An Algorithmic Framework for the Scheduling of Construction Projects based on Ant Colony Optimization and Expert Knowledge

Zhitao Xiong, David Rey, Vinayak V. Dixit and S. Travis Waller

Abstract— **The conduction of construction projects in a road network can result in a reduction of the capacity of the lanes that are under construction or maintenance. In order to mitigate the impact of construction projects, it is critical to find the best schedule in such a way that the effects of road capacity reductions, e.g. traffic delay caused by their presence, is minimized. This article propose a novel formulation for the construction projects scheduling problem using bi-level programming. A solution method is then introduced to solve this challenging scheduling problem with a focus on reducing the number of calculations in the optimization. This is achieved by integrating expert knowledge, which can be used to characterize a good schedule, in the algorithmic framework in charge of determining a near-optimal schedule. Such heuristic information can come from relevant experts or practitioners, or from the literature in this field. A solution algorithm named CoANT based on ant colony optimization is presented and implemented on realistic transportation networks. Our results show that CoANT works fast and is able to provide competitive schedules. As an extensible and modular framework, CoANT can be used by relevant transportation agencies as a decision-aid tool for the coordination of construction projects in road networks.**

I. Introduction

Lane closures that are caused by construction projects, leading to road capacity reduction, can cause traffic congestion and delay. Although lane closures cannot be avoided because of relevant construction works, e.g., bridge maintenance, pavement construction, their impacts, e.g., resulted traffic delay and agencies' cost, can be minimised and evaluated beforehand. In this paper, we present a novel algorithm, named Construction prOject scheduling based on ANT colony optimization (CoANT), which utilizes expert (human) knowledge to schedule the coordination of construction projects in urban transportation network so as to minimize the Total Traffic Delay (TTD) induced by lane closures.

The approaches proposed so far to solve the Construction Projects Scheduling Problem (CPSP) can be divided into two groups: estimation-based and optimization-based approaches. Estimation-based approaches rely mainly on human expertise to plan schedules. For instance, some non-proprietary tools used by relevant traffic agencies estimate the impact of construction projects before and after the projects are carried out [1]. However, in doing so, they fail to consider the direct impact of work zones on traffic propagation, thus potentially generating excessive costs from a user's perspective.

In contrast, optimization-based approaches utilize mathematical models and relevant solution algorithms to find efficient schedules. Given the twofold nature of the CPSP—which combines system-oriented decisions and user behaviour—most of the models developed adopt a bi-level programming formulation where the lower level is concerned with assigning traffic volumes in the network and the upper level is focused on minimizing the impact of the construction projects onto operating costs. In general, researchers have put a focus on proposing either a detailed model for schedule evaluation or an algorithm to find feasible schedules. For instance, a mixed-integer bi-level model has been proposed to minimize long-term maintenance costs and the Total System Travel Time (TSTT) [2]. This approach also takes into account traffic dynamics through a traffic flow model, the Cell Transmission Model (CTM), to represent the propagation of traffic in the network [3].

Due to the complexity of the CPSP, the solution algorithms proposed to solve the aforementioned bi-level program rely mainly on metaheuristic frameworks. For instance, Ant Colony Optimization (ACO) was used to minimize the TTD induced by the construction projects [4]. Tabu search was implemented to select the schedule with the least TTD [5]. The exact cost of each schedule can be difficult to obtain and to compare against each other as the number of candidate schedules can be large and/or the evaluation of the cost function can be computationally expensive. In particular, the lower level of the bi-level representation of the CPSP is generally formulated as a User Equilibrium (UE) problem, which can be represented by a convex mathematical program which resolution may be resource-intensive for large networks [6]. As a result, approximation methods have been proposed to evaluate the cost of each schedule where the traffic assignment costs are approximated using link distance and an all-or-nothing traffic assignment under updated link travel time [1].

Although most of the solution algorithms proposed for the CPSP claim that expert information is used within the search procedures, their scheduling heuristics are determined in a greedy manner. However, those greedy search-based methodologies can be used to guarantee the uniqueness of the scheduling solution although the searching and verification

Z. Xiong, PhD, is with the School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW, 2052, Australia. Corresponding author, z.xiong@unsw.edu.au

D. Rey, PhD, is with the School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW, 2052, Australia.

V.V. Dixit, PhD, is a Senior Lecturer at the School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW, 2052, Australia.

