Global Optimization Approach for Deterministic, Stochastic and Robust Optimizations of Dynamic Integrated Network Design and Traffic Signal Setting Design Problem: Comparison and Evaluation Results

Ampol KAROONSOONTAWONG
Lecturer, Department of Civil Engineering
King Mongkut's University of Technology Thonburi
126 Pracha-Utit Road, Bangmod, Thung-Khru,
Bangkok, 10140 Thailand
Fax: +662 427-9063; Email: ampolk@gmail.com

Abstract: The formulations of the user-optimal dynamic traffic assignment (UODTA)-based network design problem (NDP), signal setting design problem (SSD), and the integrated NDP and SSD problem (NDP-SSD) for the deterministic (DET), stochastic (SP) and robust (RO) optimizations are reviewed. The modifications of the simulated annealing (SA), genetic algorithm (GA) and reactive tabu search (RTS) for the nine problems (DET/SP/RO of NDP/SSD/NDP-SSD) are discussed. In the experiment, SA, GA and RTS are compared for the nine problems to identify best algorithms. The best algorithm for each problem is employed to find a best solution, which is then evaluated under stochastic condition. We find that the RO solutions yield the best robustness and lowest risk, whereas the DET solutions perform worst. The integrated robust approach is the most desirable, and the next best approach is the sequential robust approach. The deterministic approach can yield worse solutions than the do-nothing case.

Key Words: *Network Design Problem, Traffic Signal Setting Design, Robust Optimization, Global Optimization*

1. INTRODUCTION

The network design problem (NDP) determines an optimal budget allocation policy for expanding the capacity of pre-specified links while accounting for the user-optimal dynamic traffic assignment (UODTA) condition and the budget constraint. The signal setting design problem (SSD) determines an optimal signal setting strategy (cycle lengths, green times, time offsets and phase sequences) for the pre-specified signalized intersections while accounting for UODTA. The SSD in this paper is referred to the pre-timed signal setting design for the operational planning. The integrated NDP and SSD problem (NDP-SSD) simultaneously determines an optimal budget allocation policy and an optimal signal setting strategy while accounting for the UODTA condition and the budget constraint. These problems are bi-level in nature; i.e. the upper level problem (transport planner) minimizes the total system travel time (TSTT), and the lower-level problem observes the decisions made by the upper-level and behaves in the user-optimal manner (i.e. the UODTA condition). The three problems (NDP, SSD and NDP-SSD) can be formulated as deterministic, two-stage stochastic linear programming (SLP2), and robust optimization (RO) models. The deterministic model assumes all problem parameters are known and fixed. The SLP2 and RO models assume that the origin-destination (OD) demands are uncertain with known probability distributions. The SLP2 minimizes the expected TSTT, while the RO minimizes the expected sum of TSTT and

risk. Karoonsoontawong and Waller (2008) formulated the robust optimization model for the NDP-SSD (RO-BLPNDP-SSD), and showed that it can be reduced to eight related models in Table 1. These mathematical formulations are limited to only single-destination networks due to the underlying UODTA model, so the useful, practical application on multi-destination larger-size networks are prohibited. Moreover, the bi-level formulations are NP-hard (Bard, 1998), so there is not an efficient solution method that guarantees an optimal solution. To address these limitations, Karoonsoontawong and Waller (2006) applied the genetic algorithm (GA) and simulated annealing (SA) to efficiently solve the deterministic NDP, and Karoonsoontawong and Waller (2009) applied the reactive tabu search (RTS) to the deterministic SSD. GA, SA and RTS are global optimization techniques can efficiently find solutions beyond local optimality.

GA, SA and RTS typically perform efficiently on np-hard problems. However, a technique may outperform the others on a particular problem, but may be outperformed on another problem. This paper addresses an interesting question that which global optimization algorithm performs best on each of the nine np-hard problems in Table 1. Further, the best solutions from the nine problems are evaluated under stochastic environment in order to compare the nine problems and especially to demonstrate the virtue of the robust integrated problem. In practice, demand uncertainty is often ignored, and only average demands are used in solving NDP, SSD or NDP-SSD. This paper offers alternative approaches (stochastic and robust optimization schemes) to explicitly capture the demand uncertainty, and develop practical, efficient algorithms for these problems. Moreover, a real network is employed to demonstrate how much benefit (in terms of expected robustness, expected TSTT and expected risk) one can obtain by the proposed robust optimization model when compared with the deterministic or stochastic models.

In this paper, we compare the performance of SA, GA and RTS on a test network for the nine problems, and evaluate the solutions of the nine problems under the stochastic condition in terms of the expected robustness, the expected TSTT and the expected risk. The paper is organized as follows. Chapter 2 reviews the nine formulations, followed by the review of the three global optimization techniques in Chapter 3. Then, the modifications to the three global optimization techniques for the nine problems are discussed in Chapter 4. Chapter 5 shows the computational experiment. Chapter 6 summarizes and concludes the study.

2. REVIEW OF RELEVANT FORMULATIONS

Karoonsoontawong and Waller (2008) proposed a robust optimization model for the combined UODTA, NDP and SSD problem (RO-BLPNDP-SSD) that embeds Daganzo (1994)'s cell transmission model (CTM). Only the OD time-dependent demands are considered random with known discrete probability distribution, and the stochastic demands are assumed independent across time intervals and cells. The RO-BLPNDP-SSD is given below.

<u>Notation</u>

Sets

C ; C_S ; C_R = set of cells; set of sink cells; set of source cells

T = set of discrete time intervals

E ; E_S = set of cell connectors; set of sink cell connectors
$FS(i)$; $RS(i)$ = set of connectors emanating from cell *i*; set of connectors emanating to cell *i*
SI = set of signalized intersection cells
P(i,j) = set of phases which allow flow from cell *i* to signalized intersection cell *j*
STP_i = set of signal timing plans for intersection cell *i*
P_i = set of all phases for intersection cell *i*
Ω = set of all possible demand realizations

Parameters
B = total available budget
δ_i^t = ratio of link free flow speed and backward propagation speed
M_t = cost per time interval that yields user optimal flows in UODTA
$\tilde{d}_i^t$ = stochastic demand in cell *i* at time interval *t*
N_i^t = max number of vehicles in cell *i* at time interval *t*
Q_i^t = max number of vehicles that can flow into or out of cell *i* at time *t*
χ_i = increase in N_i^t per one unit of b_i (= χ in this paper)
ϕ_i = increase in Q_i^t per one unit of b_i (= ϕ in this paper)
phase(i,t) = phase number of signal timing plan *i* at time *t*
p^ω = probability of demand scenario ω
τ = the fixed target total system travel time (or $TSTT^{TARGET}$)
λ = the goal programming weight on the expected risk in the objective function

Variables
x_i^t = number of vehicles in cell *i* at time interval *t*
y_{ij}^t = number of vehicles moving from cell *i* to cell *j* at time interval *t*
s_k = 1 if the signal timing plan *k* is chosen, and 0 otherwise
p_{ij}^t = 1 if flow is allowed from cell *i* to *j* at time *t*, and 0 otherwise
D^ω = the risk (the up-sided deviation of TSTT from τ) for the demand scenario ω

