



A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness



Yiyong Xiao^a, Abdullah Konak^{b,*}

^a School of Reliability and System Engineering, Beihang University, Beijing 100191, China

^b Information Sciences and Technology, Penn State Berks, Tulpehocken Road, P.O. Box 7009, Reading, PA 19610-6009, United States

ARTICLE INFO

Article history:

Received 2 August 2014

Received in revised form 20 March 2015

Accepted 27 April 2015

Available online 22 May 2015

Keywords:

Greenhouse gas emissions

Vehicle routing

Weighted tardiness

Mathematical programming

Simulated annealing

ABSTRACT

We present a green vehicle routing and scheduling problem (GVRSP) considering general time-dependent traffic conditions with the primary objective of minimizing CO₂ emissions and weighted tardiness. A new mathematical formulation is proposed to describe the GVRSP with hierarchical objectives and weighted tardiness. The proposed formulation is an alternative formulation of the GVRSP in the way that a vehicle is allowed to travel an arc in multiple time periods. The schedule of a vehicle is determined based on the actual distance that the vehicle travels each arc in each time period instead of the time point when the vehicle departs from each node. Thereby, more general time dependent traffic patterns can be considered in the model. The proposed formulation is studied using various objectives functions, such as minimizing the total CO₂ emissions, the total travel distance, and the total travel time. Computational results show that up to 50% reduction in CO₂ emissions can be achieved with average reductions of 12% and 28% compared to distance-oriented solutions and travel-time-oriented solutions, respectively. In addition, a simulated annealing (SA) algorithm is introduced to solve large-sized problem instances. To reduce the search space, the SA algorithm searches only for vehicle routes and rough schedules, and a straightforward heuristic procedure is used to determine near-optimal detailed schedules for a given set of routes. The performance of the SA algorithm is tested on large-sized problems with up to 100 nodes and 10 time periods.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

It is well recognized that carbon dioxide (CO₂) is the major contributor of the global warming effect of the Earth during the past decades. Since it was first measured in 1958, the concentration of CO₂ in Earth's atmosphere has been continuously increasing, recently at a rate of 1.7 ppm/yr for 1993–2004 and 2.3 ppm/yr for 2005–2014 (lively in <http://keelingcurve.ucsd.edu/>). According to the International Energy Agency (IEA), the transportation sector, after electricity generation and heating, was the second-largest contributor of CO₂ emissions, representing 22% of the global CO₂ emissions in 2010, and almost three-quarters of the emissions from transportation were due to road transportation [33]. Traffic congestion, which results in low speeds with fluctuations on roads, often accompanied with frequent acceleration and deceleration, had greatly contributed to CO₂ emissions [2,16]. According to the International Road Transport Union (IRTU), traffic congestion had

increased CO₂ emissions by 300% and is responsible for 100 billion liters of wasted fuel, or 250 billion tons of CO₂ emissions in the United States alone [21]. Therefore, reducing fossil fuel consumption and CO₂ emissions due to road transportation is an important part of the efforts to control the global warming. The objective of this paper is to develop new mathematical models and optimization methods for reducing CO₂ emissions in the context of the Vehicle Routing Problem with traffic congestion considerations. We propose a new mathematical formulation for modeling the time-dependency in vehicle routing and scheduling problems. In the proposed formulation, the schedule of a vehicle is modeled using the total distance that the vehicle travels on each road segment in each time period. This approach is different from the literature where the schedule of a vehicle is generally determined by the time points when the vehicle departs from each node on its route. The main advantage of the proposed formulation is that more general time-dependent traffic patterns can be considered. In addition, the complexity of calculating CO₂ emissions within the model is eliminated, which leads to a linear mathematical formulation. The proposed formulation is appropriate for urban transportation

* Corresponding author. Tel.: +1 6103966310.

networks where the travel speed of a vehicle depends on the road conditions and the time of the day.

The Vehicle Routing Problem (VRP) has been a classic and important optimization problem involved in road transportation applications since its introduction by [7]. In general, the VRP is concerned with the determination of the optimal routes used by a fleet of vehicles, stationed at one or multiple depots, with the objective of fulfilling customers' demands at minimum cost. Various versions of the VRP have been developed for different applications in the past half century, such as pickup and delivery VRP, capacitated VRP, multiple depot VRP, VRP with time windows, split delivery VRP, time-dependent VRP, etc. Surveys on various formulations and algorithms of the VRP can be found in [10,18,26,35].

