1995). Let *q* be the selective pressure$\in [0,1]$, the selection probability is defined as follows:

$$P_i = c * q(1-q)^{i-1}, \quad c = 1/[1-(1-q)^{PopSize}] \tag{5}$$

Next, a roulette wheel approach is incorporated to select appropriate parents based on their selection probabilities (Michalewicz 1995). This process is conducted by spinning the roulette wheel *pop_size* times. Each time a random number *r [0,1]* is generated, then the i_{th} chromosome is selected such that $w_{i-1} < r \leq w_i$, where w_i is the cumulative probability for each chromosome.

3.4.3 Crossover and mutation

Then a crossover and a mutation operator are applied to reproduce offspring and create the 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 are described below adapted from Wang (2001):

Crossover operators:

1. Partial Mapped Crossover (PMX)

   Proposed by Goldberg and Lingle (1985), this two-point crossover exchanges the sequence of projects between two random positions in the selected parents. Then a mapping mechanism is established to correct for the possible duplication of projects by replacing the repeated projects by their corresponding projects. Figure 0-1 illustrates this process.

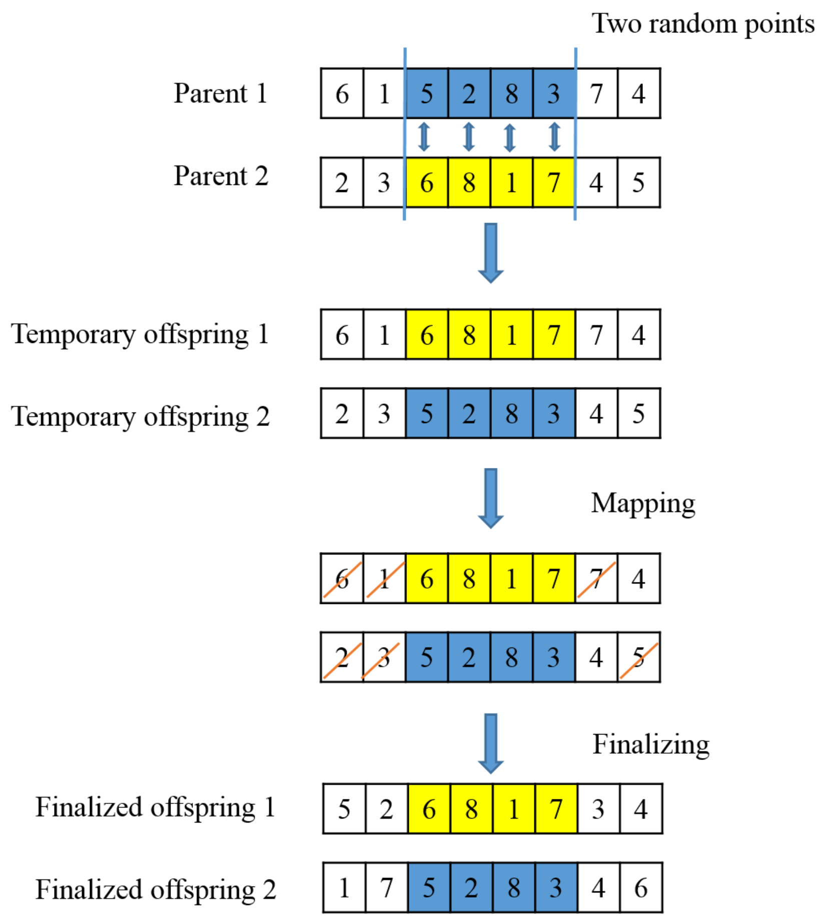


Figure 0-1 Example of Partial Mapped Crossover (PMX)

2. Position Based Crossover (PBX)

The PBX operator was proposed by Syswerda (1991). In this multi-point crossover, a set of random positions are selected from the first parent and copied to the same positions in the offspring. Then the projects that already exist in the offspring are deleted from the second parent and the rest are copied to the offspring with their original order. (Figure 0-2)

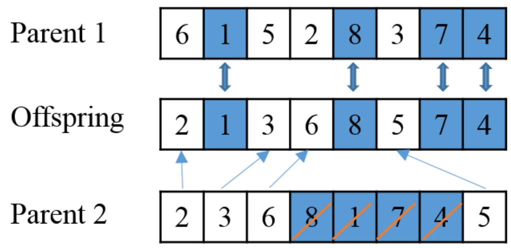


Figure 0-2 Example of Partial Based Crossover (PBX)

3. Order Crossover (OX)

   This two-point crossover operator was introduced by Davis (1985). This operator works by selecting two random points in the first parent and copying the sequence in between those points to the new offspring, keeping their original positions. The copied projects are deleted from the second parent and the remaining projects are inserted to the vacant positions in the offspring while keeping their order. (Figure 0-3)

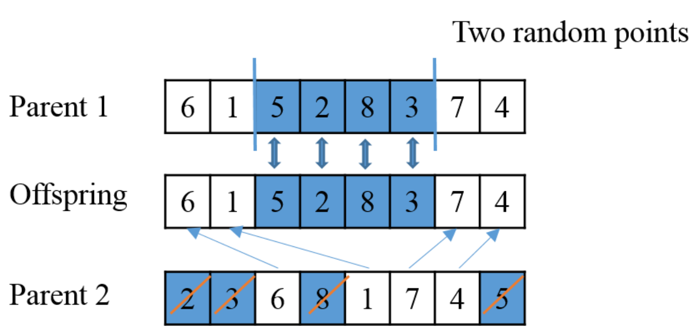


Figure 0-3 Example of Order Crossover (OX)

4. Order Based Crossover (OBX)

This operator also proposed by Syswerda (1991) is similar to PBX but imposes the selected positions in one parent on the corresponding projects in the second parent. (Figure 0-4)

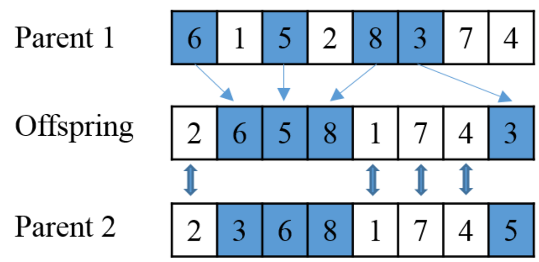


Figure 0-4 Example of Order Based Crossover (OBX)

5. Edge Recombination Crossover (ERX)

The edge recombination operator (ERX) is an operator that creates offspring exclusively by looking at the edges rather than the vertices. The idea here is to use as many existing edges, or node-connections, as possible to generate children. This operator is specifically useful when a genotype with non-repeating gene sequences is needed such as for the sequencing problem in this study. The method is introduced by Whitley et.al. (1989).

For each project *i*, the edge list consists of all other neighbor projects connected to project *i* from both parents. The construction of the offspring begins by selecting a project with the lowest number of edges. In case projects have equal number of edges, one of them is randomly chosen. The selected project is then crossed out from all the other edge lists, and the procedure continues by selecting the next project with the smallest number of edges until all projects are selected. Figure 0-5 shows an example of the ERX operator.

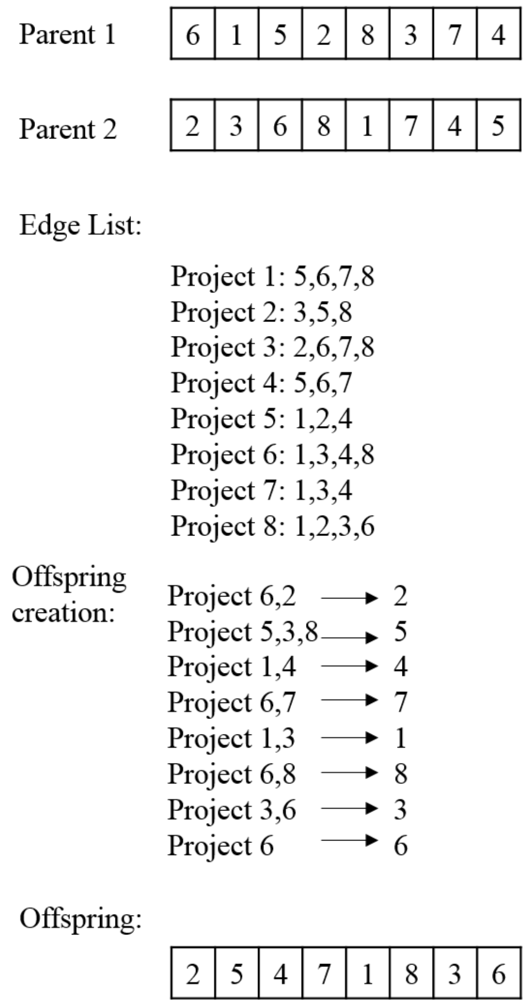


Figure 0-5 Example of Edge Recombination Crossover (ERX)

Mutation operators:

6. Insertion Mutation (IM)

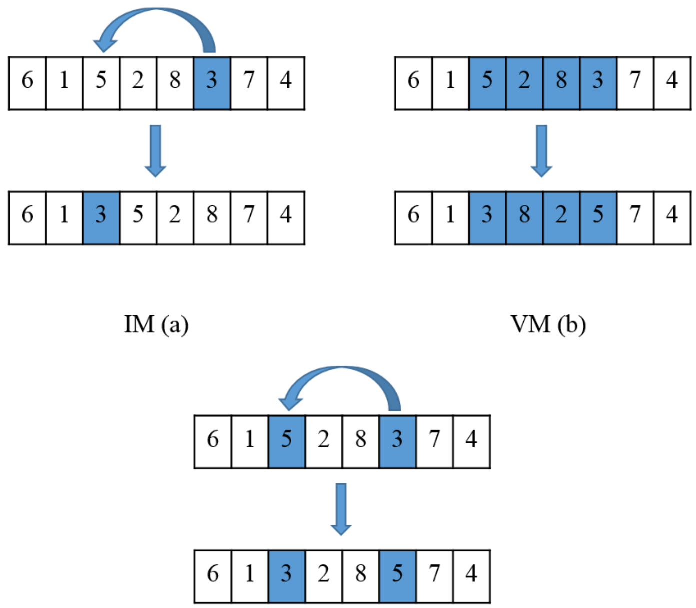
   In this operator a project is randomly selected and is inserted to a random position. Other projects are shifted over while keeping their original sequence. (Figure 0-6 a)

7. Inversion Mutation (VM)

This operator selects two random positions and inverts the subsequence between those two points. The other projects keep their positions. (Figure 0-6 b)

8. Reciprocal Exchange Mutation (EM)

The EM operator simply exchanges the position of two random projects while other projects keep their original order. (Figure 0-6 c)



IM (a) VM (b)

EM (c)

Figure 0-6 Example of Mutation Operators

The reproducing process randomly selects one operator and applies it on the selected parents.