<u>Formulation</u>

$$Robust\_obj(\lambda) = \min_{b,s,p,D} \sum_{\omega\in\Omega}\sum_{(i,j)\in E_S}\sum_{t\in T}\left(p^\omega \cdot t \cdot y_{ij}^{t,\omega}\right) + \lambda \sum_{\omega\in\Omega}\left(p^\omega \cdot \left(D^\omega\right)^2\right) \quad (1.1)$$

$$\text{subject to } D^\omega \geq \sum_{(i,j)\in E_S}\sum_{t\in T}\left(t \cdot y_{ij}^{t,\omega}\right) - \tau \quad \forall \omega \in \Omega \quad (1.2)$$

$$D^\omega \geq 0 \quad \forall \omega \in \Omega \quad (1.3)$$

$$\sum_{i\in C\setminus C_S} b_i \leq \mathrm{B} \quad (1.4)$$

$$b_i \geq 0 \quad \forall i \in C \setminus C_S \quad (1.5)$$

$$\sum_{k\in STP_j} s_k = 1 \quad \forall j \in SI \quad (1.6)$$

$$p_{ij}^t = \sum_{k\in STP_j : phase(k,t)\in P(i,j)} s_k \quad \forall (i,j) \in RS(j), \forall j \in SI, \forall t \in T \quad (1.7)$$

$$s_k \in \{0,1\} \ \forall k \in \cup_{j\in SI} STP_j \quad (1.8)$$

$$p_{ij}^{t} \geq 0 \quad \forall (i,j) \in RS(j), \forall j \in SI, \forall t \in T \tag{1.9}$$

$$\min_{x,y} \sum_{\omega \in \Omega} \sum_{(i,j) \in E_S} \sum_{t \in T} M_t y_{ij}^{t,\omega} \tag{1.10}$$

$$\text{subject to} \sum_{(i,j) \in FS(i)} y_{ij}^{t,\omega} - x_i^{t,\omega} \leq 0 \quad \forall i \in C \setminus C_S, t \in T, \ \forall \omega \in \Omega \tag{1.11}$$

$$x_i^{t,\omega} - x_i^{t-1,\omega} + \sum_{(i,j) \in FS(i)} y_{ij}^{t-1,\omega} - \sum_{(j,i) \in RS(i)} y_{ji}^{t-1,\omega} = \tilde{d}_i^{t,\omega} \ \forall i \in C \setminus C_s, t \in T, \forall \omega \in \Omega \tag{1.12}$$

$$\sum_{(j,i) \in RS(i)} y_{ji}^{t,\omega} \leq \delta_i^t (N_i^t + \chi_i \cdot b_i - x_i^{t,\omega}) \quad \forall i \in C \setminus (C_R \cup C_S), t \in T, \ \forall \omega \in \Omega \tag{1.13}$$

$$\sum_{(i,j) \in FS(i)} y_{ij}^{t,\omega} \leq Q_i^t + \phi_i \cdot b_i \qquad \forall i \in C \setminus C_s, t \in T, \ \forall \omega \in \Omega \tag{1.14}$$

$$\sum_{(j,i) \in RS(i)} y_{ji}^{t,\omega} \leq Q_i^t + \phi_i \cdot b_i \quad \forall i \in C \setminus (C_R \cup C_S \cup SI), t \in T, \ \forall \omega \in \Omega \tag{1.15}$$

$$y_{ji}^{t,\omega} \leq Q_i^t + \phi_i \cdot b_i \quad \forall (j,i) \in RS(i), i \in SI, t \in T, \ \forall \omega \in \Omega \tag{1.16}$$

$$y_{ji}^{t,\omega} \leq p_{ji}^t \left(Q_i^t + \phi_i \cdot \mathrm{B}\right) \quad \forall (j,i) \in RS(i), i \in SI, t \in T, \forall \omega \in \Omega \tag{1.17}$$

$$x_i^{0,\omega} = 0 \quad \forall i \in C \setminus C_S, \ y_{ij}^{0,\omega} = 0 \quad \forall (i,j) \in E, \forall \omega \in \Omega \tag{1.18}$$

$$x_i^{|T|,\omega} = 0 \quad \forall i \in C \setminus C_S, \forall \omega \in \Omega \tag{1.19}$$

$$x_i^{t,\omega} \geq 0 \quad \forall i \in C \setminus C_S, t \in T, \ y_{ij}^{t,\omega} \geq 0 \quad \forall (i,j) \in E, t \in T, \forall \omega \in \Omega \tag{1.20}$$

The objective (1.1) minimizes the expected robust objective (*E[Robust_Obj(λ)]*), which is the weighted sum of the expected TSTT (*E[TSTT]*) and the expected risk (*E[Risk]*). *E[TSTT]* for the analytical study is $\sum_{\omega \in \Omega} \sum_{(i,j) \in E_S} \sum_{t \in T} (p^{\omega} \cdot t \cdot y_{ij}^{t,\omega})$. *E[Risk]* is $\sum_{\omega \in \Omega} p^{\omega} (\max(0, \sum_{(i,j) \in E_S} \sum_{t \in T} t \cdot y_{ij}^{t,\omega} - \tau))^2$, which can be implied from (1.1)-(1.3). (1.4) is the budget constraint. Variable b_i is defined as the amount of the budget spent on cell i to increase Q_i^t and N_i^t by $\phi_i \cdot b_i$ and $\chi_i \cdot b_i$, respectively. (1.6) enforces that only one signal timing plan is chosen for each intersection. The possible signal settings (i.e. cycle lengths, green times, time offsets and phase sequences) are limitedly enumerated before the optimization. (1.7) determines p_{ij}^t from the sum of signal timing plan variables (s_k) of signalized intersection cell j that allow flows from i to j at time t. This implies that the variables p_{ij}^t need not be constrained to be binary (1.9). (1.8) constrains s_k to be binary. The objective of the nested UODTA (1.10) minimizes a function that makes all vehicles behave in the UO condition (Ukkusuri and Waller, 2007). There are two basic traffic flow relationships embedded in the nested UODTA program. The first is the cell mass conservation (Eq.1.12). The second states that the traffic flow between two cells is constrained by the number of vehicles occupying the upstream cell (1.11), the remaining capacity of the downstream cell (Eq.1.13), and the maximum flow that can get out of the upstream cell and into the downstream cell (1.14)-(1.15). (1.16)-(1.17) concerns signalized intersection cells: if p_{ji}^t =1, y_{ji}^t is constrained by $Q_i^t + \phi_i \cdot b_i$; and if the chosen signal timing plan does not allow flow from cell j to cell i at time t (p_{ji}^t =0), then y_{ji}^t is constrained to be zero. (1.19) enforces that all vehicles must reach the destination by the last time interval. Without this constraint, the optimal solution is erroneous; i.e. all vehicles do not leave the origin.