S.T. Waller, PhD, is a Professor at the School of Civil and Environmental Engineering and NICTA, UNSW Australia, Sydney, NSW, 2052, Australia.

can be time-consuming. Therefore, some other measures need to be taken to reduce the time spent in generating the results. For instance, [1] used link distance and an all-or-nothing traffic assignment under updated link travel time to get approximated traffic assignment costs, which is to reduce the calculation load in the phase of verification. No extra measures utilizing expert knowledge, which can indicate the necessary conditions of being a good schedule, have been proposed in the search phase.

In general, in order to develop a comprehensive framework for the CPSP, the following factors and inputs should be taken into consideration:

1) For the road network where the construction projects will be carried out:
   a) network data such as nodes and links;
   b) demand data such as OD matrices, time-dependent trip tables, etc.;
2) For each construction project:
   a) location, which is based on network data, i.e., construction projects are represented as links that they are being operated on;
   b) duration, which indicates how long this construction project will take, the unit can be hours or days if it is a short term (construction) project and months if it is a long term one;
   c) earliest start time, which indicates the earliest start time of the project; it can be represented by a specific date or time, e.g., 1/2015 or just a number, indicating the hour, day or month index within a time window, e.g., 5, indicating the 5th month, hour or year within n months, hours or years ($n \geqslant 5$)
   d) latest finish time, which indicates the latest finish time of the project;
   e) closed lanes, which indicate which lanes will be closed, this can be ignored in some cases and be represented simply as a capacity reduction rate;
   f) capacity reduction rate, which indicates the capacity reduction of the link where the project is located at;

When more factors have to be considered, the models proposed to solve the CPSP can vary and have different emphasis. In order to reduce the time that is spent on solving the problem, three approaches can be used:

1) minimize the number of schedules that are needed to be evaluated by providing a search direction for the upper level, i.e., provide heuristic information that can be used to characterize a *good* schedule. For instance, a good schedule should avoid parallel construction sites [7]. This information can be acquired from relevant experts or practitioners;
2) approximate the evaluation of the cost functions of the model so that each schedule can be evaluated in a reasonable time;
3) transfer the cost evaluation of the entire road network into the area of interest, i.e., reducing the number of nodes and links in a road network where the projects will be carried out, so the subnetwork analysis[8] can be a candidate by reducing the size of the network being used. This is to construct a network to include only, e.g., an area that need to examined due to construction projects.

In this paper, we focus on the first approach and develop a metaheuristic framework that integrates expert information. In particular, we use ACO and propose a tailored solution algorithm herein referred to as CoANT (Construction prOject scheduling based on ANT colony optimization). Compared to the existing methodologies, Algorithm CoANT seek to nominate a candidate schedule with optimized start times based on expert information, which is used to identify good schedules.

This paper is organised as follows: Section II provides a formal description of the CPSP and Section III present the CoANT algorithm used to solve the CPSP. Section IV is focused on the implementation of Algorithm CoANT on a realistic urban transportation network and discusses the results obtained. Section V summarizes the findings of this research and underlines promising extensions.

II. Problem Statement

Let $G=(N,A)$ be a network where N is the set of nodes and A is the set of arcs. Let x_a be the flow on arc a and let $t_a(x_a)$ be the delay function on arc a. Let Q_{ij} be the travel demand for every pair of nodes $i,j \in N$ and let Π_{ij} be the set of paths between this pair of nodes. The UE formulation for the static traffic assignment problem can be represented as follows [6]:

Model 1 (UE):

$$\min \sum_{a\in A} \int_0^{x_a} t_a(w)dw \tag{1}$$

subject to:

$$\sum_{k\in\Pi_{ij}} f_k^{ij} = Q_{ij} \quad \forall i,j \in N \tag{2}$$

$$x_a = \sum_{i\in N}\sum_{j\in N}\sum_{k\in\Pi_{ij}} f_k^{ij}\delta_{a,k}^{ij} \quad \forall a \in A \tag{3}$$

$$f_k^{ij} \geq 0 \quad \forall i,j \in N, \forall k \in \Pi_{ij} \tag{4}$$

$$x_a \geq 0 \quad \forall \in A \tag{5}$$

where f_k^{ij} represents the flow on path $k \in \Pi_{ij}$ and $\delta_{a,k}^{ij}$ is an indicator which takes value 1 if arc a belongs to path k between the Origin-Destination (OD) pair i,j and 0 otherwise. For the reminder of this paper, we assume that the arcs delay functions can be represented using the BPR functions, that is:

$$t_a(x_a) = t_a^0\left(1+\alpha\left(\frac{x_a}{C_a}\right)^{\beta}\right), \quad \forall a \in A \tag{6}$$

where C_a is the capacity of arc a, t_a^0 is the free-flow travel time on a and α and β are design parameters.