3.5 Simulated Annealing

Simulated Annealing (SA) is a probabilistic meta-heuristic method for global optimization of an objective function which may possess several local optima. The algorithm which is introduced independently by Kirkpatrick et.al (1983) and Černý (1985) is inspired by a process which involves heating and gradual cooling of a material to reach minimum energy configuration. Starting from an initial solution *(S)*, the value of 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 the 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 $exp(-\Delta/T)$, where *T (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 *T* decreases within a *cooling function* $(T=T*\alpha)$ where α is a constant parameter by which the temperature decreases after each iteration. The algorithm finally stops when the stopping criterion is satisfied. This procedure is summarized as follows:

1. Generate an initial solution *S*.
2. Compute the initial temperature T_0 described in section 4.2.2.
3. Generate a neighborhood solution *S'*.
4. Let $\Delta=f(S')-f(S)$.
5. If $\Delta<0$, let *S*= *S'*

   Otherwise, let *S*= *S'* with probability $exp(-\Delta/T)$.

6. Stop if the stopping criterion is met

   Otherwise, let T=T*α and go to step 3.

3.5.1 Neighborhood

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. Figure 0-7 illustrates the neighbor generation process.

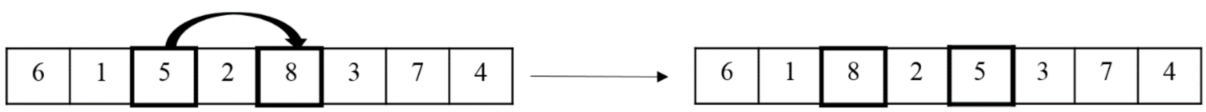


Figure 0-7 Neighbor Generation Example

3.5.2 Initial Temperature

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

$$T_{n+1} = T_n \left(\frac{\ln(\hat{\chi}(T_n))}{\ln(\chi_0)}\right)^{\frac{1}{\rho}} \tag{6}$$

where χ_0 is the desired acceptance probability, ρ is a real number ≥ 1 and $\hat{\chi}(T_n)$ 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:

$$\hat{\chi}(T_n) = \frac{\sum_{p\in P} \exp(-\frac{f(S')_p}{T})}{\sum_{p\in P} \exp(-\frac{f(S)_p}{T})} \tag{7}$$

The iteration stops as $\hat{\chi}(T_n)$ becomes sufficiently close to χ_0 and the value of T_n can be used as a good approximation for the initial temperature. The algorithm for this process is as following:

1. Estimate $||S||$, the number of samples needed to compute $\hat{\chi}(T)$.
2. Generate *S* random positive transactions.
3. Set T_1 at a strictly positive number.
4. Calculate $\hat{\chi}(T_n)$ from equation 6.
5. If $|\hat{\chi}(T_n) - \chi_0| < \varepsilon$, return T_n.

   Otherwise: - Compute T_{n+1} from equation 5,

- Go to step 4.

3.6 Tabu Search

Tabu Search (TS) is a meta-heuristic created 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 are explored in the search for the global optimal solution.

A move is defined as the position number in the project list selected for swapping. The following summarizes the procedure:

1. Generate a random solution *S.*
2. Generate a subset *N* of solution such that either one of them violates the tabu condition or the aspiration condition holds.
3. Let *S'* the best solution in subset *N.*
4. If $f(S')-f(S) < 0$, let $S=S'$.
5. Update tabu list.
6. Stop if the stopping criterion is met

   Otherwise, go to step 2.

3.6.1 Neighborhood

Similarly to simulated annealing, the neighbors of current solutions are generated by swapping the position of projects in the project sequence.

3.6.2 Move

According to the neighborhood generation described in the previous section, moves are defined by the position numbers of swapped projects. For example, if project 6 in position #3 is swapped with project 2 in position # 7, then the attributes of the move is (3,7).

3.6.3 Tabu list

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 reaches to a non-visited solution. In order to avoid the possibility of overlooking of 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.

Chapter 5: Case Study

3.7 Network Configuration

Figure 0-1 presents the network that is used in this work for testing user equilibrium and the metaheuristic algorithms. This network consists of 24 nodes and 76 links. Table 0-1 describes the hourly travel demand between each origin destination pair. These numbers are assumed as the peak hour demand and the off peak travels is considered half of these values. It is also assumed that the demand increases exponentially as a function of time over the planning horizon as follows:

$$d_{ij}^{t} = d_{ij}^{0} * (1 + r)^{t} \tag{1}$$

Where d_{ij}^{t} is the demand between origin i and destination$j$, d_{ij}^{0} is the base demand for the ij O/D pair at time 0, and r is the growth rate.

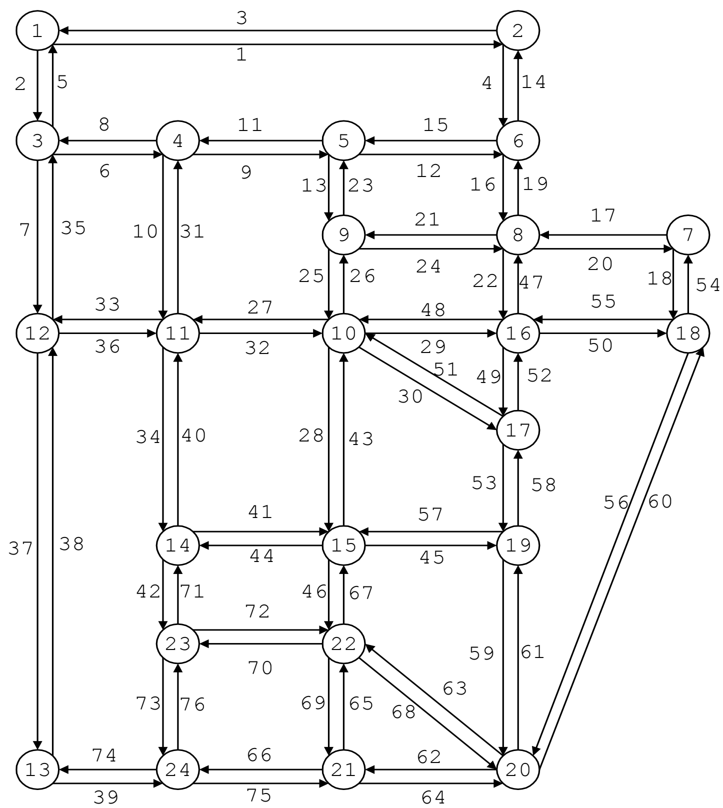


Figure 0-1 Sioux Falls Network

Table 0-1 Trip Table between Each Two Node Pairs (Vehicle per Hour)

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	***20***	***21***	***22***	***23***	***24***
1	0	20	12	36	18	24	30	48	36	84	36	18	36	18	30	36	30	12	18	18	6	24	18	12
2	20	0	6	18	6	30	12	30	18	36	12	12	18	6	12	24	18	6	6	12	6	12	6	6
3	12	6	0	18	6	18	6	12	12	18	18	18	12	6	6	12	6	0	6	6	6	6	6	6
4	36	18	18	0	30	30	30	42	48	72	90	42	36	30	30	48	30	6	18	24	12	24	30	18
5	18	6	6	30	0	18	12	36	48	60	36	12	12	12	18	36	18	6	12	12	6	12	12	6
6	24	30	18	30	18	0	24	48	24	48	24	18	18	12	18	60	36	6	18	24	6	18	12	6
7	30	12	6	30	12	24	0	66	36	114	30	48	30	18	30	84	60	60	30	36	18	36	12	6
8	48	30	12	42	36	48	66	0	0	96	54	36	36	24	42	132	84	18	42	54	24	36	24	12
9	36	18	12	48	48	24	36	0	0	168	90	42	36	36	60	90	60	12	30	42	24	42	36	12
10	84	36	18	72	60	48	114	96	168	0	240	126	114	132	240	264	0	42	108	156	78	162	108	54
11	36	12	18	90	36	24	30	54	90	240	0	90	60	96	90	84	60	12	30	42	30	66	84	36
12	18	12	18	42	12	18	48	36	42	126	90	0	84	42	48	42	42	12	18	30	24	48	42	30
13	36	18	12	36	12	18	30	36	36	114	60	84	0	36	42	42	36	6	24	42	36	78	48	48
14	18	6	6	30	12	12	18	24	36	132	96	42	36	0	45	42	42	6	24	30	24	72	66	24
15	30	12	6	30	18	18	30	42	60	240	90	48	42	45	0	78	90	18	48	66	48	156	60	30
16	36	24	12	48	36	60	84	132	90	264	84	42	42	42	78	0	168	100	84	102	36	72	36	18

17	30	18	6	30	18	36	60	84	60	0	60	42	36	42	90	168	0	42	102	102	42	102	36	18
18	12	6	0	6	6	6	60	18	12	42	12	12	6	6	18	100	42	0	24	100	6	24	6	6
19	18	6	6	18	12	18	30	42	30	108	30	18	24	24	48	84	102	24	0	78	30	78	24	12
20	18	12	6	24	12	24	36	54	42	156	42	30	42	30	66	102	102	100	78	0	78	0	42	30
21	6	6	6	12	6	6	18	24	24	78	30	24	36	24	48	36	42	6	30	78	0	114	42	36
22	24	12	6	24	12	18	36	36	42	162	66	48	78	72	156	72	102	24	78	0	114	0	0	72
23	18	6	6	30	12	12	12	24	36	108	84	42	48	66	60	36	36	6	24	42	42	0	0	48
24	12	6	6	18	6	6	6	12	12	54	36	30	48	24	30	18	18	6	12	30	36	72	48	0

The Sioux Falls network illustrated in Fig. 1 is selected for demonstrating the performance of the proposed algorithms. As mentioned earlier, this is not considered a realistic network since it mainly includes the city's major arterial roads and omits many characteristics of its transportation system. However, it has widely been used to examine and compare studies on networks (LeBlanc et.al 1975). After running the traffic assignment model, the critical lanes with high volume-capacity ratio were identified as an initial set of candidate projects. It should be noted that our model allows volume-capacity ratios above 1.0 since we are using a BPR function (Ref) for estimating link performances. Since our demand matrix is symmetrical for all origin and destination nodes, each link expansion improvement is assumed to be implemented in both directions between the two connected nodes, i.e. each project is defined as expanding two links between each pair of connected nodes. This assumption is also justified economically because it saves costs due to the parallel use of resources and construction equipment. After identifying an initial collection of candidates, all projects are further investigated through a benefit-cost analysis to identify and rank the economically beneficial projects. The finalized set of candidate projects includes links: {(2 & 5), (4 & 14), (6 & 8), (10 & 31), (13 & 23), (16 & 19), (22 & 47), (25 & 26), (27

& 32), (29 & 48), (33 & 36), (34 & 40), (39 & 74), (41 & 44), (49 & 52), (53 & 58), (65 & 69), (66 & 75), (70 & 72), (73 & 76) }. In order to find appropriate initial solutions, the traffic assignment model is conducted for all improvement scenarios. The first column in Table 0-2 Greedy Order and Bottle Neck Order Solutions shows the order of projects based on their benefit-cost ratio in a descending order. In this context, the benefit is the present worth value of travel time savings, and the cost is the present value of implementation cost. The third column displays the sequence of projects based on their congestion severity. More specifically, the links with lower level of service have higher priority.