Two special cases of RO-BLPNDP-SSD are SLP2-BLPNDP-SSD and BLPNDP-SSD (see Table 1 for abbreviation descriptions). When we set λ=0 in (1.1), (1.2)-(1.3) can be removed,

and the resulting model is SLP2-BLPNDP-SSD. When the uncertain parameters ($\tilde{d}_i^t$) in SLP2-BLPNDP-SSD become deterministic, the resulted model is BLPNDP-SSD. The integrated NDP and SSD models can easily be reduced to either NDP models or SSD models. For example, RO-BLPNDP-SSD can be reduced to RO-BLPSSD by setting B to zero; RO-BLPNDP-SSD can be reduced to RO-BLPNDP by deleting (1.6)-(1.9) and setting *SI* to an empty set.

3. REVIEW OF GLOBAL OPTIMIZATION TECHNIQUES

A global optimization (a.k.a. metaheuristic) refers to a master procedure that guides and modifies other heuristics to produce solutions beyond those that are normally generated for a local optimum (Glover, 1989). The heuristics guided by such a metaheuristic may be high-level algorithms or a simple description of available moves from the current solution to the next. The metaheuristics outperform the gradient search algorithms on many difficult optimization problems involving discontinuous, noisy, high-dimensional and multimodal objective functions (Goldberg, 1989). When a difficult problem (e.g. NP-hard), is solved by metaheuristics, one has to accept a sub-optimal solution within the allotted CPU time. Naturally, specific task-dependent termination criteria can be introduced such as a threshold on the quality of the solution (Battiti and Tecchiolli, 1994). However, in this study, we wish to compare the efficiency and effectiveness of different metaheuristic methods; thus the employed stopping criterion is the maximum number of objective functional evaluations (also called trial). Three popular metaheuristics are considered in this paper, namely, simulated annealing (SA), genetic algorithm (GA) and reactive tabu search (RTS). The general descriptions of the algorithms are next given.

3.1 Simulated Annealing

The simulated annealing (SA) algorithm is motivated by an analogy to the statistical mechanics of annealing of solids (Kirkpatrick *et al.*, 1983). The system is said to be in thermal equilibrium at a temperature T if the probability of being in state i with energy E_i is governed by a Boltzman distribution. The annealing process leads to this probability law for energy states. A particular configuration for the annealing of physical systems is analogous to the vector of variables, and the energy is analogous to its objective functional value for minimization problem. SA for continuous variables was originally proposed by Vanderbilt and Louie (1984). Karoonsoontawong and Waller (2006) developed the SA based on the work by Lee (1998) to solve the UODTA-based NDP, and also calibrated the SA parameters for the NDP. In this paper, we employ this SA algorithm with the calibrated set of SA parameters (assume that the parameters also work well for all nine problems in Table 1). The detailed SA algorithm can be found in Karoonsoontawong and Waller (2006).

3.2 Genetic Algorithm

Inspired by Darwin's theory of survival of fittest during evolution, the genetic algorithm (GA) is an iterative procedure that maintains a population of candidate solutions (chromosomes) to the objective function (Goldberg, 1989; Grefenstette, 1990). Each chromosome represents a vector of decision variables, and the meaning associated with each chromosome is unknown to the GA. During each generation, the current population is evaluated, and a new population of candidate solutions is formed on the basis of this evaluation. In this paper, we employ the

GA source code by Grefenstette (1990) to solve the nine problems. The parameters calibrated in Karoonsoontawong and Waller (2006) for the DTA-based NDP are used in this study.

3.3 Reactive Tabu Search

As a variation of tabu search, the reactive tabu search (RTS) was proposed by Battiti and Tecchiolli (1994). They made an analogy between the evolution of the direct search process and the theory of dynamical systems. Battiti and Tecchiolli (1994) proposed three memory-based mechanisms to tackle three possibilities that could prevent the direct search from finding the global optimum. First, RTS employs the tabu list and tabu tenure to avoid being trapped in local optima. Second, RTS dynamically updates the tabu tenure according to the status of the search, so that the occurrence of cycles can be eliminated. Third, RTS employs the diversification strategy to escape from chaotic attractors (confinements), which are limited regions of the solution space. The avoidance of cycles and confinements assures that the available time is spent in an efficient exploration of the search space. Karoonsoontawong and Waller (2009) proposed and compared three variations of the modified RTS (called RTS-MT0, RTS-MT1 and RTS-MT2) based on three different neighborhood definitions to solve the DTA-based SSD problem. Karoonsoontawong and Waller (2008) employed the three RTS variations to solve the DTA-based NDP problem. In both studies, RTS-MT2 outperforms RTS-MT0 and RTS-MT1. In this paper, we, thus, employ the RTS-MT2, and refer to it as RTS. The detailed algorithm of RTS-MT2 can be found in Karoonsoontawong and Waller (2009). Note that RTS does not have parameters for calibration.

4. MODIFICATIONS OF THREE GLOBAL OPTIMIZATION TECHNIQUES

The major modifications to the three global optimization algorithms for the nine problems include the solution representation, the associated encoding and decoding procedures, and the functional evaluation.

4.1 Solution Representation, Encoding Procedure and Decoding Procedure

The decision variables for NDP-SSD are simply the combination of the NDP variables and the SSD variables. Thus, we discuss the solution representation of only NDP-SSD. For NDP, Karoonsoontawong and Waller (2006) indicated that the available budgets are always exhausted for the optimal solutions based on the analytical results; thus, the decision variables of NDP can be seen as fractional values of the total available budget (i.e., b_c can be represented as $fb_c \cdot B$, where fb_c is the fraction of B assigned to cell c). Since all cells within a link should be assigned an equal amount of budget to avoid a bottleneck, we consider a variable associated with a link rather than a cell. Formally, we define the variables fbp_i as the fraction of TAB assigned to each cell of link i, the variables fbp'_i as the fraction of B assigned to link i. As such, $fbp_i = fbp'_i / N\_fbp_i$, where N_fbp_i is the number of cells on link i.

For SSD, the SSD decision variables are cycle lengths, green splits, phase sequences and time offsets of pre-specified intersections in the network. We assume that the possible signal phases, possible phase sequencing, the feasible range of cycle length, and the min green times are exogenously determined. Note that the time offset is the time difference between the beginning of the first signal phase 1 and the beginning of simulation period. Karoonsoontawong and Waller (2009) defined the vector of fractional variables f, whose

elements are bounded between 0 and 1 for three intersection types: $f_{3-leg} = \{f_1, f_2, f_3, f_4, f_5\}$ for a 3-leg intersection; $f_{4-leg} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$ for a 4-leg intersection; and $f_{5-leg} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$ for a 5-leg intersection. As such, the size of the vector of fractional variables for SSD (*fssd*) depends on the number and types of intersections in the network. Due to the space limit, the fractional variables of the 3-leg, 4-leg and 5-leg intersections and corresponding decoding procedures are referred to Karoonsoontawong and Waller (2009).

To illustrate, let's consider a network with one of each intersection type and 3 pre-specified links for capacity expansion. The vector of fractional variables (*fndp_ssd*) is $fndp\_ssd = \{fbp, fssd\} = \{fbp, f_{3-leg}, f_{4-leg}, f_{5-leg}\}$ as portrayed in Figure 1. In Figure 1, denote by $N_{ndp\_variable}$ the number of fractional variables for NDP; $N_{ssd\_variable}$ as the number of fractional variables for SSD; and $N_{ndp\text{-}ssd\_variable}$ as the number of fractional variables for NDP-SSD.