Let P be a set of construction projects to be carried out on specific arcs of the network G. Namely, let $R_{a,p} \in [0,1]$ be capacity reduction rate of project $p \in P$ on link $a \in A$. $R_{a,p} = 1$ means that the capacity of arc a is entirely consumed by project p. In contrast, $R_{a,p} = 0$ means that the capacity of arc a is not affected by project p. The completion of each construction project requires a given amount of time units $D_p > 0$ and all the projects should be carried out within $|T|$ time units, where T is set of time periods. The objective of the workzone scheduling problem is to determine the optimal starting times of the construction project such that the total delay induced by the arc capacity reduction is minimized. Let s_p be the starting time of project $p \in P$, S_p be the earliest starting time of project $p \in P$ ($0 \leq S_P \leq |T| - D_p$), F_p be the latest finish time of project $p \in P$ ($D_p \leq F_P \leq |T|$), the scheduling constraints can be represented as:

$$S_p \leq s_p \leq \min\{|T|, F_p\} - D_p \quad \forall p \in P \tag{7}$$

To represent the impact of each project onto the capacity of the arcs of the network, we introduce the decision variables y_p^t defined as:

$$y_p^t \equiv \begin{cases} 1 & \text{if } 0 \leq t - s_p \leq D_p \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Variables y_p^t take value one if project p is scheduled at the time period $t \in T$. They can be represented using the constraints:

$$y_p^t(t - s_p) \leq D_p \quad \forall p \in P, \forall t \in T \tag{9}$$
$$y_p^t(t - s_p) \geq 0 \quad \forall p \in P, \forall t \in T \tag{10}$$

To ensure that every project $p \in P$ is active during exactly D_p time periods, we impose the constraint:

$$\sum_{t \in T} y_p^t = D_p \quad \forall p \in P \tag{11}$$

Using the above notation, we can express the capacity of arc a at time period t using the formula:

$$C_a^t = C_a \left(1 - \sum_{p \in P} y_p^t R_{a,p}\right)^+ \tag{12}$$

where $(X)^+ \equiv \max\{X, 0\}$. Substituting C_a by C_a^t in the arc delay function $t_a(x_a)$ and solving the associated UE traffic assignment problem for all time periods, provides a set of flow patterns $x_t = [x_a^t], \forall t \in T$ that takes into consideration the impact of the construction projects onto the arc capacity. To measure the efficiency of the system we use the TSTT which is defined as:

$$TSTT(x) = \sum_{a \in A} x_a t_a(x_a) \tag{13}$$

Therefore to minimize the TTD induced by the schedule of the construction projects over all time periods, we seek to minimize the sum of the TSTTs resulting from the schedule of the projects. Let $\mathbf{s} = [s_p]$ be the array of schedules, the CPSP problem can be represented by the following bi-level program.

Model 2 (Bi-level program for the CPSP):

$$\min_{\mathbf{s}} \sum_{t \in T} TSTT(x_t) \tag{14}$$

subject to:

$$x_t = \arg\min \left\{ \sum_{a \in A} \int_0^{x_a^t} t_a^0 \left(1 + \alpha \left(\frac{w}{C_a^t}\right)^\beta\right) dw : \sum_{k \in \Pi_{ij}} f_k^{ij} = Q_{ij} \right\} \quad \forall t \in T \tag{15}$$

$$C_a^t = C_a \left(1 - \sum_{p \in P} y_p^t R_{a,p}\right)^+ \quad \forall a \in A, \forall t \in T \tag{16}$$
$$s_p \geq S_p \quad \forall p \in P \tag{17}$$
$$s_p \leq \min\{|T|, F_p\} - D_p \quad \forall p \in P \tag{18}$$
$$y_p^t(t - s_p) \leq D_p \quad \forall p \in P, \forall t \in T \tag{19}$$
$$y_p^t(t - s_p) \geq 0 \quad \forall p \in P, \forall t \in T \tag{20}$$
$$\sum_{t \in T} y_p^t = D_p \quad \forall p \in P \tag{21}$$
$$y_p^t \in \{0,1\} \quad \forall p \in P, \forall t \in T \tag{22}$$
$$s_p \in \mathbb{N} \quad \forall p \in P \tag{23}$$

In the next section, we present the metaheuristic framework developed to solve Model 2.