Table 0-2 Greedy Order and Bottle Neck Order Solutions

Greedy Order Solution	Project Benefit (dollar)	Bottle-neck Order Solution	VC Ratio
16 & 19	217300346	**16 & 19**	2.17
39 & 74	193368891.2	**39 & 74**	1.89
4 & 14	189404178.1	**73 & 76**	1.79
33 & 36	161423612.9	**25 & 26**	1.62
13 & 23	117425401.3	**13 & 23**	1.59
29 & 48	91362676.79	**53 & 58**	1.48
2 & 5	87751582.78	**65 & 69**	1.42
2 & 14	74066280.27	**33 & 36**	1.41
49 & 52	71863521.6	**29 & 48**	1.36
34 & 40	70811859.82	**34 & 40**	1.35
6 & 8	69331975.33	**4 & 14**	1.35
53 & 58	68775533.16	**27 & 32**	1.32
66 & 75	61764580.07	**70 & 72**	1.31
22 & 47	61099053.87	**66 & 75**	1.22
41 & 44	60702083.07	**41 & 44**	1.21
27 & 32	60135953.13	**2 & 14**	1.12
25 & 26	59110008.45	**6 & 8**	1.11
65 & 69	44182898.22	**2 & 5**	1.11
70 & 72	36073907.44	**22 & 47**	1.09
73 & 76	5242573.323	**49 & 52**	1.04

3.8 Project Interdependence

Table 0-3. Travel Time Reduction due to Link Expansion shows a sample of the travel time savings from separate implementation of projects in the network. The second column presents the initial link travel times prior to project implementations while columns three to seven 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 interdependence. (Conceptually, if the capacity increases in one link the network, congestion and average travel times tend to increase in other links that are “in series” with it and decrease 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 after project implementation. Notably, the sum of all the cells in one column is not equal to the travel time changes on the links which are getting expanded. This, in effect, confirms the interrelation among links and the possible shifting of bottlenecks to surrounding links. Furthermore, the last column implies that the total system cost reduction from implementing two projects together is different from the sum of cost savings for the two individual projects, emphasizing that the cost saving of multiple projects is not a linear summation of their individual savings.

Table 0-3. Travel Time Reduction due to Link Expansion

link	Link travel time	link travel time reduction (min/veh)					
		expanding 2 & 5	expanding 4 & 14	expanding 6 & 8	expanding 10 & 31	expanding 13 & 23	expanding (2&5)&(4 &14)

	without projects						
1	3.594	0.006	-0.018	0.008	0.000	-0.004	-0.023
2	5.021	**1.115**	0.720	-0.408	-0.062	0.031	**1.659**
3	3.594	0.006	-0.018	0.008	0.000	-0.004	-0.023
4	10.356	2.338	**5.712**	3.468	0.658	-0.342	**7.240**
5	5.021	**1.115**	0.720	-0.408	-0.062	0.031	**1.659**
6	4.550	-0.346	0.570	**0.927**	-0.269	-0.144	0.736
7	2.618	0.016	0.041	0.032	0.021	0.041	0.052
8	4.550	-0.346	0.570	**0.927**	-0.269	-0.144	0.736
9	1.629	-0.093	0.166	-0.060	0.015	-0.206	0.164
10	7.374	-0.038	-4.221	-4.398	**2.803**	2.051	-2.308
11	1.629	-0.093	0.166	-0.060	0.015	-0.206	0.164
12	3.001	-0.344	0.340	-0.526	-0.272	-0.851	0.367
13	7.390	0.053	1.214	0.933	0.883	**2.392**	1.013
14	10.356	2.338	**5.712**	3.468	0.658	-0.342	**7.240**
15	3.001	-0.344	0.340	-0.526	-0.272	-0.851	0.367
16	7.882	-0.494	-1.261	-0.065	0.881	2.222	-2.483
17	1.796	0.013	0.089	0.028	0.015	0.018	0.087
18	1.312	0.000	0.001	0.001	0.000	0.000	0.001
19	7.882	-0.494	-1.261	-0.065	0.881	2.222	-2.483
20	1.796	0.013	0.089	0.028	0.015	0.018	0.087
21	7.333	0.691	1.464	1.392	0.647	0.267	1.475
22	4.369	0.151	0.788	0.999	0.500	0.378	0.537
23	7.390	0.053	1.214	0.933	0.883	**2.392**	1.013
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
75	3.287	0.090	0.189	0.164	-0.190	-0.173	0.139
76	1.431	-0.003	0.041	-0.087	0.050	0.033	0.079
Total travel time saving		9.753	15.656	8.054	13.632	18.037	25.216

It is assumed that each project improvement adds one lane equivalent to 700 vehicle/hour additional capacity to each link, and the equivalent annual cost of each lane expansion is assumed to be 4,000,000 $/lane-mile. The main cost saving of link expansion projects is the reduced travel time for all the users. These travel time reductions can be computed through the traffic assignment model by comparing the total system travel time before and after project implementation. Next we use the meta-heuristic methods described in previous sections to find near optimal solutions for the sequence and schedule of selected projects. When optimizing, it is desired to find a sequence of projects which can be implemented within the planning horizon (30 years). Therefore, each project with a scheduled completion time after the planning horizon is eliminated from the sequence. Additionally, the projects with unacceptable marginal benefit-cost ratio are discarded form the sequence list during the evaluation stages and replaced by other justifiable projects.

Chapter 6: Comparison of Optimization Methods

In this study network with 76 links, 20 improvement projects were selected based on the method explained in previous section. These candidate projects are evaluated by the user equilibrium traffic assignment model. Since the solution space for such problem is as large as 20!= 2.4329e+18, three metaheuristic algorithms are applied to search the project sequence. Table 0-1, Table 0-2 and Table 0-3 describe the characteristics and basic parameters of SA, GA and TS respectively.

Table 0-1 SA Parameters

Parameter	Value
Neighborhood Size	100
# of samples for initial temp	0.5
Cooling Ratio	0.8
Trial Count	20
Move method	Swap

Table 0-2 GA Parameters

Parameter	Value
Population Size	20
Mutation Rate	0.5
Crossover Rate	0.5
Selective Pressure	0.1
Sampling Mechanism	Roulette Wheel

Table 0-3 TS Parameters

Parameter	Value
Neighborhood Size	100
Tabu Tenure	3
Trial Count	20
Move method	Swap

This section analyzes the results obtained from GA, SA and TS in terms of *(i)* quality of the final results, *(ii)* computation speed and *(iii)* consistency of the optimized solutions. The study further compares each algorithm in the aforementioned categories.

3.9 Quality

Each meta-heuristic is tested for 50 replications, each encompassing 150 iterations, which is considered a reasonable number of iterations for comparison purposes since all three 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 Figure 0-1, presenting the performances of the GA, SA and TS.

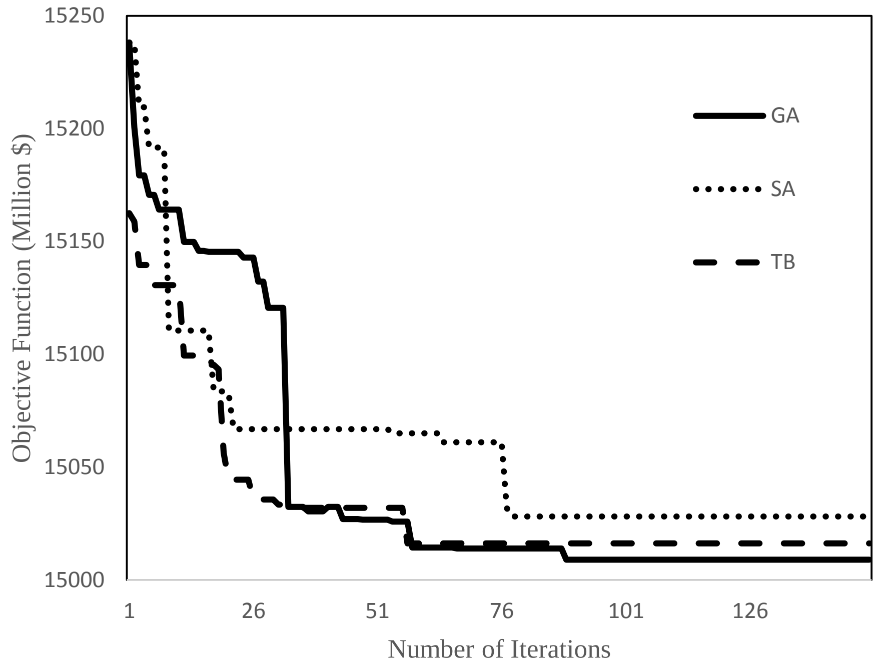


Figure 0-1 Performances of the GA, SA and TS for 150 Iterations

These results suggest that after running all the algorithms long enough for stable convergence, the GA performs better in terms of producing solutions with lower objective functions and TS performs better than SA. In this case, the present values of total system cost are: GA=15009, SA=15028, TS=15016 million dollars, which are remarkably close. Furthermore, the resulting selection, sequencing and scheduling of projects are presented in Table 0-4 which also demonstrates the comparison between the meta-heuristic solutions and the solution ranked according to congestion severities. The severity ranked solution has a total cost of $15605 million while the solutions obtained from the meta-heuristics have lower total costs, emphasizing the significance of project interdependencies. In fact, the present worth of total cost is reduced by 596, 577 and 589 million dollars when applying GA, SA and TS, respectively, compared to the severity-ranked order.

Table 0-4 Selection, Sequencing and Scheduling of Projects

	GA		SA		TS	
Bottle Neck Order	Sequence	Scheduled Completion Year	Sequence	Scheduled Completion Year	Sequence	Scheduled Completion Year
16 & 19	16 & 19	1.8	16 & 19	1.8	16 & 19	1.8
39 & 74	4 & 14	6	39 & 74	6	4 & 14	6
73 & 76	39 & 74	9	33 & 36	9	39 & 74	9
25 & 26	33 & 36	13	13 & 23	9.8	33 & 36	13
13 & 23	13 & 23	15.2	49 & 52	12	13 & 23	15.2
53 & 58	29 & 48	17.4	13 & 23	14	41 & 44	17.4
65 & 69	25 & 26	18.2	66 & 75	17.6	25 & 26	18.2
33 & 36	34 & 40	20	6 & 8	19.8	10 & 31	21.2
29 & 48	27 & 32	21.6	10 & 31	22.8	53 & 58	22.4
34 & 40	2 & 5	26.2	25 & 26	23.6	29 & 48	24.6
4 & 14	41 & 44	28.4	34 & 40	25.4	6 & 8	28.2
27 & 32	-		53 & 58	27.6	66 & 75	29.1
70 & 72	-		-			
15605	15009		15028		15016	
NPV of total cost (million $)						

3.10 Computation Time

The meta-heuristic results may also be compared in terms of computation time. For this purpose, the average running time per iteration is computed for all algorithms and is as following: GA= 87.5 sec, SA=19.3 sec and TS=37.7 sec. The results indicate that the GA has the most and the SA has the least computation time. This is due to 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 models to reach convergence, then GA yields better solutions than SA and TS.