Recently, the Green VRP (GVRP) [8,27], which is characterized by the objective of balancing environmental and operational costs, has attracted the attention of researchers in the VRP literature. Ericsson et al. [13] identified the impact of traffic disturbance events on fuel consumption and proposed a model for estimating the potential reduction in fuel consumption through route optimization. Kara et al. [23] proposed a cost function in terms of energy consumption for the VRP and named it as the Energy Minimizing VRP (EMVRP), which minimizes the total energy consumption in the route (instead of the total distance) to minimize fuel consumption. Reducing CO₂ emissions also helps to save the fuel cost because CO₂ emissions are usually proportional to the amount of the fuel consumption. The fuel cost accounts for more than 60% of the total transportation cost [30,31] of road logistic companies. Tavares et al. [32] considered the effect of both road inclination and vehicle load on the fuel consumption of waste collection vehicles. Kuo [24] proposed a mathematical model to calculate the total fuel consumption for the time-dependent VRP, considering both load weight and the "non-passing" property. Figliozzi [15] proposed a partial emissions minimizing VRP (EVRP) model to optimize the departure times of vehicles on the routes that are found by a time-dependent VRP algorithm. Xiao et al. [38] incorporated the fuel consumption rate resulted from vehicle's load (which is decreasing/increasing throughout the tour) into the Capacitated VRP (CVRP) and proposed a model considering fuel consumption rate for the CVRP. Erdogan and Miller-Hooks [12] presented a GVRP and developed solution techniques to aid organizations with alternative fuel-powered vehicle fleets working in large driving range in conjunction with limited refueling infrastructure. Kwon et al. [25] developed a Heterogeneous fixed fleet VRP model that considers the carbon emissions trade cost in the objective function. Gaur et al. [17] studied the cumulative VRPs with the objective to minimize fuel consumption and proposed an approximation algorithm for cumulative VRPs when vehicles have finite capacity and an arbitrary number of depot offloads are allowed. Demir et al. [8] proposed an adaptive large neighborhood search algorithm (ALNS) to minimize the fuel consumption and the driving time with Pareto optimality.

Traffic congestion usually causes road conditions to be time-dependent, and the VRP under this case is called the time-dependent VRP (TDVRP) [5,14,28,29]. Compared with the distance-based VRP, the node departure times of a vehicle are decision variables in the TDVRP. Because the total travel time depends on node departure times as well as routes, the TDVRP is usually studied with alternative objectives such as minimizing the travel time or the fuel consumption (or CO₂ emissions) [15,24]. The TDVRP is a more challenging problem to solve than the traditional VRP because the solution space is increased exponentially by the introduction of node departure time decisions.

Bektas and Laporte [4] presented the pollution routing problem (PRP) by extending the classical VRP with time windows (VRPTW) as well as with a more comprehensive objective function including fuel, emissions, and driver costs. In the PRP, the vehicle load and

speed on each route segment are considered as decision variables. Demir et al. (2012) developed a two-stage heuristic algorithm to solve the PRP of large-size instances with up to 200 nodes. In this two-stage approach, a large neighborhood search (LNS) heuristic is used to solve traditional VRPTW in the first stage, and the optimal speed for each arc of the route is determined optimally in the second state. Franceschetti et al. [16] extended the PRP to Time-Dependent PRP (TDPRP) by considering traffic congestion. In their approach, the planning horizon is divided into two periods: (1) an initial period of traffic congestion with a lower and constant travel speed and (2) a period with a travel speed range where the travel time linearly depends on the departure time.

Table 1 summarizes and compares the formulations of the PRP in the literature and the proposed formulation. In this paper, we consider a general case of the time-dependency for vehicle routing and scheduling with the hierarchical objectives of the total CO₂ emissions, the total arrival time, the total travel time, and the total travel distance. A tardiness objective is considered for the first time in the context of the GVRP. Tardiness objectives are frequently used in the scheduling literature to model the timeliness of customer service when it is impossible to satisfy all customer orders on time due to capacity limitations. In a tardiness objective, the late service of a customer is penalized by a tardiness penalty coefficient that represents the loss of customer goodwill. The magnitude of the tardiness penalty depends on the relative importance of customers.