III. Algorithmic Framework Description

CoANT seek to utilize expert knowledge together with ACO in order to search for a feasible and cost-effective solution for the CPSP, i.e. a good schedule for the coordination of the construction projects.

As illustrated in Fig. 1, Algorithm CoANT has five components, one of which is optional. CoANT first generates a candidate schedule $\hat{\mathbf{s}}$ based on the feasible schedule range, i.e. $[S_p, F_p]$, of each project. For instance, if $P = \{p_0, p_1\}$ and the parameters of the CPSP are:

1) $S_{p_0} = 0$, $S_{p_1} = 1$;
2) $D_{p_0} = 3$, $D_{p_1} = 4$;
3) $F_{p_0} = 3$, $F_{p_1} = 4$;
4) The time horizon is 5 months, i.e., $T = 5$.

p_0 can then start in month 0 or 1 ($s_{p_0} \in \{0,1\}$) and p_1 can only start in month 1 (the 2nd month, $s_{p_1} = 1$). In the initial schedule, all projects start from their earliest start times. Then, the ACO-based scheduler is invoked and an optimised schedule is generated, after which CoANT determines the cost of the current solution — the TTD induced by the initial schedule. Finally, the current schedule is passed to the Final Cost Scheduler and a final cost is obtained. As the scheduler in CoANT does not consider each obtained schedule's cost, the obtained schedule may result in a larger cost compared to the one of the initial schedule. In order to avoid this situation, the scheduler and the cost evaluation are performed multiple times until convergence conditions are met. The schedule with a minimal cost is selected as the final schedule. In the following sections, each component of the CoANT algorithm is detailed.

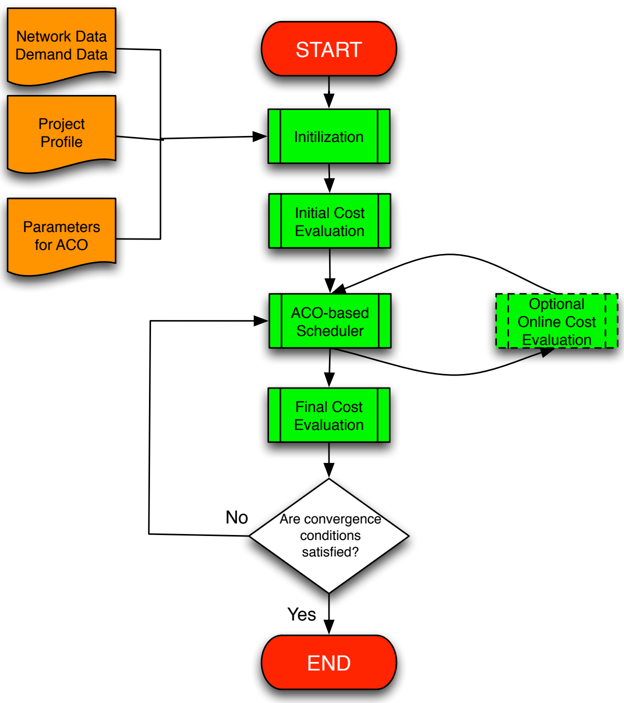


Fig. 1. Workflow of CoANT

1) Initialization:

This component is the first step of CoANT. It is used to initialize the data that will be used in the rest of the procedures. CoANT reads the network data, demand data and parameter values for ACO as input and store them in relevant data structures.

2) Initial Cost Evaluation:

This component is used to evaluate the TTD induced by the initial schedule based on the earliest feasible starting time for each project, i.e. $s_p = S_p, \forall p \in P$. The cost of this schedule is denoted $T(\hat{\mathbf{s}})$. This component is also used to calculate the sum of TSTT over all time periods when no construction project is conducted, this value is denoted T_0. Those two costs can be used to evaluate the final schedule generated by CoANT, indicating the relative impact that an optimised schedule can have. Namely, the delay induced by the initial schedule is $T(\hat{\mathbf{s}}) - T_0$.

3) ACO-based Scheduler:

ACO has been chosen to be the core of this solution algorithm for the CPSP because of its ability to integrate expert information, which can include the necessary conditions of a good schedule.