4.1.1 Solution Representation, Encoding Procedure, and Decoding Procedure for SA

The solution representation for SA is encoded as the vector of real variables *fr*={*fr_ndp*, *fr_ssd*}. *fr* corresponds to *fndp_ssd*; *fr_ndp* and *fr_ssd* correspond to NDP and SSD fractional variables (fbp' and *fssd*), respectively. As such, the dimension of the vector *fr* is equal to $N_{ndp\text{-}ssd\_variable} = N_{ndp\_variable} + N_{ssd\_variable}$. The decoding procedure translates the vectors *fr_ndp* and *fr_ssd* to the vectors *fbp* and *fssd* from Equations (2) and (3), respectively.

$$fbp_i = \frac{fbp_i'}{N\_fbp_i} = \frac{fr_i}{\sum_{j=1}^{N_{ndp\_variable}} (fr_j)} \cdot \frac{1}{N\_fbp_i} \qquad \forall i = 1,\ldots,N_{ndp\_variable} \tag{2}$$

$$fssd_i = fr_i - \lfloor fr_i \rfloor \qquad \forall i = 1,\ldots,N_{ssd\_variable} \tag{3}$$

4.1.2 Solution Representation, Encoding and Decoding Procedures for GA and RTS

The encoding procedure encodes the vector of fractional variables $fndp\_ssd = \{fbp', fssd\}$ into a binary string structure. The lower bound, upper bound and required precision of each decision variable must be specified. All decision variables (genes) have the same lower bound ($fr_i^{\min} = 0$) and upper bound ($fr_i^{\max} = 1$). Given the required precision (*prec* after decimal point), the required bits (*m*) for each decision variable is determined from Equation (4) (Goldberg, 1989):

$$2^{m-1} < \left(\left(fr_i^{\max} - fr_i^{\min}\right) \cdot 10^{prec} + 1\right) \le 2^m \tag{4}$$

We consider *prec*=2 is sufficient, so *m*=7 for our implementation. Thus, the total bits required to represent a solution (i.e. the length of a chromosome) are $m \cdot N_{variable}$. For NDP, each fractional variable is associated with each link to be assigned budget in the network. For SSD, the fractional variables are associated with the cycle lengths, green times, time offsets and phase sequences of the signalized intersections (see Karoonsoontawong and Waller, 2009).

For NDP, the decoding procedure translates a binary string to the vector of fractional variables *fr*, since there are the boundary constraints imposed on *fr* (i.e.

$0 = fr_i^{\min} \le fr_i \le fr_i^{\max} = 1 \quad \forall i = 1,2,...,N_{ndp\_variable}$). The decoding must transform *fr* to *fbp′* such that $\sum_{i=1}^{N_{ndp\_variable}} fbp'_i = 1$ by Equation (2). For SSD, the decoding procedure that translates a binary string solution to the vector of fractional variables, *fssd*, is Equation (5):

$$fssd_i = fr_i = \frac{real_i}{2^m - 1} \quad \forall i = 1,2,...,N_{ssd\_variable}. \tag{5}$$

where $real_i$ is a real number corresponding to the binary string associated with the ith variable.

4.2 Objective Functional Evaluation

This study employs the UODTA module in the Visual Interactive System for Transport Algorithms (VISTA) (Ziliaskopoulos and Waller, 2000) to evaluate different solutions for larger-size problems. The UODTA module in VISTA is a departure-time-based version of the simulation-based UODTA approach using a mesoscopic simulator based on an extension of the cell transmission model, to propagate traffic and satisfy capacity constraints. The DTA module iteratively employs the time-dependent shortest path algorithm to generate vehicle paths, and the inner approximation dynamic user equilibrium (IADUE) algorithm (Chang, 2004) for equilibration. The implicit assumption is that drivers have perfect information and can divert to alternate paths if it reduces travel time.

The objective of deterministic problems is to minimize the TSTT, whereas that of stochastic problem is to minimize the expected TSTT. The objective of robust problems is to minimize the expected robust objective. Specifically, for deterministic problems (DET-NDP, DET-SSD and DET-NDP-SSD), the UO DTA module in VISTA is called to evaluate the current solution *fr*, and the objective function value *z*(*fr*) is set to the TSTT:

$$z(fr) = TSTT \quad \text{for deterministic problems}$$

With the current solution *fr*, the functional evaluation procedures of both stochastic (SP-NDP, SP-SSD and SP-NDP-SSD) and robust (RO-NDP, RO-SSD and RO-NDP-SSD) problems involve running the UODTA for *N_d* times (where *N_d* is the number of demand scenarios); each UODTA run corresponds to each OD demand scenario generated in the initialization step of the global optimization algorithms. The objective function values for the stochastic and robust problems are:

$$z(fr) = \frac{\sum_{i=1}^{N\_d} TSTT_i}{N\_d} \quad \text{for stochastic problems}$$

$$z(fr) = \frac{\sum_{i=1}^{N\_d} TSTT_i}{N\_d} + \lambda \cdot \frac{\sum_{i=1}^{N\_d} \max^2(0, TSTT_i - TSTT^{TARGET})}{N\_d} \quad \text{for robust problems}$$

where $TSTT_i$ is the TSTT from the ith run of UODTA for $i = 1,\ldots,N\_d$.

5. COMPUTATIONAL EXPERIMENT

The test network is first described, followed by the three parts of the experiment: Part 1- algorithm performance comparison, Part 2- metaheuristic search, and Part 3- solution robustness evaluation under stochastic environment. The outcome of Part 1 is the

identification of the winner on each of the nine problems. These winner algorithms will be employed in Part 2 to search for a "good" or near-global solution in each problem. The outcome of Part 2 is the best solution found so far by the winner algorithm in each of the nine problems. These best solutions are evaluated in Part 3.

5.1 Test Network

The well-known Sioux Fall network in Figure 2a is employed in our experiment. The network is the aggregated network of the city of Sioux Falls, South Dakota, used by many researchers in the literature. All links are 2-lane with the capacity of 1200 vehicles per hours (vph) and the free flow speed of 49.5 miles per hour (mph). Nodes 6, 8, 9, 10, 16, 17 and 18 are signalized intersections. Ten links are candidates for capacity expansion: links (6,8), (8,6), (7,8), (8,7), (9,10), (10,9), (10,16), (16,10), (13,24) and (24,13). The study period is a peak hour with 33 O-D pairs (see Figure 2b). All OD demands are determined from a normal random variable (the mean of 900 vph and standard deviation of 100 vph) multiplied by the factors in Figure 2b. We employ the NDP parameters: B = 1500, χ=160 and ϕ=160. For SSD parameters,

- all red time = 2 seconds per movement phase,
- yellow time = 4 seconds per movement phase,
- Node 10 (a 5-leg intersection)
    - Min green times for movement phases 1, 2, 3, 4 and 5 = 10 seconds
    - Min and Max cycle length = 80 and 250 seconds
- Nodes 8 and 16 (4-leg intersections)
    - Min green times for movement phases 1, 2, 3, 4 = 10 seconds
    - Min and Max cycle length = 64 and 210 seconds
- Nodes 6, 9, 17 and 18 (3-leg intersections)
    - Min green times for movement phases 1, 2, 3 = 10 seconds
    - Min and Max cycle length = 48 and 180 seconds