In this paper, the formulated problem is referred to as Green Vehicle Routing and Scheduling Problem (GVRSP). The proposed GVRSP formulation contributes to the existing body of the research on the green time-dependent VRP in several ways. While the existing time-dependent GVRP formulations assume that an arc is allowed to be traveled in only one time period, in the GVRSP herein an arc can be traveled in multiple time periods. We present an alternative formulation for modeling time-dependency using the total distance traveled on an arc in each period as the primary decision variable. Thereby, the proposed modeling approach allows to incorporate more general patterns of traffic scenarios in the formulation. The proposed GVRSP formulation is linear and can be solved optimally by the existing MIP solvers for small-sized problems. In this sense, the proposed formulation is a more general model for combined routing and scheduling decisions in the context of the GVRSP. We also propose a simulated annealing (SA) algorithm to solve medium & large-sized problems with near-optimal solutions. Computational experiments are carried out to examine the proposed MILP model and solution approaches using a large set of problem instances.

The rest of the paper is organized as follows. We first describe the GVRSP in Section 2. In Section 3, we present the MILP model with hierarchical objectives and develop a solution approach based on Pareto optimality for its real-life application under a carbon trading system. Computational studies are also carried out in this section. In Section 4, a SA algorithm is proposed and computational experiments are carried out to examine its effectiveness and efficiency. Finally, in Section 5 we conclude the paper.

2. Problem description

The GVRSP in this paper can be described as follows. A fleet of homogeneous vehicles is going to visit a set of customers randomly located in a region. The problem is formulated on a complete network $G(N, A)$ where N is the set of nodes representing customers and a depot, and A is the set of arcs representing the roads between the nodes. All vehicles are located in the depot and expected to return to the depot after completing their tours. Each customer i should be visited within the time-window $[0, T_i]$ where T_i is the due-time of the customer. A late visit results in a tardiness penalty

Table 1
A comparison of existing pollution routing models.

Reference	Linear?	Time dependent	Objective function	Heterogeneous vehicle	Weight	Independent Decision Variables	Idle Time	Solution Approach
Kara et al. [23]	Yes	No	Energy consumption	No	Yes	Routes	No	MIP solver
Kuo [24]	–	Yes	Fuel consumption	No	Yes	Routes	No	Simulated annealing
Figliozzi [15]	No	Yes	Vehicles cost; Total Distance; Route durations; Emissions	No	No	Routes; Service time	Yes, only at nodes	Two optimizing procedures for departure time and route
Bektas and Laporte [4]	Yes	No	Drivers cost; Operational costs; Emissions cost	No	No	Route; Speed	No	ALNS by Demir et al. [8,9]
Jabali et al. [22] (no model)	–	Yes	Driver cost; Fuel cost Emissions cost	No		Routes; Speed; Route start time	No	Tabu search
Xiao et al. [38]	Yes	No	Fuel consumption	No	Yes	Routes	No	Simulated Annealing
Kwon et al. [25]	Yes	No	Operation cost; Carbon trading cost	Yes	No	Routes; Vehicle types	No	Tabu-search
Gaur et al. [17] (no model)	–	No	Fuel consumption	No	No	Routes	No	Approximation algorithms
Franceschetti et al. [16]	Yes	Only two fixed periods	Emissions cost; Driver wage	No	Yes	Routes; Speed; Departure time	Only at nodes	MIP & DSOP algorithm, Maximum size: 20 nodes.
This paper	Yes	Yes	Hierarchical objectives with tardiness: CO ₂ Emissions; Customer satisfaction; Travel time; Travel distance	No	No	Routes; Travel distance in each period; Node arrival times; Node departure times	Yes, both at nodes and arcs	MIP & Simulated Annealing, Maximum size: 100 nodes × 10 periods

Note: “No model” means the model is not formulated as a mathematical programming model.

indicating the importance of the customer while an early visit does not result in any reward, but it is still encouraged as a secondary objective to improve customer satisfaction. The average travel speeds on the arcs are assumed to be time-dependent as

shown in Fig. 1. Given the vehicle parameters (e.g., engine efficiency, weight etc.) and arc travel conditions (e.g., average travel speed and slope), CO₂ emissions rate (Lb/mile) for each arc in each time period can be calculated using the emission models from