As first developed by [9], ACO seeks to mimic the food finding process of ants. Ants deposit some chemicals called pheromones on their paths and decide where to proceed based on the pheromones left by their predecessors. The path with the most pheromones is regarded as the best path. Several dialects of ACO exist and put different emphasis on how ants' decisions can be utilised, e.g., if ants can be ranked, then ants with higher ranks will have the right to leave pheromones. In this research, we assume that every ant is able to leave pheromones and share the same privilege.

Pheromones can be used to represent some simplified expert knowledge when choosing a preferred schedule, i.e. a path for carrying out project construction actions. This simplified heuristic information has been identified based on three factors [7]:

1) Importance of a link, represented by $\frac{C_a}{\hat{C}}$ where C_a represents the link's capacity and $\hat{C}$ represents the maximum capacity in link set A;
2) Distance of projects being carried out, represented by d_{ij}, the distance between two imaginary nodes, each of which is the middle point of an arc where a project is located at;
3) Parallel index θ, represented by the maximum number of projects being carried out in one month.

As depicted in Fig. 2, the ACO-based scheduler uses m ants to find a good schedule. Each ant will try to build their own schedule by searching in the solution space. Every ant is first assigned to a feasible schedule and then go to the next feasible schedule based on a probability. The probability of ant k moving from schedule i to schedule j can be expressed as follows[9]:

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ik}(t)]^{\beta}}{\sum\limits_{s\in\Omega_k}[\tau_{is}(t)]^{\alpha}\cdot[\eta_{is}(t)]^{\beta}}, & \text{if } j \notin \Omega_k \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

Where,

$\tau_{ij}(t)$ is the pheromone left in a path connecting any two elements i and j at time t;
$\eta_{ik}(t)$ is the heuristic function for finding next element (element i to element k);
Ω_k represents the Tabu list, containing the elements that have been travelled by ant k;
α is the pheromone factor that reflects the importance of pheromone when an ant makes decisions;
β is the heuristic factor that reflects the importance of heuristic information for finding next element to try;

Based on the aforementioned three factors, $\tau_{ij}(t)$, which is the pheromone that a link between elements i and j has at time t, is determined as:

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t-1) + \sum_{k=1}^{m}(\frac{1}{\theta})_k \tag{25}$$

Where,

ρ represents the evaporation rate, which indicates how fast the pheromone will evaporate and vanish along any link;

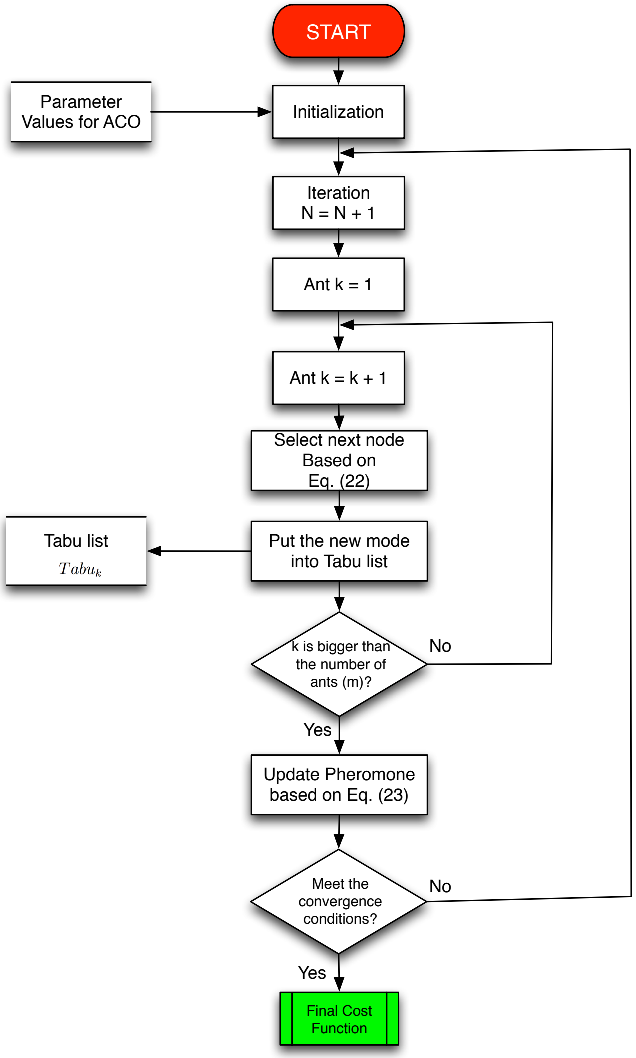


Fig. 2. Workflow of ACO-based Scheduler

$(\frac{1}{\theta})_k$ represents the pheromone left by ant k by considering the maximum parallel index in a month within the current schedule.