In all three parts, for the deterministic problems, the three metaheuristic algorithms are performed using the average OD demand matrix, which is obtained by multiplying 900 vph to the factors in Figure 2b. In Part 1, the stochastic and robust problems are solved using a small number of demand realizations (say 3), so that the CPU time for each run is manageable. The small size of demand realizations is employed with the assumption that the relative performances of SA, GA and RTS remain the same regardless of the size of demand realizations. Furthermore, the same set of demand realizations is employed to perform all iterations of SA, GA and RTS, so that the results are comparable. Note that the OD demand realizations are generated by the Monte Carlo simulation in Parts 1, 2 and 3. The generated OD demand samples are independent and identically distributed from the distribution of OD demands. In Part 2, the stochastic and robust problems are solved using a larger number (i.e. 20) of demand realizations (see Figure 2c). The same set of demand realizations is employed in all nine problems using the corresponding winner algorithm. In Part 3, the best solution in each problem is evaluated under stochastic environment, using a new set of demand realizations with the sample size 30 (see Figure 2d). We employ the new set of demand realizations, so that any possible evaluation bias can be eliminated.

5.2 Experimental Results

5.2.1 Part 1: Algorithm Performance Comparison

The performances of metaheuristics are compared according to solution quality and convergence speed. The CPU time is dominated by the functional evaluation time, so the CPU times of all metaheuristics for the same total trials are approximately the same. Especially for the stochastic and robust problems, a trial involves performing multiple UODTA runs corresponding to all OD demand realizations. The common stopping criterion of the three metaheuristics is the total trials. We employ 250 trials for DET-SSD and DET-NDP; 500 trials for DET-NDP-SSD; 150 trials for SP-SSD, SP-NDP, RO-SSD and RO-NDP; and 300 trials for SP-NDP-SSD and RO-NDP-SSD. Since the employed total trials are relatively small, especially for the stochastic and robust problems, the better solution quality also implies the better convergence speed. Thus, in this particular study, these two criteria may be considered the same, and we only discuss the comparison in terms of solution quality as this is of higher interest. Due to the space limit, we show the convergence characteristics of the three algorithms for only RO-NDP-SSD in Figure 2e. We rank the three metaheuristics with respect to the solution quality for the deterministic, stochastic and robust problems on the test network in Figure 2f. Apparently, there is not the single winner that outperforms the others across all nine problems on the test network. For the deterministic problems, RTS appears best; GA the second best; and SA the third best. For the stochastic problems, RTS, GA and SA appear best, second best and third best, respectively. For the robust problems, there is not a clear order.

5.2.2 Part 2: Metaheuristic Search for Best Solutions

Based on the results in Figure 2f, the identified best metaheuristic algorithm for each problem is employed to find a best solution. Since we would like to evaluate and compare the deterministic, stochastic and robust solutions, the same number of total trials is employed. Specifically, we employ 75 total trials for the SSD and NDP problems and 150 total trials for the NDP-SSD problems. For the three SSD problems (DET-SSD, SP-SSD and RO-SSD), all metaheuristics optimize the SSD variables with the original link capacity to obtain the best signal settings. Then, the three best signal settings obtained from the deterministic, stochastic and robust SSD are fixed for running the respective deterministic, stochastic and robust NDP. This allows us to compare the sequential SSD and NDP against the combined NDP-SSD approach. For a fair comparison, the number of trials (i.e. the number of functional evaluations) for NDP-SSD is two times as many as that for SSD and NDP. That is, given the same total trials, the sequential approach is compared with the combined approach. For the robust problems, we set $TSTT^{TARGET}$ equals to the minimal $E[TSTT]$ obtained from the sequential stochastic approach (SP-SSD & SP-NDP) and the combined approach (SP-NDP-SSD). It is noted that the desirable λ value in robust problems depends on the preference of transportation planners, and the sensitivity analysis of λ value (see Karoonsoontawong and Waller, 2007) can be utilized to help elicit the decision makers' preference information. We arbitrarily use λ=1. It is also noted that we tradeoff the OD demand sample size (the larger sample size implies the better quality of solution) for the reasonable computational time. According to the rule of thumb in the Monte Carlo-based stochastic optimization, we use the sample size of 20. The larger sample size is indeed desirable; however, it costs too long computational time, given that we have to do multiple metaheuristic runs. Since DET-NDP, SP-NDP and RO-NDP employ different traffic signal settings, these three may not be comparable. As such, we employ the original signal settings in Figure 2g for NDP problems, and we denote the three NDP problems by NDP-D (where D stands for the default signal settings). Specifically, three additional NDP problems are solved (DET-NDP-D, SP-NDP-D and RO-NDP-D). The detailed best solutions found by the metaheuristics for all twelve problems (i.e., DET/SP/RO of SSD/NDP/NDP-D/NDP-SSD) are given in Figure 2h. All

deterministic, stochastic and robust SSD and NDP-SSD solutions employ different time offsets and different cycle lengths for intersections on both networks. This means we benefit from relaxing the common cycle length assumption and from the traffic signal coordination such that the great number of vehicles is progressed through the network.