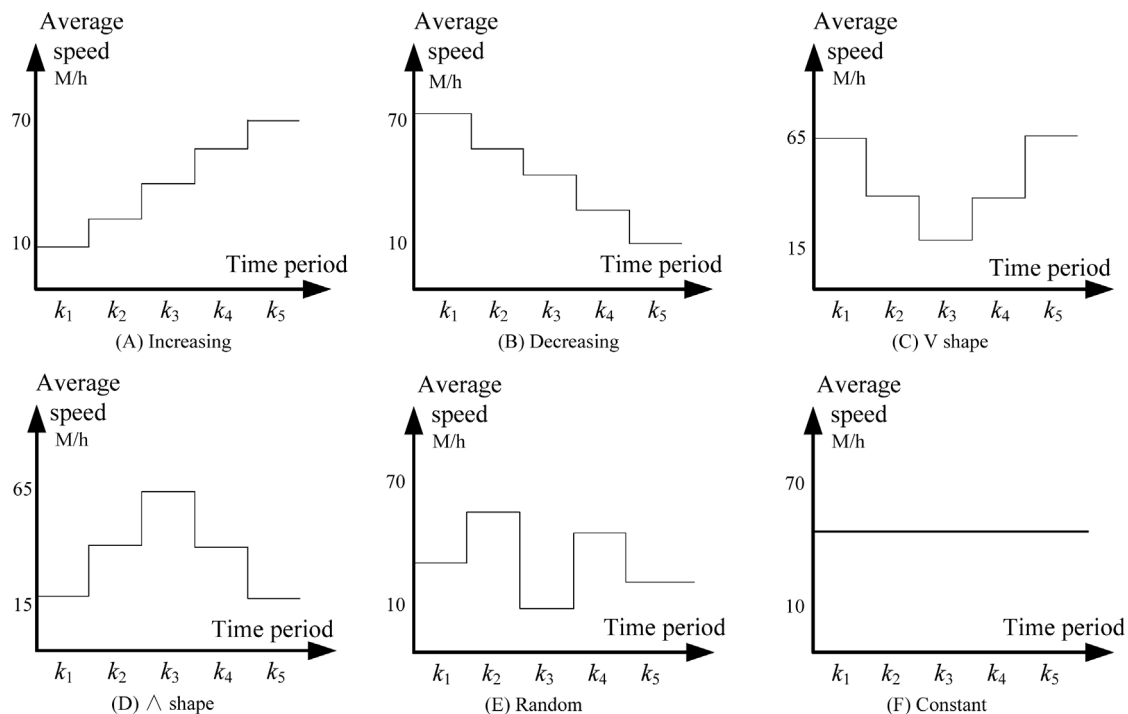


Fig. 1. Road speed patterns.

the literature (e.g., [2,3,16]) or using the Motor Vehicle Emission Simulator (MOVES) [11]. The objective of CO₂ emissions may also be strictly restricted to a maximum emissions allowance amount under a carbon-trading market. If a company exceeds its CO₂ emissions limit, it may be penalized by extra tax or required to buy more allowance from the carbon trading market. Therefore, minimizing CO₂ emissions is considered as the primary objective in the GVRSP. Meanwhile, other secondary objectives such as the total arrival time (TAR), which reflects the customer satisfaction level, the total route time, which reflects the driver and vehicle usage cost, and the total route distance (for benchmarking) are also considered in a hierarchical manner.

The decision variables of the GVRSP can be grouped into two types: (1) the routing decision variables to select arcs in order to visit all customers and return back to the depot, and (2) the scheduling decision variables to determine the distance traveled on each arc in each time period and to calculate the departure time from each selected node. The routing decision is exactly the same as in the traditional VRP. The scheduling decision, which is considered on the continuous time domain in this paper, deals with how to schedule detailed travel assignments within the planning horizon of multiple time periods.

2.1. Modeling time-dependency

The time-dependency usually involves how to model the time-dependent travel speed and travel time. In the literature, the departure time from a node is considered as a decision variable [5,14,28,29], and it is assumed that an arc can be traveled only in a single time period. In the GVRSP herein on the other hand, an arc can be traveled in multiple time periods. We assume that the planning horizon is arbitrarily divided into multiple time periods. For each arc, the travel speed in a time period is assumed to be constant, representing the average speed during the period. This assumption is well justified in the case of urban and sub-urban roads where the average travel speed depends on the time of the day and is dictated by the road conditions. Fig. 1 shows six general types of travel-speed patterns that may be observed in real-life situations. These patterns also provide the basis for the numerical simulation in the following section of computational studies. The pattern *Increasing* and *Decreasing* in Fig. 1A and B simulates a steady increasing or decreasing travel-speed pattern in the planning horizon. The pattern *V shape* in Fig. 1C simulates a typical rush-hour traffic congestion. The pattern *A shape* in Fig. 1D simulates a time range between the two peaks of traffic congestion. The pattern *Random* in Fig. 1E simulates other irregular patterns for travel-speed patterns that may be caused by school zones, engineering works, etc. Fig. 1F simulates a road with either a high or low, but steady travel-speed in all periods.