3.11 Consistency

While running replications of the meta-heuristics we 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 they reach steady state? To address this question, after running 50 replications the coefficient of variation (CV) of the objective function is estimated for each number of iterations. Figure 0-2 shows the CV value for each algorithm as the number of iterations increases. It indicates that the variation of results is relatively low at the beginning since the set of initial solutions is quite similar, then it increases during the process, and finally drops after the 80^{th} iteration converging to 0.07%. This means that running different replications of the GA method yields almost similar results after the 80^{th} iteration. Similarly for TS and SA, the value of CV fluctuates along the number of iterations and finally converges to 0.13% and 0.22%. In this case, the GA is the most consistent algorithm, followed by TS and SA.

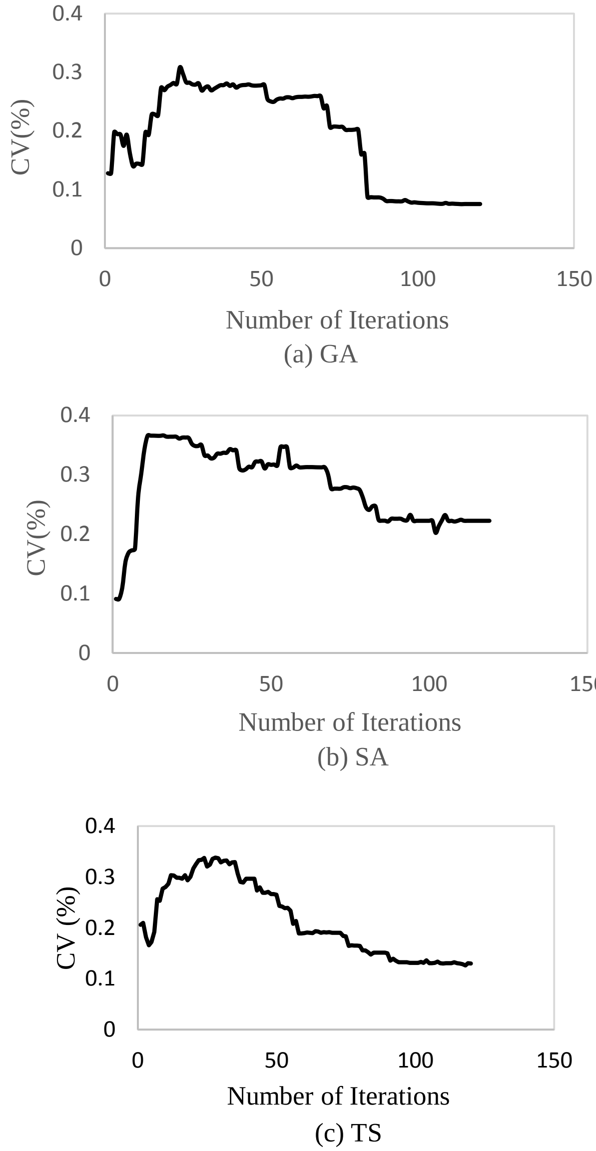


Figure 0-2 Coefficient of Variation of the Objective Function for a) GA, b) SA, c) TS

Chapter 7: Optimal Selection and Scheduling

In this study, an urban network with 24 nodes, 76 links and 20 improvement projects is selected. As discussed in previous sections, a traffic assignment model is designed to evaluate the candidate projects over the planning horizon and three metaheuristic algorithms are proposed to search for good near optimal solutions. This section analyzes the GA results and compares the basic scenario with no improvement projects with the scenario where projects are implemented.

Table 0-1 Optimal Sequence and Schedule

Optimal Sequence	Completion Time (year)
16 & 19	1.8
73 & 76	5.9
39 & 74	8.8
13 & 23	10.8
25 & 26	14.8
2 & 14	16.2
65 & 69	20.7
53 & 58	22.7
66 & 75	25.0
33 & 36	28.0
NPV of Total Cost$\times 10^6$	8535.93

Table 0-1 presents the optimal sequence and the corresponding schedule of projects along with the objective value. It should be reminded that the optimized schedule is directly determined by the sequence of selected projects, assuming it is reasonable to fund and finish each project one at a time, and gain its benefits as soon as it is completed. By this definition, the schedule of each project is appointed to a time when the available budget equalizes the project cost.

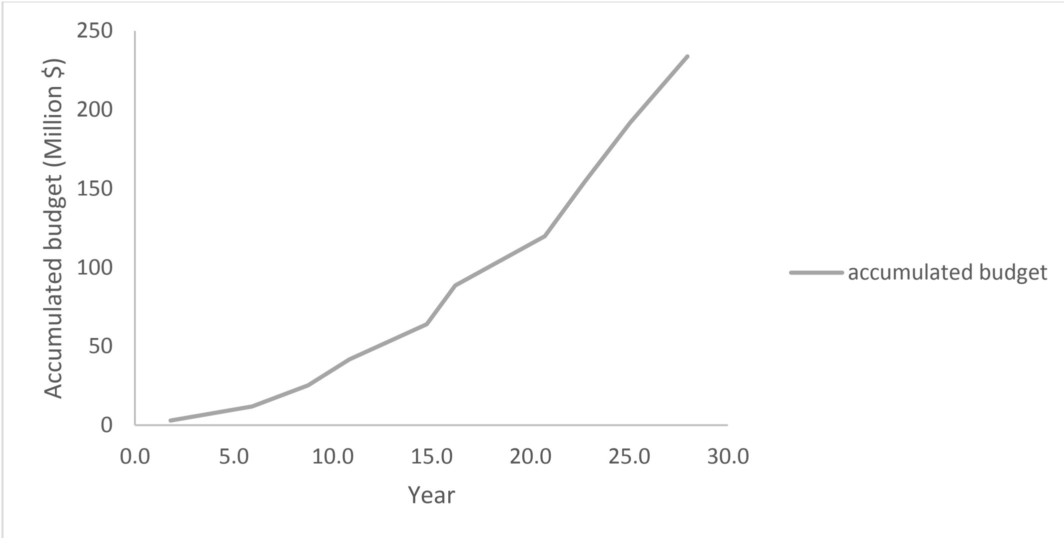


Figure 0-1 Accumulated Budget over Study Period

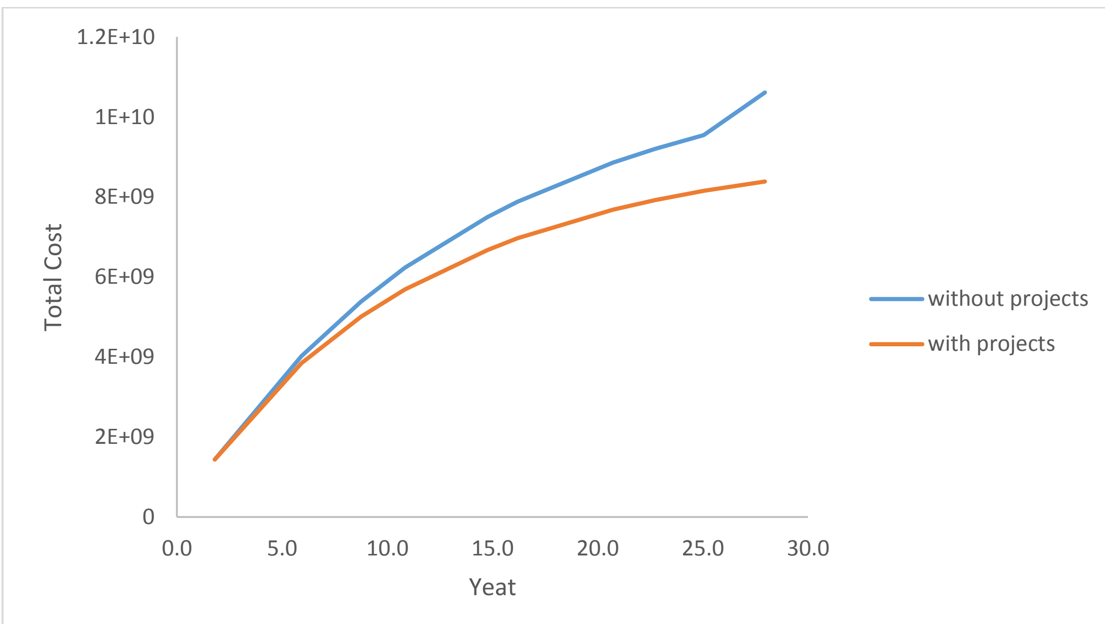


Figure 0-2 Accumulated Total Delay Cost with and without projects

As mentioned earlier in this study two sources of funding is considered: 1. External budget: which is assumed to be a simple linear function of time. 2. Internal budget: which is the fuel tax revenues collected from the users. Figure 0-1 demonstrates the overall budget flow over the planning horizon. Figure 0-2 shows the accumulated total delay costs with and without projects over the 30 years of analysis. These results indicate that at the end of 30 years, the improvement projects can save up to 21% of the total delay costs.

In general, no existing methods can guarantee that results of heuristic algorithms are globally optimal, and it is somewhat difficult to assess the goodness of solutions obtained by the evolutionary methods. In this study, a statistical experiment is conducted to examine the effectiveness of the algorithm. For this purpose, first a sample of randomly generated solutions is created. Ideally the sample must be created in a way that the solutions are independent of each other in order to satisfy the statistical requirements. The next step is to fit an appropriate distribution to the fitness values. The distribution of sample values should approximate the actual distribution since the sample is randomly generated. The final step is to calculate the cumulative probability of the solution found by the algorithm based on the fitted distribution. It is desirable to obtain a very low probability to prove the goodness of the solution.

Accordingly, a random sample of 50,000 solutions is created. Table 0-2 summarizes the statistical results drawn from this sample. After exploring different distributions, the Lognormal (mu= 9660, sigma= 0.0248) distribution appears to yield the best fit. Figure 0-3 shows the fitted distribution and the data derived from random sampling. It is evident that the minimum value in the distribution of 50,000 random solutions is higher (costlier) than the optimal solution presented in Table 0-1. In other words, the solution found by the algorithm excels all solutions in the distribution.

Table 0-2 Statistical results for the random solutions

Min	Max	Average	Standard Deviation
8709.19×10^6	15769.69×10^6	9421.77×10^6	236.38×10^6

The cumulative probability of the best solution found by the GA according to the Lognormal distribution is $p = F(x|\ \mu, \sigma) = F(8535.93 \times 10^6|\ 9660, 0.0248) = 3.597 \times 10^{-5}$ which can be derived from the following equation:

$$p = F(x|\ \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^x \frac{e^{\frac{-(\ln(t)-\mu)^2}{2\sigma^2}}}{t}\, dt \qquad (8)$$

This result implies that the best solution obtained with the algorithm dominates 99.999% of the random solutions in the distribution. Therefore, the solution found by the GA although not necessarily globally optimal, is very good compared to other possible alternatives in the solution space and the deviation from global optimality is likely to be very small compared to uncertainties and errors in the problem's inputs.