5.2.3 Part 3: Solution Robustness Evaluation under Stochastic Environment

To evaluate and compare all solutions from the twelve problems on the test network, we use a separate set of 30 OD demand realizations that are independent and identically distributed from the distribution of OD demands. Evaluating a solution involves running a DTA corresponding to an O-D demand realization. After all 30 DTA runs, we compute the expected value (E[.]), standard deviation (SD[.]), error ($Error$[.]) and two-sided 95% confidence interval (95%CI[.]) of *Robust_Obj(λ=1)*, *TSTT* and *Risk*. Note that $Error[.] = t_{0.025,29} \cdot SD[.] \big/ \sqrt{30}$; 95%$CI$[.] = E[.] ± $Error$[.]; and t(0.025, 29)=2.045. The detailed evaluation results of the original networks (do-nothing) and the solutions for all twelve problems are given in Figure 2i. Figures 3-5 depict the 95% CIs of *E[Robust_Obj(λ=1)]*, *E[TSTT]* and *E[Risk]* for the original network and all twelve problems (DET/SSD/RO of SSD/NDP/NDP-D/NDP-SSD). Figure 6 shows *V[TSTT]* for the original network and the twelve problems. Since the expected risk in the objective is not the variance of TSTT, the pattern of *V[TSTT]* is not consistent with that of *E[Risk]*. However, the RO-NDP-SSD solution with the least *E[Risk]* also has the least V[TSTT]. The comparisons of point estimates are conducted in three dimensions. First, we compare among the deterministic, stochastic and robust solutions on the test network; that is, the solutions within the following three cases are compared against each other within the case: case 1 (DET-SSD, SP-SSD, RO-SSD), case 2 (DET-NDP-D, SP-NDP-D, RO-NDP-D) and case 3 (DET-NDP-SSD, SP-NDP-SSD, RO-NDP-SSD). We find that the RO solutions yield the least *E[Robust_Obj(λ=1)]* and *E[Risk]* as expected, whereas the DET solutions perform the worst with the highest *E[Robust_Obj(λ=1)]*, *E[TSTT]* and *E[Risk]*. Since the metaheuristics for robust problems are not designed to minimize the *V[TSTT]*, the solutions of RO problems do not always yield the least *V[TSTT]*. Second, we compare the sequential SSD and NDP approaches against the integrated NDP-SSD approaches; i.e., DET-NDP versus DET-NDP-SSD; SP-NDP versus SP-NDP-SSD; and RO-NDP versus RO-NDP-SSD. In terms of *E[Robust_Obj(λ=1)]*, the integrated approach performs better than the sequential approach except the case of DET-NDP-SSD and DET-NDP. In terms of *E[TSTT]*, the integrated approach performs better than the sequential approach in all cases. In terms of *E[Risk]*, the integrated approach outperforms the sequential approach in most cases. Third, we compare the original network and the twelve problems altogether. Apparently, the RO-NDP-SSD solution outperforms the others on the test network in terms of *E[Robust_Obj(λ=1)]*, *E[TSTT]*, *E[Risk]* and *V[TSTT]*, whereas the sequential robust approach (i.e. RO-SSD & RO-NDP) performs the second best. The solutions from the twelve problems perform better than the original network in terms of *E[Robust_Obj(λ=1)]*, *E[TSTT]* and *E[Risk]* except DET-NDP-D. This shows that the deterministic approach can yield worse solutions than the do-nothing case when evaluated under stochastic condition. Thus, the integrated robust approach (RO-NDP-SSD) should be desirable whenever possible. Unlike the analytical result in Karoonsoontawong and Waller (2008), the sequential robust approach (RO-SSD & RO-NDP) performs the second best. This implies that for larger-size multi-destination networks where the analytical approach is not applicable, the integrated robust approach is still the most desirable; otherwise, the next best approach is the sequential robust approach. It is noted that we have made rather weak comparisons that one solution is better than another in expected

values. Apparently, one can observe the overlapping 95% confidence intervals of *E[TSTT]*, *E[Risk]* and *E[Robust_Obj(1)]* of many solutions. We can make a stronger comparison via paired *t* tests. We perform 36 paired *t* tests to test the null hypotheses that the RO-NDP-SSD solution outperforms the other 12 solutions in terms of the three measures (*E[TSTT]*, *E[Risk]* and *E[Robust_Obj(1)]*). The minimal *p*-value from the 36 paired *t* tests is 0.7950, implying that on the test network we fail to reject the null hypothesis when the level of significance $\alpha <$ 0.7950.

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

The computational experiment on the well-known modified Sioux Fall network is composed of three parts. In Part 1, SA, GA and RTS are compared on the test network for the nine problems in order to identify the best algorithms. In Part 2, the best algorithm for each problem is employed to find a best solution given the total number of functional evaluations (i.e. total trials). In Part 3, the best solution for each problem is evaluated using a separate set of OD demand realizations in order to prevent possible bias. The evaluation measures are *E[Robust_Obj(λ)]*, *E[TSTT]* and *E[Risk]*. We find that the RO solutions yield the best robustness and lowest risk as expected, whereas the DET solutions perform the worst. In terms of robustness, the integrated approach performs better than the sequential approach except the case of DET-NDP-SSD and DET-NDP. In terms of *E[TSTT]*, the integrated approach performs better than the sequential approach in all cases. In terms of *E[Risk]*, the integrated approach outperforms the sequential approach in most cases. The RO-NDP-SSD solution outperforms the others on the test network in terms of robustness, *E[TSTT]*, *E[Risk]* and *V[TSTT]*, whereas the sequential robust approach (i.e. RO-SSD & RO-NDP) performs the second best. Thus, the integrated robust approach is the most desirable; otherwise, the next best approach is the sequential robust approach. The solutions from the twelve problems perform better than the original network except DET-NDP-D, implying that the deterministic approach can yield worse solutions than the do-nothing case when evaluated under stochastic condition.

Note that the experimental results of Part 1 imply that there is not a single winner algorithm across the nine problems; therefore, in order to achieve best solutions, one should employ a best algorithm for a particular problem. Although the findings may not necessarily be generalized, they provide interesting and insightful information. For deterministic/stochastic/robust NDP, SSD and NDP-SSD, either RTS or GA should be employed. The experimental results of Part 2 and Part 3 reiterate the usefulness of the integrated approach over the sequential approach and demonstrate the merit of the robust integrated approach. In reality, the probability distributions of OD demands have to be derived from the OD survey, and the best algorithm can be employed to determine the best network capacity expansion policy from NDP, to determine the best signal timing plans from SSD, or the combined results from NDP-SSD.

For the future research, we will explore alternative methods that can improve the performance of Monte Carlo simulation in generating the OD demand realizations, such as the Quasi-Monte Carlo simulation and the variance reduction techniques. The better representative sample from the probability distributions of OD demands will allow the global optimization to obtain a better solution. Also, in the evaluation of the solution, we can obtain tighter confidence intervals of various measures, given the same sample size of demand realizations.

REFERENCES

Bard, J. F. (1998) **Practical Bilevel Optimization Algorithms and Applications**. Kluwer Academic Publishers.

Battiti, R. and G. Tecchiolli. (1994) The Reactive Tabu Search. **ORSA Journal on Computing 6**, No.2, p.126-140.

Chang, E. J. (2004) **Time-varying intermodal person trip assignment**, Ph.D. Dissertation, Northwestern University.

Daganzo, C. F. (1994) The Cell Transmission Model: A Simple Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory. **Transportation Research B 28**, No. 4, pp. 269-287.

Glover, F. (1989) Tabu Search - Part I. **ORSA Journal on Computing 1**(3), pp 190-206.

Goldberg, D.E. (1989) **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley, Reading, MA.

Grefenstette, J.J. (1990) **A User's Guide to GENESIS Version 5.0**. October, http://www.aic.nrl.navy.mil/galist/src/

Karoonsoontawong, A. and S.T. Waller. (2008) Integrated Network Capacity Expansion and Traffic Signal Optimization Problem: Robust Bi-level Dynamic Formulation. **Networks and Spatial Economics, DOI /10.1007s-9071-008-11067x**.

Karoonsoontawong, A. (2006) **Robustness Approach to the Integrated Network Design Problem, Signal Optimization and Dynamic Traffic Assignment Problem**. Ph.D. Dissertation, The University of Texas at Austin.

Karoonsoontawong, A., S.T. Waller (2006) Dynamic Continuous Network Design Problem: Linear Bi-level Programming and Metaheuristics. **Journal of the Transportation Research Board 1964**, 104-117.

Karoonsoontawong, A., S.T. Waller (2007) Robust Dynamic Continuous Network Design Problem. **Journal of the Transportation Research Board 2029**, 2007, pp.58-71.

Karoonsoontawong, A. and S.T. Waller (2009) An Application of Reactive Tabu Search for Combined Dynamic User Equilibrium and Traffic Signal Optimization Problem. **Forthcoming in Journal of the Transportation Research Board**.