In this research, the heuristic information $\eta_{ik}(t)$ for each ant between any two elements has been chosen as

$$\eta_{ik}(t) = d_{ij}/\frac{C_a}{\hat{C}} \tag{26}$$

Where,

d_{ij} represents the the distance between two imaginary nodes in the road network, each of which is the middle point of a link where two projects p_i and p_j are located at respectively;

$\frac{C_a}{\hat{C}}$ represents the importance of a link a, where the candidate fake project for traversing is located at.

This is to say that the ant should try to find the next feasible schedule, so that it can be 1) as far as possible from the present one and 2) as less important as possible. Each ant stops its search after it has covered all projects. Moreover, each ant can ignore a feasible schedule if it is derived from previously traversed schedules. In this case, the distance between the two feasible schedules is zero, so $\eta_{ik}(t)$ is zero.

4) Final Cost Evaluation:

This component is the last step of CoANT and is used to evaluate the cost of the current schedule by ACO-based Scheduler $T(\hat{\mathbf{s}})$. The TTD is determined by substracting the cost of the total TSTT when no projects are conducted, i.e. $T(\hat{\mathbf{s}}) - T_0$. The scheduler and cost evaluation are executed several times according to the convergence conditions that specify the number of iterations of the solver, hence several feasible schedules can be generated. The schedule with the minimum cost is chosen as the final schedule or near-optimal schedule.

5) Optional Online Cost Evaluation:

This optional component can be used to evaluate each schedule when ACO-based scheduler is running. At every iteration, each ant considers not only the three factors mentioned in Section III-.3 as pheromones, but also the corresponding cost of the schedule each ant finds in an iteration. This optional component has been provided for future research regarding item two on Page 2, which is to approximate the calculation of cost based on the optimization model so that each schedule can be evaluated in a reasonable time.

IV. Implementation and Results

CoANT has been implemented in C++ under Ubuntu Linux 12.04 LTS 64 bit (Intel® i7 4770 CPU 3.40 GHZ, 16 GB memory, 256 GB Solid State Disk) and used in an example scenario. Dial's algorithm B [10] has been used to solve the lower level problems (within the Initial and Final Cost Evaluation components). A generic ACO algorithm has been used in the scheduler with 83 ants and 500 iteration limits. The file reading interface from Mathew Steel's work[1] has been used to be the standard input interface for network data. The scenario seeks to test the performance of CoANT in providing 1) candidate schedule in different trials; 2) stable output in different trials and 3) relatively better reduction rate compared to previous studies in [4] (11.1%).

The scenario tested is based on the Chicago sketch network [2], which contains 933 nodes, 2522 links and 387 zones (minutes, miles, cents, toll factor = 0.02 minutes/cent; distance factor = 0.04 minutes/mile). Six construction projects are to be scheduled, as shown in Table I and the distances between pairs of projects are shown in Table II.

The six projects in the scenario including the durations and time window were selected randomly in the beginning of the experiment, so each ant will stop searching after it has traversed six nodes in $\hat{P}$. In addition, based on some pre-test runs, the corresponding parameter values for ACO have been selected for this experiment: $\rho = 0.1$, $\alpha = 1$, $\beta = 1$

[1]https://github.com/MatthewSteel/EquilibriumSolver

[2]http://www.bgu.ac.il/ bargera/tntp/, 9/2013

TABLE I

THE PROJECTS USED IN THE EXAMPLE SCENARIO

Project ID	Duration *(Month)*	From Node	To Node	Capacity (*vehicles per hour*)	Maximum Capacity	Capacity/Max Capcity	Capacity Reduction with Construction	Length (*miles*)	Free Flow Travel Time (*min*)	Time Window (Proposed Total Duration, *Month*)
0	8	454	453	3500	49500	0.07	50%	9.13311	10.27	36
1	5	914	780	2000	49500	0.04	50%	11.5535	13.52	36
2	6	913	801	3500	49500	0.07	50%	20.8982	24.92	36
3	7	890	863	3000	49500	0.04	50%	6.34517	8.98	36
4	6	878	881	4500	49500	0.09	50%	6.22469	7.94	36
5	8	918	919	1500	49500	0.03	50%	5.978	7.89	36