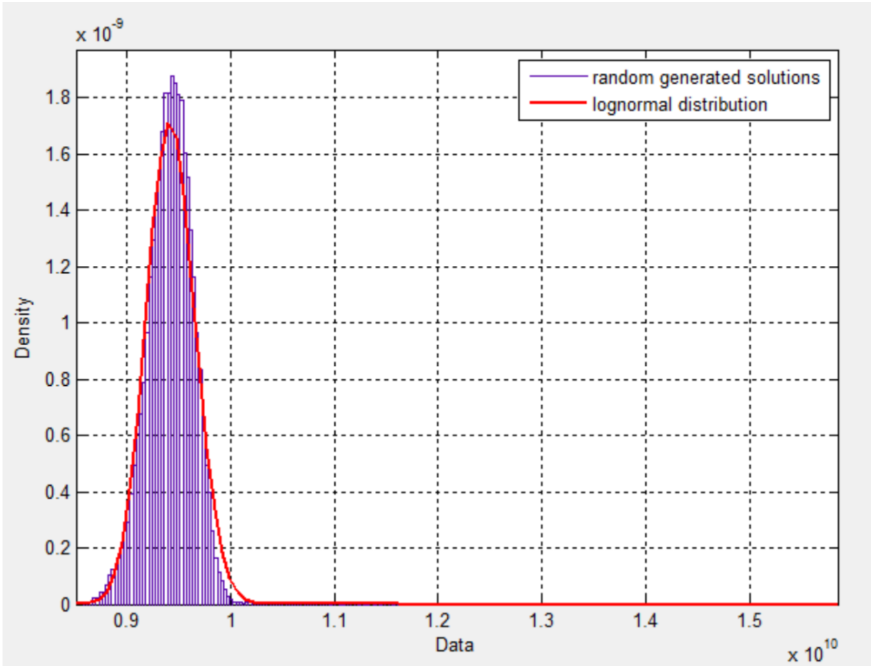


Figure 0-3 Fitted Lognormal distribution

Chapter 8: Sensitivity Analysis

This section studies of how the uncertainty in the output of the optimization model can be apportioned to different sources of uncertainty in inputs. This is useful in understanding model behavior and enhancing the efficiency of the proposed methodology. For this purpose, sensitivity analysis is conducted to investigate the effects based on both Genetic parameters and network specifications. Several factors such as population size, selective pressure, crossover and mutation rate influence the performance of genetic algorithms. Other factors such as the problem size are especially significant in determining the computation time thus, important in determining the efficiency and feasibility of the algorithm. This section provides sensitivity analysis based on population size, selective pressure, crossover/mutation rate and problem size that affects the GA performance and impact the optimization. This section continues with further analysis on network parameters and system settings such as demand growth rate, cost of projects and fuel tax rates. The goal is to examine the uncertainty of such parameters and observe their impacts on system output.

3.12 Population size

Population size is one of the key parameters of genetic algorithm. Usually decreasing population size increases optimization speed to a certain point, however, further decreasing may cause premature convergence. On the other hand, increasing population size increases optimization reliability. Therefore, a good selection of population size may reduce the optimization speed considerably. This experiment examines four different population sizes namely 10, 20, 30 and 40 while each experiment is implemented 10 times. In all four cases the search stops when convergence is reached for at least 5 generations.

Table 0-1 presents the optimization outcomes including the computation size, total system cost and the optimal sequence for different population sizes. Figure 0-1 shows that by increasing the population size the computation time grows dramatically, However as seen in Figure 0-2 increasing the population size results in sequences with lower objective function value (total cost over 30 year horizon). Therefore, it is important to set the population size such that balances between computation time and solution quality.

Table 0-1 Sensityivty Analysis (Population Size)

Population Size	*Computation Time (min)*	*Total Cost (*10^6 $)*	*Optimal Sequence*
10	73.34	85955.23	6,13,2,10,5,11,1,4,9,14
20	126.2660945	85390.99	6,2,13,11,8,5,10,1,17,9,16,18
30	270.8740385	85389.29	6,2,13,11,5,1,9,10,20,7,14
40	416.795875	85373.22	6,2,13,5,11,1,10,4,3,15

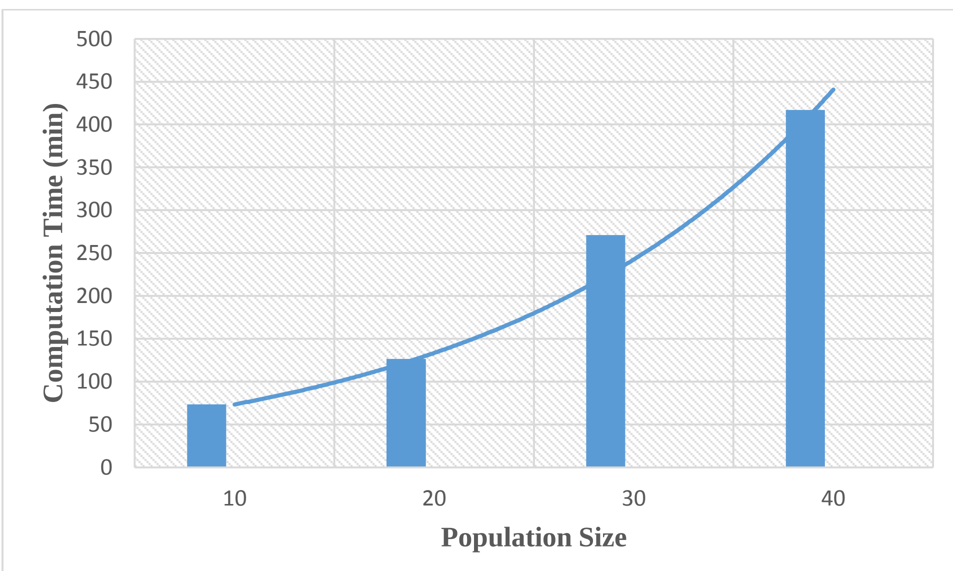


Figure 0-1 Computation Time for Different Population Size

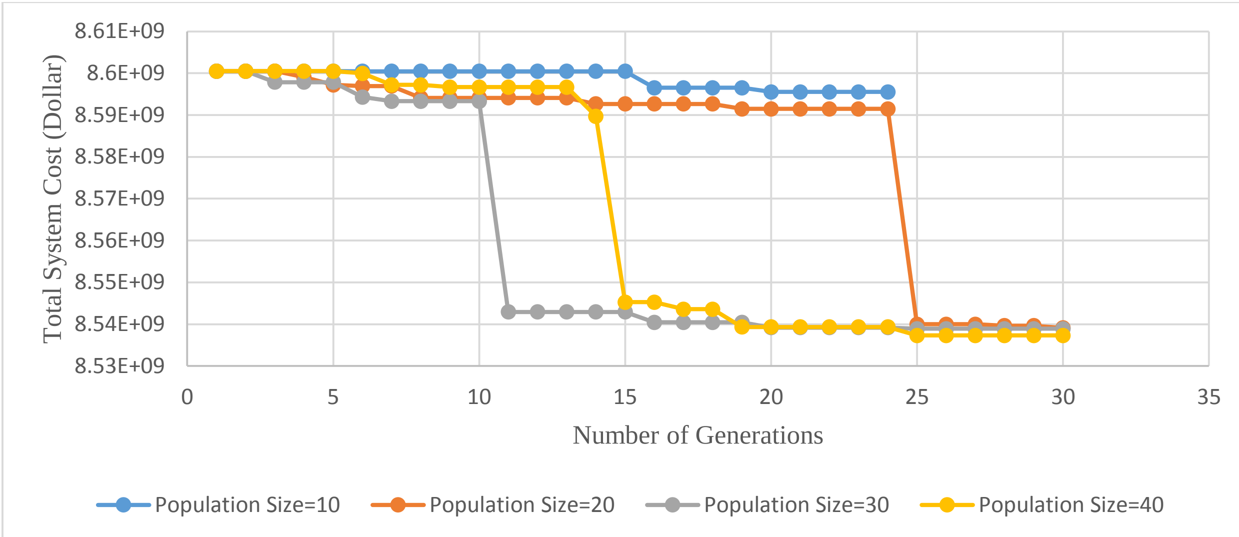


Figure 0-2 Optimization Process for Different Population Sizes

One can see significant improvement by increasing population size from 10 to 20. However, selecting population size of 30 or 40 does not yield much significant improvement in terms of solution quality but requires much more computation time. Thus it seems population size of 20 is the most reasonable choice.

3.13 Crossover and Mutation rate

A judicious choice of crossover and mutation rates is essential to the success of genetic algorithms. One should seek a proper balance between exploration and exploitation ability of the searching algorithm. In genetic algorithms, mutation operators are mostly used to provide exploration and cross-over operators are widely used to lead population to converge to a good sub optimal solution (exploitation). Consequently, while crossover tries to converge to a specific point, mutation attempts to avoid convergence and explore more areas. In more detail, crossover rate indicates a ratio of how many couples will be picked for mating, hence usually a higher rate is maintained with the expectation of converging faster using the already explored regions. Higher mutation rate increases the probability of searching more areas in search space, however, prevents population to converge to any optimum solution. On the other hand, too small mutation rate may result to premature convergence, and falling to local optima instead of global optimum. For this study, we examine crossover rates ranging from 0.2 to 0.5 and mutation rates from 0.1 to 0.3 separately while keeping the other parameters fixed. By default, the crossover rate is set to 0.5 and the mutation rate is set to 0.2 in the entire study.

Figure 0-3 indicates that a crossover value of 0.5 yields better a solution than the other values. Table 0-2 presents the optimal sequence and their corresponding objective function value for different crossover values. Accordingly, the crossover rate of 0.5 gives a better sequence with a total cost of $8532*10^6$ dollars over the entire study period.