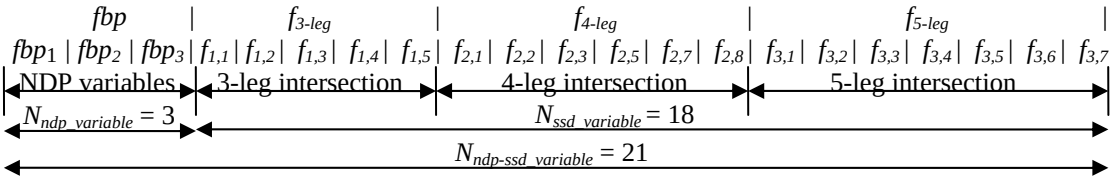
Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi. (1983) Optimization by Simulated Annealing. **Science 220**, 671-680.

Lee, C. (1998) **Combined traffic signal control and traffic assignment: algorithms, implementation and numerical results**. Ph.D. Dissertation, The University of Texas at Austin, Austin, Texas.

Ukkusuri, S.V. and S.T. Waller (2007) Linear Programming Models for the User and System Optimal Dynamic Network Design Problem: Formulations, Comparisons and Extensions, **Networks and Spatial Economics, DOI 10.1007/s11067-007-9019-6**.

Vanderbilt, D. and S. G. Louie. (1984) A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables. **Journal of Computational Physics 56**, 259-271.

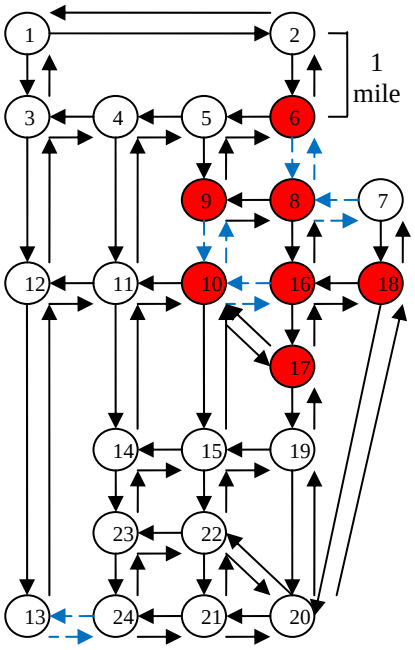
Ziliaskopoulos, A.K. and S.T. Waller. (2000) An Internet-based geographic information system that integrates data, models and users for transportation applications. **Transportation Research Part C 8**, 427-444.



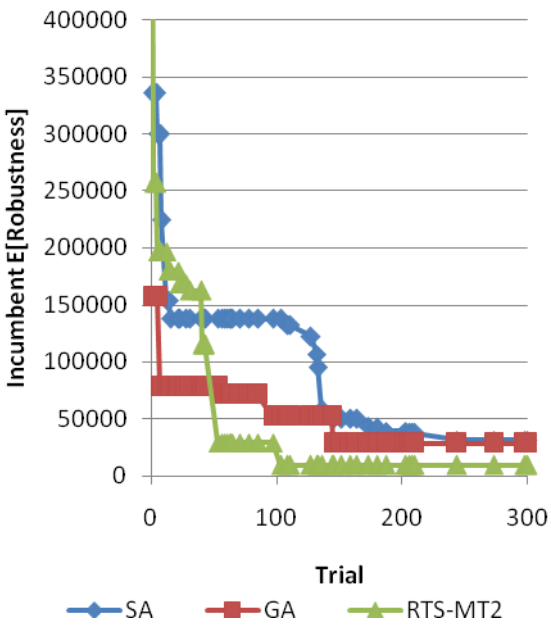
Note: For 4-leg intersection, we assume non-overlapping movement; i.e. $f_{2,3}=f_{2,4}$ and $f_{2,5}=f_{2,6}$.

Figure 1 Fractional variables of example network with 3 pre-specified links for improvement and one of each intersection type for NDP-SSD problems

a) Modified Sioux Falls network (Nodes 6, 8, 9, 10, 16, 17 and 18 are signalized intersections; and ten dashed links are candidates for capacity expansion.)



e) Algorithm performance comparison for RO-NDP-SSD



b) Table F.2. Peak-hour O-D trip matrix for the modified Sioux Fall network with 33 OD pairs (page 244 in Karoonsoontawong, 2006)
c) Table F.5. 20 OD demand realizations for metaheuristic search for the modified Sioux Fall network with 33 OD Pairs (page 246 in Karoonsoontawong, 2006)
d) Table F.6. 30 OD demand realizations for evaluation for the modified Sioux Fall network with 33 OD pairs (pages 247-248 in Karoonsoontawong, 2006)
g) Table 6.10b. Original traffic signal settings (page 198 in Karoonsoontawong, 2006)
h) Tables G5-G8. in Appendix G: Detailed solutions from metaheuristics (pages 251-253 in Karoonsoontawong, 2006)
i) Tables H1 and H3 in Appendix H: Robustness evaluation results of deterministic, stochastic and robust solutions from metaheuristics (Karoonsoontawong, 2006)

f) Results of algorithm performance comparisons based on the solution quality

	Deterministic			Stochastic			Robust		
Rank	SSD	NDP	NDP-SSD	SSD	NDP	NDP-SSD	SSD	NDP	NDP-SSD
1	RTS	RTS	GA	RTS	RTS	RTS	RTS	GA	RTS
2	GA	GA	RTS	GA	GA	SA	GA	SA	GA
3	SA	SA	SA	SA	SA	GA	SA	RTS	SA