TABLE II

DISTANCES BETWEEN EVERY TWO PROJECTS (FEET)

Project ID	0	1	2	3	4	5
0	0	220687	83848	176813	207705	402313
1		0	171244	308817	282253	192242
2			0	250665	266563	334487
3				0	70445	499893
4					0	467034
5						0

TABLE III

ILLUSTRATION OF THE BEST SOLUTION IN 30 EXECUTIONS

Project \ Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0																													■	■	■	■	■	■	■	■
1																								■	■	■	■	■								
2																			■	■	■	■	■	■												
3														■	■	■	■	■	■	■																
4										■	■	■	■	■	■																					
5			■	■	■	■	■	■	■	■																										

TABLE IV

ILLUSTRATION OF THE WORST SOLUTION IN 30 EXECUTIONS

Project \ Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0																			■	■	■	■	■	■	■	■										
1																														■	■	■	■	■		
2																													■	■	■	■	■	■		
3																											■	■	■	■	■	■	■			
4																■	■	■	■	■	■															
5																												■	■	■	■	■	■	■	■	

and the convergence conditions for CoANT has been set as 10 executions which takes approximately 45 minutes wall time to run.

This scenario was tested 30 times in order to test the performance and the robustness of CoANT. As shown in Fig. 3, CoANT generated candidate schedules as desired in 30 tests with five different solutions, whose TTD reduction rates are ranging from 42.4% to 53.5%, which is relatively better compared to previous studies [4] (11.1%). However, the stability of CoANT needs to be further investigated and the solution should be reduced to only one by, e.g., adopting more heuristic information. The best and worst schedules have been illustrated in Fig. III and Fig. IV respectively.

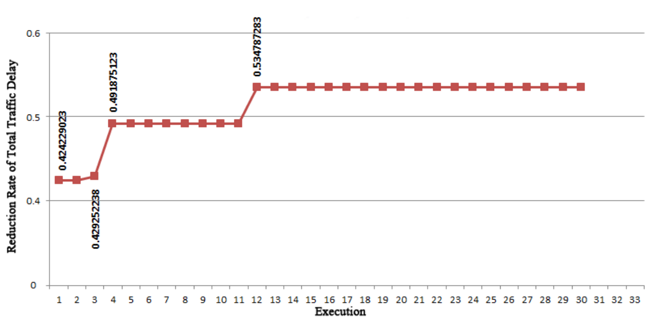


Fig. 3. Illustration of Solutions in 30 Executions

As demonstrated in the two solutions, the best and the worst, we can see that the heuristic information used in CoANT is taking effects as desired:

1) the maximum parallel index was four in the worst solution, which should be improved and reduced to less than two; however, since the distances between those project sites are relatively far as shown in Table II, the resulted delay reduction rate (42.4%) is still acceptable
2) the parallel projects are relatively far from each other, e.g., project 0 is 207705 feet (63307 metres) from project 4; which means, CoANT was trying to make parallel projects far from each other and reduce the traffic impact;
3) the nearest project sites are the pairs of 1) projects 0 and 2 and 2) projects 3 and 4. As illustrated in Fig. III and Fig. IV, those two pairs have been successfully scheduled separately without being executed in parallel.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a novel formulation for the CPSP and a solution algorithm were proposed to find the best schedule for a set of construction projects. The algorithmic framework developed is based on ACO and is able to integrate expert information. As indicated by the implementation carried out, CoANT is working as desired and the integration of heuristic information has influenced the scheduling process in a positive manner. CoANT can find a schedule that results in a better delay reduction rate compared to the ones reported in [4] (11.1%). Moreover, it calculated the total traffic delay 10 times in 45 minutes when scheduling, which is more time-efficient than greedy search-based methodologies.

Specifically, CoANT has the following advantages: 1) it relies on expert information, which can be used to incorporate any special needs from practitioners; 2) it tested an knowledge based approach and showed its performance in finding a near-optimal schedule with a better delay reduction rate and 3) it is modular and extensible. However, CoANT still needs development regarding its searching strategy. In general, future work can include:

1) tune the parameter values within CoANT, as the parameters for ACO can significantly influence the solution searching process and how long it will take to converge. Searching for an optimal parameter set will be the focus of future research;
2) introduce additional factors to accurately represent the impact of construction projects, e.g. day or night construction mode;
3) acquire experience or knowledge from project managers and/or relevant expert practitioners;
4) enable the Optional Online Cost Evaluation Module, so that the ACO-based Scheduler can rapidly evaluate the cost of a candidate schedule. This module should be based on some efficient approximations, such as the approximate UE computation [1], [8].