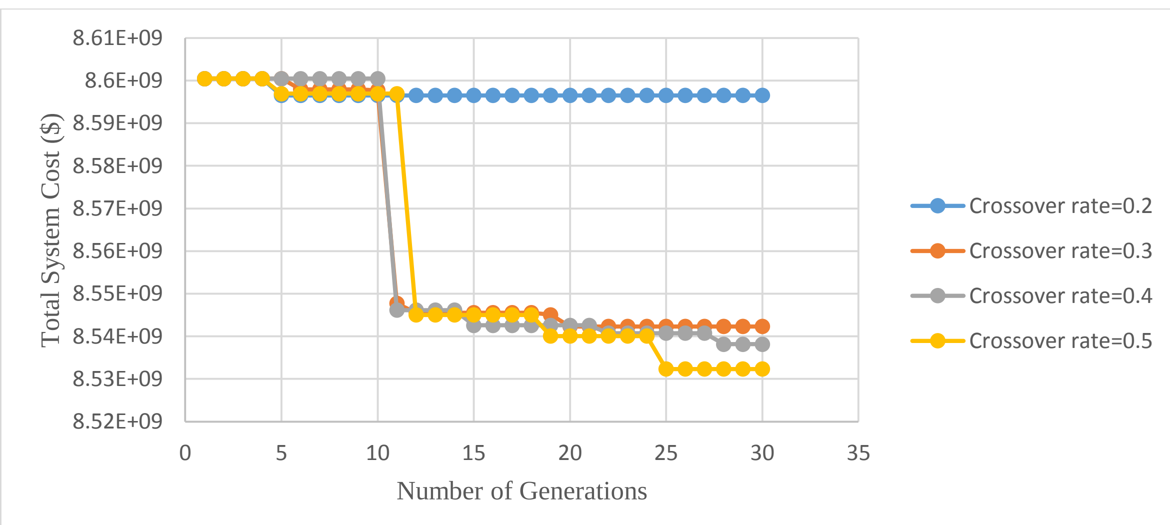


Figure 0-3 Optimization Process for Different Crossover Rates

Table 0-2 Sensitivity Analysis (Crossover Rate)

Crossover Rate	*Number of Generations*	*Total System Cost*	*Optimal Sequence*
0.2	30	8596.53	6,13,2,11,5,10,1,4,9,14
0.3	30	8542.33	6,13,2,11,5,10,1,8,15,19,20,12,17
0.4	30	8538.15	6,13,2,11,5,1,10,9,12,3
0.5	30	8532.33	6,2,13,5,11,10,1,4,12,15

Figure 0-4 indicates that a Mutation rate value of 0.2 yields better a solution than the other values. Similar to the crossover rates, Table 0-3 provides the optimal sequence and the total system cost for different mutation rates. According to the table, the mutation rate of 0.2 gives a better sequence with a total cost of $8548*10^6$ dollars over the entire study period. In this case, the mutation rate of 0.3 converges to a better solution with less total cost ($$8547*10^6$) but requires more generations to reach the optimal solution. Therefore the value of 0.2 is selected for this study.

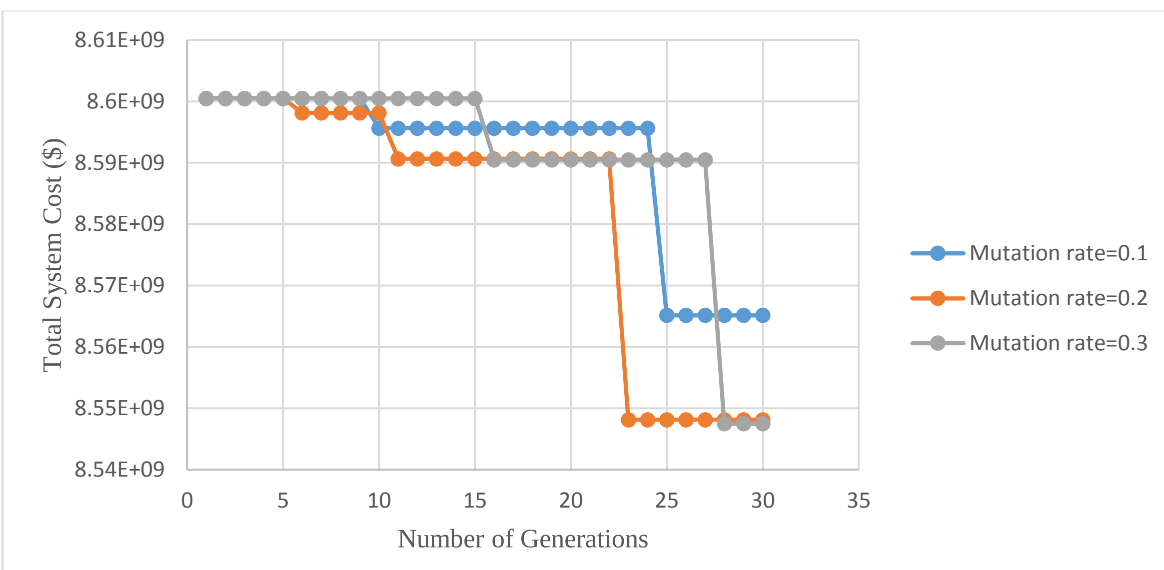


Figure 0-4 Optimization Process (Mutation Rate)

Table 0-3 Sensitivity Analysis (Mutation Rate)

Mutation Rate	*Number of Generations*	*Total System Cost*	*Optimal Sequence*
0.1	30	8565.13	6,13,2,11,5,10,1,3,16,18
0.2	30	8548.12	6,13,2,5,11,10,1,18,15,12,3
0.3	30	8547.48	6,13,2,10,5,11,1,4,15,3

3.14 Selective pressure

Selective pressure controls the tendency to select the best members of current parents to propagate to the next generation, and is a requirement to direct the GA to an optimum solution. On the other hand, maintaining the diversity of the population, is also required to ensure that the solution space is adequately searched, especially in the earlier stages of the optimization process. If the selective pressure is too high, the genetic diversity may decrease so that the global optimum is overlooked and the GA converges to a local optimum. However, if the selective pressure is too low, the GA may not converge to an optimum in a reasonable time. Selecting a proper value for selective pressure while maintaining the diversity shall lead to convergence in a reasonable time to a global optimum.

Figure 0-5 reveals the performance of the GA having to different selective pressure values. The results for both values seem relatively identical, however the selective pressure of 0.2 yields a slightly better solution with a total cost of \$8543$*10^6$ over the study period and converges to the optimum in earlier generations. Subsequently, a value of 0.2 is selected for the selective pressure in this research.

Table 0-4 Sensitivity Analysis (Selective Pressure)

Selective Pressure	*Number of Generations*	*Total System Cost*	*Optimal Sequence*
0.1	30	8544.30	6,2,13,11,5,10,1,4,3,15
0.2	30	8543.72	6,2,13,5,11,10,1,4,15,12,19

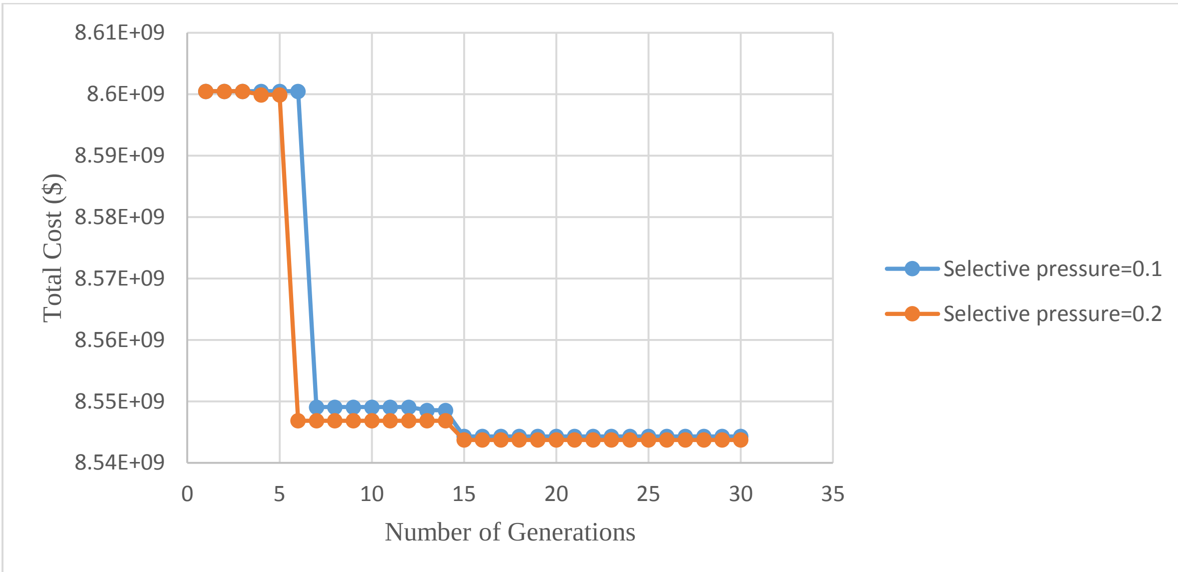


Figure 0-5 Optimization Process for Different Slective Pressure Values

3.15 Problem size

Computation time is very much related to the problem size and is of great concern. In general, computation time grows more than linearly as the problem size increases, thus important for investigation. In selecting and scheduling projects, problem size is defined as the number of candidate projects for implementation. While increasing the problem size, the population size should also increase to guarantee sufficient exploration of the solution space. It should be noted that due to project interdependency, the computation time is also related to network configuration. More specifically, if the problem size is the same for different network sizes (i.e. number of nodes and links), the computation time needed to solve the problem will be different. For this reason, a network with the same characteristics is tested in this section, and the only variable that changes is the problem size.

This section explores 5 alternatives with different problem sizes for 5, 7, 10, 15 and 20 candidate projects. The planning period is adjusted according to the problem size for more realistic results. Table 0-5 and Figure 0-6 show the optimization results and the computation time for each problem size.

Table 0-5 Sensitivity Analysis (Problem Size)

Problem size	*Population size*	*Computation time*	*Total system cost*	*Optimal sequence*	*Planning Horizon (years)*
5	10	38.78534174	63248.78	3,2,5,4,1	12
7	13	46.1506325	69301.94	3,2,5,4,1,6,7	15
10	15	78.58381202	80453.39	5,2,4,9,1,3,8,10,6,7	20
15	18	96.4942549	89128.00	5,2,11,4,14,1,6,9,8,15,10,7,13	25
20	20	126.2660945	85390.99	6,2,13,11,8,5,10,1,17,9,16,18	30

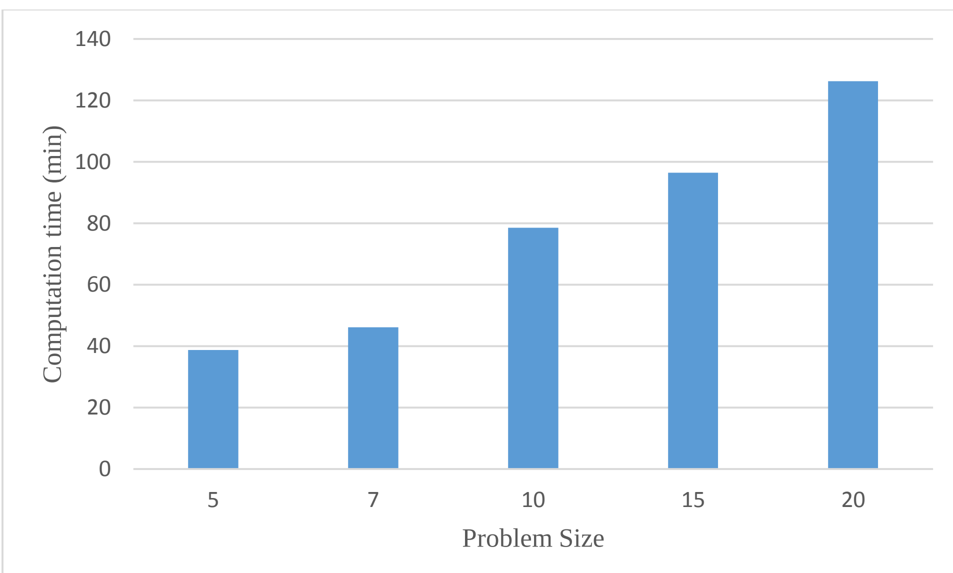


Figure 0-6 Computation Time for Different Problem Sizes

3.16 Demand growth rate

As explained in previous sections, it is assumed that the demand increases exponentially as a function of time over the planning horizon as follows:

$$d_{ij}^{t} = d_{ij}^{0} * (1 + r)^{t} \quad (1)$$

This section explores how changes in demand growth rate affect the optimization procedure and the optimal solution. For this purpose, growth rates of 0, 0.005, 0.01, 0.015 and 0.02 are tested. Too high growth rate may cause over saturation in the network which may require more time to reach user equilibrium or even cause system failure. On the hand, too low growth rate may not reflect the trend in real situation.