Figure 2. Test network and algorithm performance comparison results

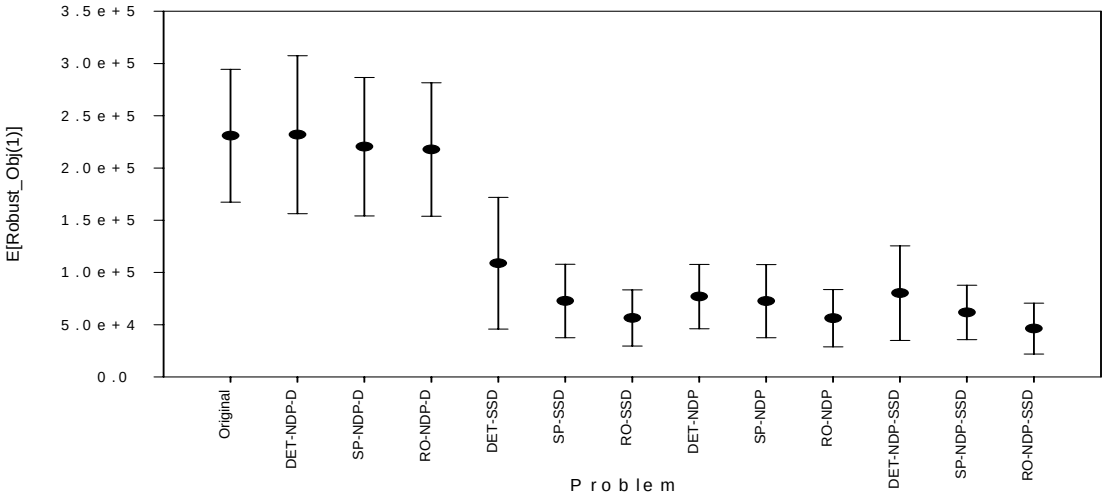


Figure 3. 95% confidence intervals of *E[Robust_Obj(λ=1)]*

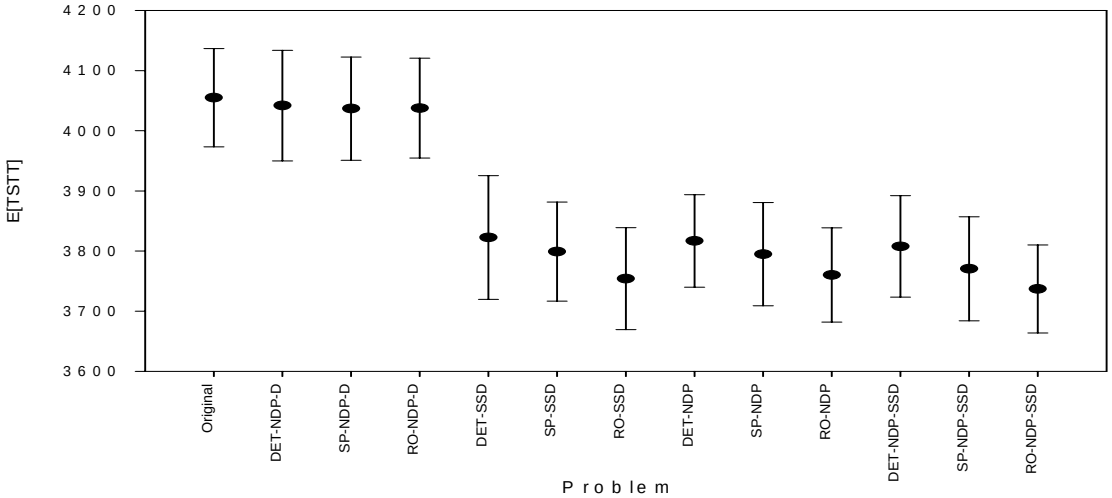


Figure 4. 95% confidence intervals of *E[TSTT]*

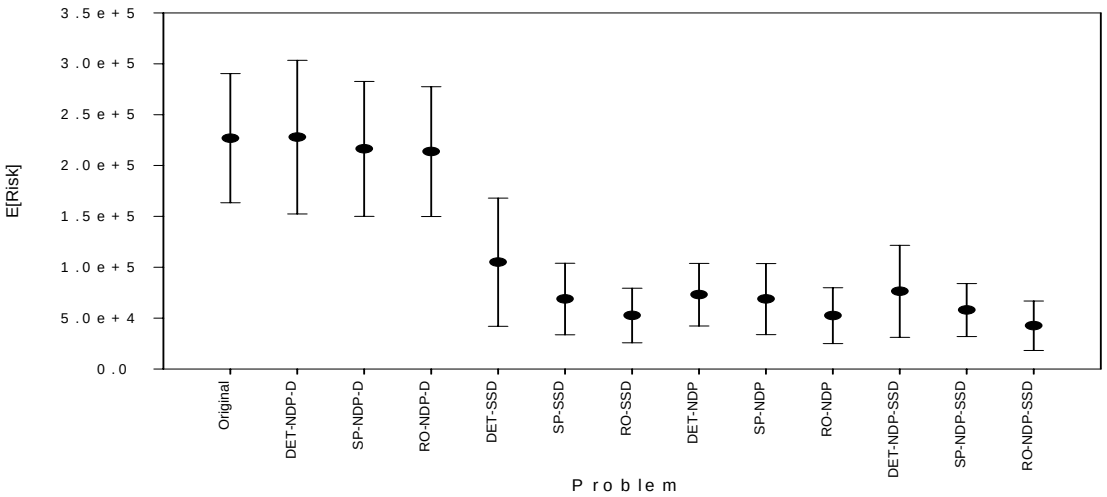


Figure 5. 95% confidence intervals of *E[Risk]*

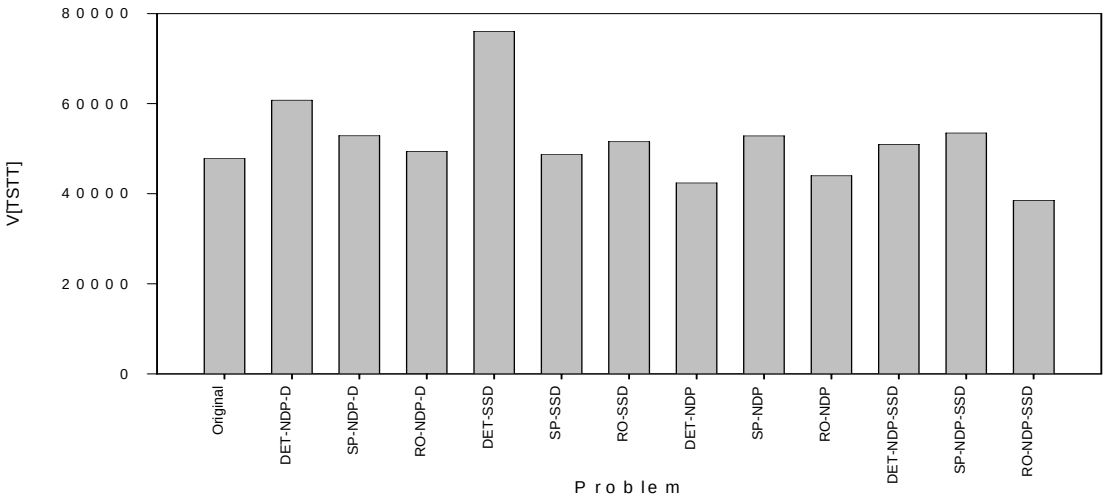


Figure 6. Variance of TSTT

Table 1. Abbreviations and descriptions of nine problems

Math Formulation	Problem Abbreviation	Problem Description
RO-BLPNDP-SSD	RO-NDP-SSD	Robust Bi-Level Integrated NDP, SSD and UODTA Model
SLP2-BLPNDP-SSD	SP-NDP-SSD	Two-Stage Stochastic Bi-Level Integrated NDP, SSD and UODTA Model
BLPNDP-SSD	DET-NDP-SSD	Deterministic Bi-Level Integrated NDP, SSD and UODTA Model
RO-BLPSSD	RO-SSD	Robust Bi-Level Combined SSD and UODTA Model
SLP2-BLPSSD	SP-SSD	Two-Stage Stochastic Bi-Level Combined SSD and UODTA Model
BLPSSD	DETSSD	Deterministic Bi-Level Combined SSD and UODTA Model
RO-BLPNDP	RO-NDP	Robust Bi-Level Combined NDP and UODTA Model
SLP2-BLPNDP	SP-NDP	Two-Stage Stochastic Bi-Level Combined NDP and UODTA Model
BLPNDP	DET-NDP	Deterministic Bi-Level Combined NDP and UODTA Model