In conclusion, as an algorithm designed to tackle the CPSP by integrating expert knowledge, CoANT has shown some promising results on realistic scheduling scenarios. An additional effort is required so as to integrate more specific project-oriented information within the ACO-scheduler and the Final Cost Evaluation and allow the impact of a schedule to be evaluated based on individual needs. Moreover, CoANT can be also adapted to alternative bi-level optimization problems.

References

[1] H. Zheng, C. Cai, E. J. Nava, V. V. Dixit, Y.-C. Chiu Ph D, E. Radwan, and D. Ismart, "Optimization of renewal-based project scheduling in an urban network," in *Transportation Research Board 91st Annual Meeting*, no. 12-4527, 2012.

[2] M. Ng, D. Y. Lin, and S. T. Waller, "Optimal long-term infrastructure maintenance planning accounting for traffic dynamics," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 7, pp. 459–469, 2009.

[3] C. F. Daganzo, "The cell transmission model, part ii: network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.

[4] H.-Y. Lee, "Optimizing schedule for improving the traffic impact of work zone on roads," *Automation in Construction*, vol. 18, no. 8, pp. 1034–1044, 2009.

[5] Y.-Y. Chang, O. Sawaya, and A. Ziliaskopoulos, "A tabu search based approach for work zone scheduling," in *Proceeding of the Transportation Research Board 80th Annual Meeting, Washington, DC*, 2001.

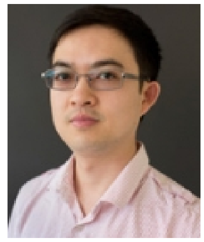
[6] Y. Sheffi, "Urban transportation networks: equilibrium analysis with mathematical programming methods," 1985.

[7] K. Lukas and A. Borrmann, "Minimizing the traffic impact caused by infrastructure maintenance using ant colony optimization," in *Proc., the 28th International Conference on Automation and Robotics in Construction, Seoul, Korea*, 2011.

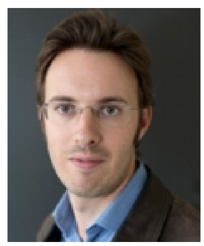
[8] S. D. Boyles, "Bush-based sensitivity analysis for approximating subnetwork diversion," *Transportation Research Part B*, vol. 46, pp. 139–155, 2012.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, 1996.

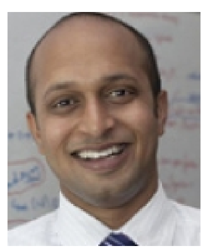
[10] R. B. Dial, "A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration," *Transportation Research Part B: Methodological*, vol. 40, no. 10, pp. 917–936, 2006.



Zhitao Xiong is a post-doctoral researcher at the Research Center for Integrated Transport Innovation (rCITI) in the University of New South Wales (UNSW). He started his post-doc appointment at rCITI in his fourth year of PhD at the Institute for Transport Studies in the University of Leeds. His PhD research focused on creating repeatable and realistic driving simulation for human factor researches. His research interests include driver models, driving behaviours, knowledge representation and automated action planning.



David Rey is a post-doctoral researcher in transport engineering at the Research Center for Integrated Transport Innovation (rCITI) at the University of New South Wales (UNSW). He joined UNSW after obtaining his PhD in operations research from the Université de Grenoble during which he studied conflict resolution strategies arising in air traffic management. His key research interests lie in resource allocation schemes in transportation networks, mathematical programming as well as fair optimization.



Vinayak V. Dixit is a Senior Lecturer at the Department of Civil and Environmental Engineering at the University of New South Wales (UNSW). His key research interest lies in studying risk in the transportation infrastructure system as it relates to highway safety, travel time uncertainty, as well as natural and man-made disasters.

S. Travis Waller is the Evans & Peck Professor of Transport Innovation and Director of the Research Centre for Integrated Transportation Innovation (rCITI) at the University of New South Wales (UNSW) in Civil and Environmental Engineering. Professor Waller was named one of the top 100 young innovators in science and engineering in the world in 2003 by MIT's Technology Review magazine, was awarded the National Science Foundation's CAREER award in 2004, received TRB's Fred Burggraf research award in 2009 and the 2011 Hojjat Adeli Award for Innovation in Computing.