Table 0-6 Sensitivity Analysis (Demand Growth Rate)

Demand growth rate	*Computation time*	*Total system cost*	*Optimal sequence*
0	70.36	7445317456	6,2,13,5,8,10,16,19,7,1,20,18
0.005	85.85	7819339098	6,2,13,18,11,5,1,10,16,12,3
0.01	131.19	8431728530	6,2,13,18,11,5,1,10,16,12,3
0.015	272.72	8853373713	6,2,13,18,11,5,1,10,16,12,3
0.02	1567.48	10039171025	6,13,2,15,18,11,12,3,17,1,19

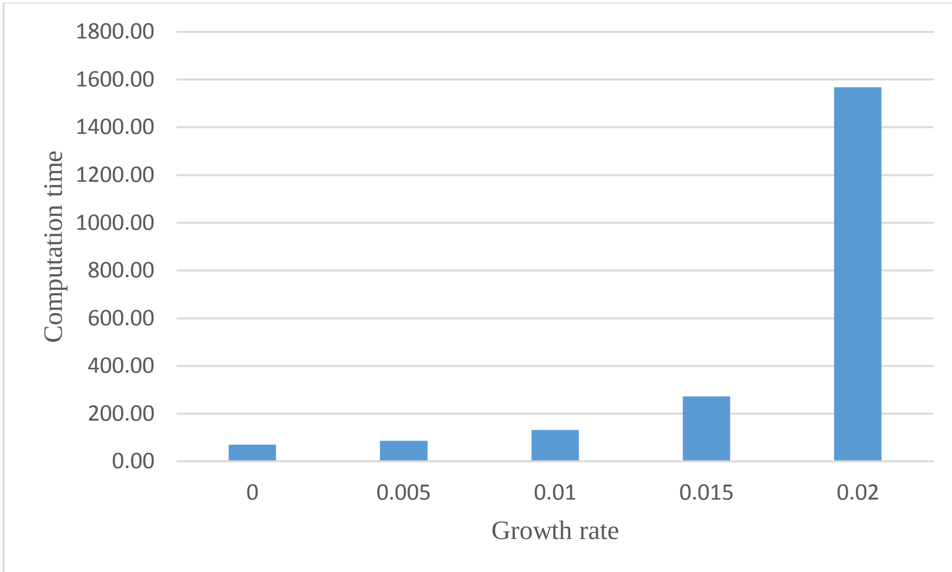


Figure 0-7 Computation Time for Different Demand Growth Rate Values

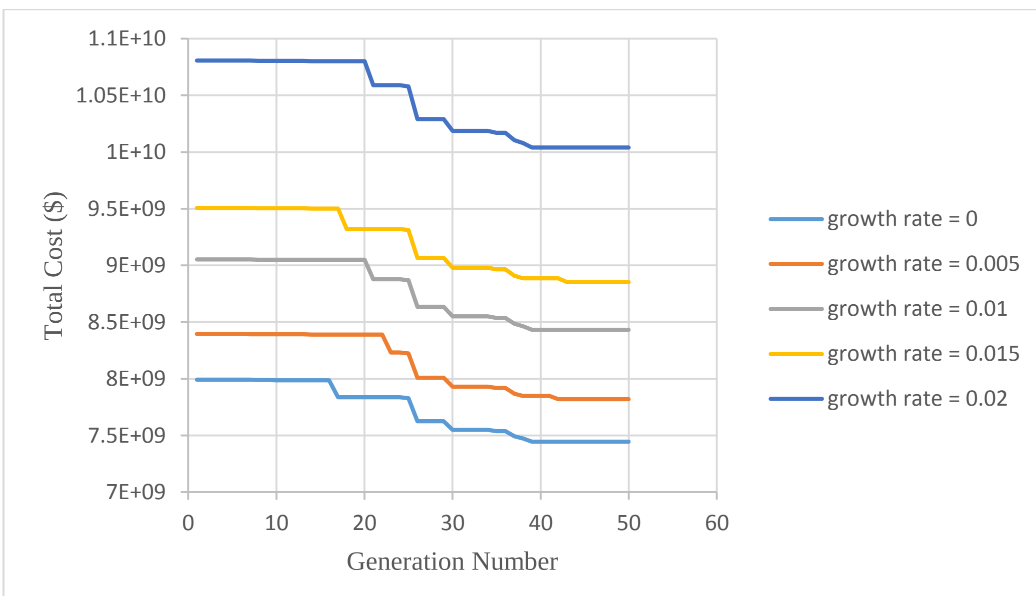


Figure 0-8 Optimization process for different demand growth rates

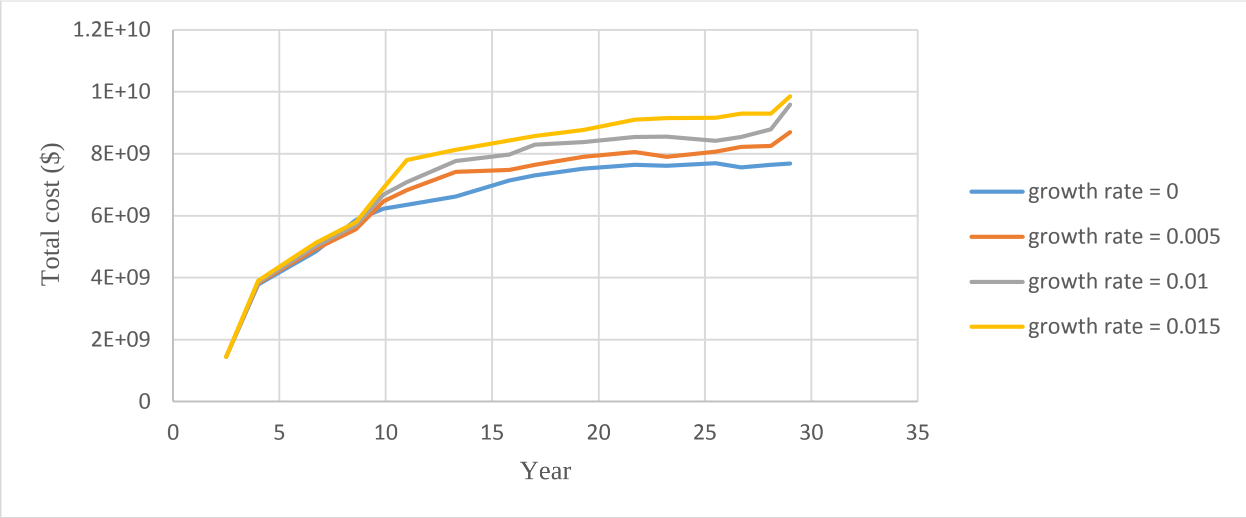


Figure 0-9 Total cost flow over analysis period

Table 0-6 presents the optimization results for different growth rate values. It can be seen that the optimal sequence alters as we increase the growth rate from zero, remains constant for values between 0.005 and 0.015 and again changes when we use 0.02 growth rate. This suggests that the optimal sequence somewhat depends on the assumed growth rate and the accuracy of demand forecasting can be very crucial in deciding the final project sequence. Figure 0-7 gives a visual comparison for computation time of different growth rates. It can be seen that the analysis time generally increases as the growth rate is increasing but rises drastically as we approach to 0.02 growth rate. This is due to the fact that the increased demand may lead to an over saturation circumstance that causes disproportionate convergence time in the traffic assignment model, or even lead to system failure. Figure 0-8 depicts the optimization procedure which indicates how the GA converges to the optimal solution. As expected, the overall system cost shifts up as the growth rate increases because naturally the user cost (travel time cost) increases as more users enter the system.

Figure 0-9 compares the flow of the total cost over the 30 year analysis period for different values of growth rate. It is evident that the slope of the total cost growth increases as the demand growth gets higher.

3.17 Project costs

As described in previous sections, the budget constraint is defined as follows:

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

$$\begin{cases} x_i(t) = 0 & if\ t < t_i \\ x_i(t) = 1 & if\ t > t_i \end{cases}$$

where t_i is the time when project *i* is finished, $x_i(t)$ is a binary variable specifying whether project *i* is finished by time *t* and c_i is the cost of project *i*. In this study the cost of each project is a function of the length of improvement. For this purpose, the cost of adding a lane is considered $4,000,000 per lane mile. This section explores how variations of project costs may affect the optimization results by increasing the cost per lane mile up to 5%, 10% and 20%. Table 0-7 shows the optimizing results for different widening costs. It is evident that the total cost of optimal solution increases as project costs increase. It is also clear that the optimal order of projects changes and fewer number of projects are accommodated in the sequence because the cost of projects increases while the budget remains unchanged. Figure 0-10 provides a more detailed demonstration of how the total cost changes over the study period.

Table 0-7 Sensitivity Analysis (Project Cost per Lane)

Widening cost per lane mile	*Total system cost*	*Optimal sequence*
$4,000,000	8611886171	6,2,13,18,11,5,1,10,16,12,3
$4,200,000	9747438334	6,2,13,11,5,10,1,12,4,20
$4,400,000	9570418735	6,2,13,5,8,11,9,1,10,4
$4,800,000	9983438568	6,2,13,8,5,11,1,10,16

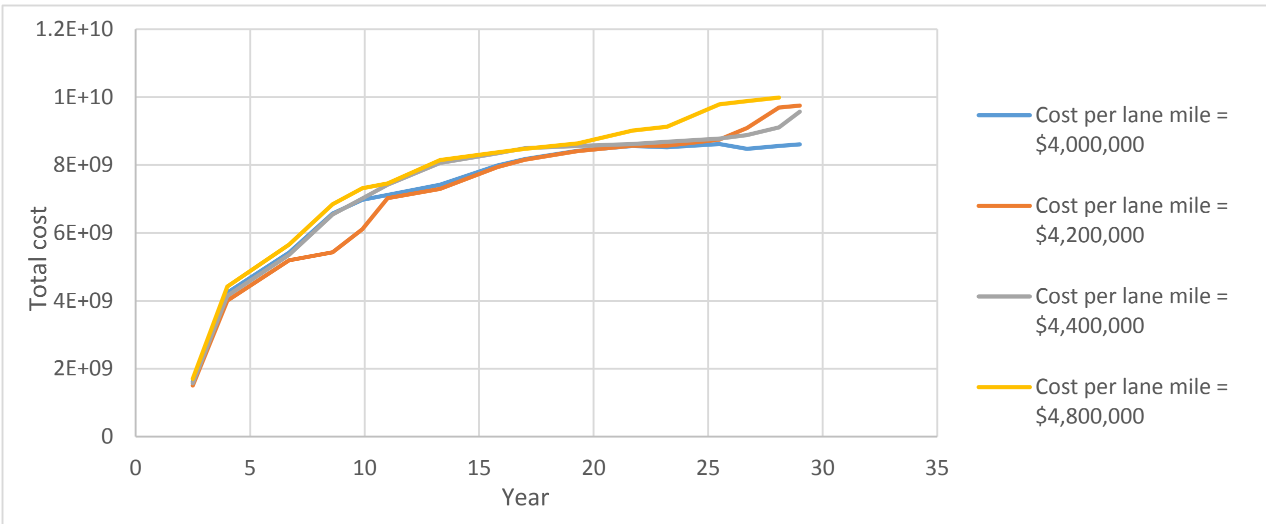


Figure 0-10 Total cost flow over analysis period

Chapter 9: Conclusion and Summary

The selection and scheduling of interrelated projects is an interesting problem for policy makers and researchers in various fields, including economics, operation research, business, management, and transportation. Although it is crucial to consider the interrelation among projects when evaluating and prioritizing them, the problem is not sufficiently solved in the literature. This study combines a simple traffic assignment model for evaluating the objective function with three meta-heuristic algorithms, including genetic, simulated annealing and tabu search, for optimizing the sequence and schedule of the interrelated expansion projects. In particular, the optimized schedule is directly determined by the sequence of selected projects. More specifically, under the limited budget, which is continuously distributed over time, it is reasonable to fund and finish each project one at a time, and gain its benefits as soon as it is completed.

The main contribution of this research is to provide an extensive comparative analysis in terms of quality, speed and consistency for the three meta-heuristics used. A second contribution is considering an internal source of budget for future projects. More specifically, the fuel tax revenues from users is added to an external budget, constituting the overall budget for next projects. Another significant contribution is to account for the possibility that projects may become economically unjustified after the implementation of previous projects, before the end of the study period. In order to apply the proposed algorithms and demonstrate the numerical results, a sample network is examined through the evaluation and optimization process. The outcomes are further used for a multilateral comparison.

After finding the optimum sequence and schedule of the projects, the comparative analysis indicates that the GA, SA and TS decrease the present worth of the total cost by 596, 577 and 589 million dollars, respectively, compared to a congestion-ranked solution, thus indicating that the GA yields a better solution with less total cost than the other two. However, the SA and TS reach better solutions in the earlier stages of the search and thus seem preferable if budgets for computation are limited. 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 similar final solutions after a sufficient number of iterations. In addition to comparing the metaheuristic algorithms, an extensive sensitivity analysis is performed on both genetic parameters and system specifications. The goal is to acquire a comprehensive understanding of the uncertainty in the output of the optimization model that can be apportioned to different sources of uncertainty in inputs. The analysis consists of testing the following parameters: population size, crossover/mutation rate, selective pressure, problem size, demand growth rate, project costs and fuel tax rates.

In a nut shell, the major objective this study is to set an example of a more general applicable method for optimizing planning and scheduling decisions on infrastructure while considering the interrelation among them and major relevant uncertainties. This method is initially set for a road network problem but can be easily extended to not only other transportation infrastructure applications (e.g. airports, rail transit routes and inland waterways), but also beyond transportation.

References

Bagloee, S. A., and Tavana, M. (2012). An efficient hybrid heuristic method for prioritising large transportation projects with interdependent activities. International Journal of Logistics Systems and Management, 11(1), 114-142.

Bhattacharyya, R., Kumar, P., and Kar, S. (2011). Fuzzy R&D portfolio selection of interdependent projects. Computers & Mathematics with Applications, 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.

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

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

Clark, J. J., Hindelang, T. J., and Pritchard, R. E. (1989). "Capital budgeting: planning and control of capital expenditures." Prentice Hall.

Cochran, M. A., Pyle, E. B., Greene, L. C., Clymer, H. A., and Bender, A. D. (1971). “Investment model for R and D project evaluation and selection.” *IEEE T. Eng. Manage.*, (3), 89.

Davis, L. (1985), “Applying Adaptive Algorithms to Domains”, Proceedings of International Joint Conference on Artificial Intelligence, pp.162-164.

Disatnik, D. J., & Benninga, S. (2007). Shrinking the covariance matrix. The Journal of Portfolio Management, 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.*, 13.3: 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.*, 13(2), 115-123.

Erlenkotter, D. (1973). “Sequencing of interdependent hydroelectric projects.” *Water Resour. Res.*, *9*(1), 21-27.

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

Fox, G. E., Baker, N. R. and Bryant, J. L. (1984), "Economic Models for R and D Project Selection in the Presence of Project Interactions", Management Science, Vol. 30, No. 7.

Gear, A. E. and Cowie, G. C. (1980), "A Note on Modeling Project Interdependence in Research and Development", Decision Science, Vol. 11, pp. 738-748.

Gen M., and Cheng R. (1997). "Genetic algorithms & engineering design." Wiley Interscience.

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

Goldberg, D. and R. Lingle (1985), "Alleles, Loci and the Traveling Salesman Problem", Proceedings of the First International Conference on Genetic Algorithms, Lawrnece Erlbaum Associates, Hillsdale, NJ, pp.154-159.

Goldberg, D. E. (1996). "Genetic algorithms in search, optimization and machine learning." Addison Wesley.

Janson, B. N. (1988), "Method of Programming Regional Highway Projects", Journal of Transportation Engineering, ASCE, Vol. 114, No. 5, pp.584-605.

Janson, B. N., Buckels, L. S., and Peterson, B. E. (1991). “Network design programming of US highway improvements.”*J. Transp. Eng-ASCE*, 117(4), 457-478.

Jong, J. C., and Schonfeld, P. (2001). Genetic algorithm for selecting and scheduling interdependent projects. Journal of waterway, port, coastal, and ocean engineering, 127(1), 45-52.

Kirkpatrick, S., and Vecchi, M. P. (1983). Optimization by simmulated annealing.science, 220(4598), 671-680.

Wang, S. L., and Schonfeld, P. (2005). “Scheduling interdependent waterway projects through simulation and genetic optimization.” *J. Waterway, Port, Coastal, Ocean Eng*, 131(3), 89-97.

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.

Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1), 77-91

Martinelli, D. (1993). "Investment planning of interdependent waterway improvement projects.", Ph.D. Dissertation, University of Maryland, College Park.

Martinelli, D., and Schonfeld, P. (1995). "Approximating delays at interdependent locks.", *J. Waterway, Port, Coastal, Ocean Eng*, 121(6), 300-307.

Michalewicz, Z. (1995)." Genetic algorithms + data Structure = evolution programs.", 3^{th} Edition, Springer.

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.

Mollanejad, M., and Zhang, L. (2014). "Incorporating Equity into Interurban Road Network Design." *Transport Geography*, 39: 156-164

Mollanejad, M. , Vemulapati, S.V. , and Zhang, L. (2014). “Supporting sustainable megaregion development through optimizing the impacts of transportation investments.” *ASCE Journal of Urban Planning and Development*, 1-10.

Morin, T. L., and Esogbue, A. M. (1971)."Some efficient dynamic programing algorithms for the optimal sequencing and scheduling of water supply projects." *Water Resour. Res.* 7(3) 479-484.

Morin, T. L., and Esogbue, A. M. (1974). “A useful theorem in the dynamic programing solution of sequencing and scheduling problems occurring in capital expenditure planning.” *Water Resour. Res.*, 10(1), 49-50.

Nemhauser, G. L., and Ullmann, Z. (1969). “Discrete dynamic programming and capital allocation.” *Manage. Sci.*, 15(9), 494-505.

Sivaraj, R., and Ravichandran, T. (2011). “A review of selection methods in genetic algorithm.” *International Journal of Engineering Science and Technology*, 3(5), 3792-3797.

Syswerda, G. (1991), “Scheduling Optimization Using Genetic Algorithms”, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, pp.824-829.

Szimba, E., and Rothengatter, W. (2012). “Spending scarce funds more efficiently—including the pattern of interdependence in cost-benefit analysis.” *J. Infrastruct. Syst.*, 18(4), 242-251.

Tao, X., and Schonfeld, P. (2005). “Lagrangian relaxation heuristic for selecting interdependent transportation projects under cost uncertainty.” *Transp. Res. Record: Journal of the Transportation Research Board*, 1931(1), 74-80.

Tao, X., and Schonfeld, P. (2007) “Island Models for a Stochastic Problem of Transportation Project Selection and Scheduling.” *Transp. Res. Record*, 2039, 16-23.

Wang, S. L., and Schonfeld P. (2005). "Scheduling interdependent waterway projects through simulation and genetic optimization." *J. Waterway, Port, Coastal, Ocean Eng,* 131(3), 89-97.

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

Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). “Project scheduling with finite or infinite number of activity processing modes–A survey.” *Eur. J. Oper. Res.*, 208(3), 177-205.

Wei, C. H., & Schonfeld, P. M. (1993). "An artificial neural network approach for evaluating transportation network improvements." *J. Adv. Transport.*, 27(2), 129-151.

Weingartner, H. M. (1966). "Capital budgeting of interrelated projects: survey and synthesis." *Manage. Sci.*, 12(7), 485-516.

Whitley, D., Starkweather, T., and Fuquay, D'A. (1989) , "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmanm Publishers, San Mateo, CA, pp. 133-140.

Zhang, L., and Levinson, D. (2008). "Investing for reliability and security in transportation networks." *Journal of the Transportation Research Board*, 2041: 1-10.

Zhang, L., Levinson, D., and Zhu, S. (2008). "Agent-based model of price competition, capacity choice, and product differentiation on congested networks." *Journal of Transport Economics and Policy*, 42(3): 435-461.

Zhang, L., and Levinson, D. (2009). "The economics of road network ownership: An agent-based approach." *International Journal of Sustainable Transportation*. 3(5):339-359.

Zhang, L., and Yusufzyanova, D. (2012). "Impact of regulation and network topology on effectiveness of roadway privatization." *Journal of the Transportation Research Board*, 2297: 172–180.

Chen, X. , Zhu, Z. , He, X. , and Zhang, L. (2015), "Surrogate-based optimization for solving mixed integer network design problem." *Journal of the Transportation Research Board*