In the GRVSP formulation, arc (i, j) represents the path between nodes i and j . Let D_{ij} be the total distance of arc (i, j) and d_{ijk} denotes the distance traveled on arc (i, j) in period k , such that $D_{ij} = \sum_{k=1}^m d_{ijk}$ where m is the number of time periods. Then, arc (i, j) 's travel time t_{ij} can be calculated as $t_{ij} = \sum_{k=1}^m d_{ijk}/v_{ijk}$, where v_{ijk} is the average travel speed on arc (i, j) in period k . Because v_{ijk} can be pre-calculated based on the traffic conditions of arc (i, j) in period k before solving the GRVSP formulation, the relationship between the time and the distance can be expressed as a linear equation in the formulation. In addition, CO₂ emissions (c_{ijk}) per unit distance traveled on arc (i, j) in period k can also be pre-calculated. Thereby, the objective function of the total CO₂ emissions also becomes linear. In real-life implementations, c_{ijk} can be estimated from empirical data or calculated based on emission models as discussed earlier.

The starting and ending times for traveling arc (i, j) in period k are represented as l_i and a_j , respectively. Since a vehicle's travel

speed is determined by the periods when it actually travels the arcs, not by the departure time points from the nodes, the *passing* phenomenon, i.e. a late departed vehicle may pass an earlier departed vehicle due to their different starting speeds [1], will not happen. The “first-in-first-out (FIFO)” property [20] is also satisfied. Moreover, the concept *idle time*, which is proposed by Franceschetti et al. [16] to avoid high levels of emissions, can also be calculated and scheduled at the customer nodes or on the roads, allowing a more flexible approach to scheduling.

2.2. Optimizing CO₂ emissions with hierarchical objectives for a carbon trading system

In 1997, the foundation of a carbon trading market was established as an outcome of the well-known Kyoto Protocol [36] to reduce the global carbon emissions. The EU emissions trading system was first started in 2005 and used grandfathering to allocate free allowances to some 13,000 companies in 31 countries, covering around 45% of the EU's greenhouse gas emissions [6]. According to Thomson Reuters Point Carbon [34], the value of the globally traded carbon market will increase by two thirds in 2014 from 2013 to reach €64bn, (€39bn in 2013), with the volume increasing by 3 percent to 9.6 Gt CO₂. A participating firm can sell any surplus emission allowances to another firm that needs emission allowances, or it can save them up for the future use.

One of the objectives of the GVRSP formulated herein is to assist transportation companies to optimize their amount of CO₂ emissions by making decisions about purchasing and selling emissions in a carbon market. Therefore, the GVRSP sets the amount of CO₂ emissions with a tardiness penalty as the primary objective in the first stage and optimizes other sub-objectives based on a controllable relaxation on the minimized CO₂ emissions. The sub-objectives, such as (1) the total arrival time (TAR)-representing the customer satisfaction, (2) the total travel time on road (TOR)-representing company's business performance indicators (e.g., driver and vehicle costs), and (3) the total distance traveled, are necessary to assist companies to pursue a higher level of business performance or lower operational costs.

3. A mixed integer linear programming model for the GVRSP

In this section, we present a mixed integer linear programming (MILP) model for the GVRSP with multiple objectives and develop an approach using the AMPL/CPLEX scripting language to solve the MILP model. Computational experiments are also carried out to study of the GVRSP and to examine the performance of the solution approach.

3.1. The MILP model

In the following, the parameters and decision variables to model the GVRSP are listed.

- q : number of homogeneous vehicles
- n : total number of nodes (customers plus the depot)
- i : index of nodes (the depot is represented by 0)
- N : set of the nodes including the depot, $i \in N$
- N' : set of the nodes excluding the depot, $N' = N \setminus \{0\}$
- A : set of the arcs formed by all pairs of nodes, $(i, j) \in A, \forall i \in N, j \in N, i \neq j$
- T_i : due-time of the i th customer
- ω_i : tardiness penalty coefficient of the i th customer
- D_{ij} : distance of arc (i, j)
- m : number of time periods

k : index of a time period, $k = 1, 2, \dots, m$

K : set of time periods, $k \in K$

$[b_k, e_k]$: beginning and ending time of time period k

Z_k : length of time period k , i.e., $Z_k = e_k - s_k$

v_{ijk} : average travel speed on arc (i, j) within time period k

c_{ijk} : CO₂ emissions rate per unit distance on arc (i, j) in time period k (pre-calculated)

M : a large number

The binary decision variables and non-negative continuous decision variables used in the MILP model are defined as follows:

X_{ij} : binary variable indicating whether arc (i, j) is traveled ($X_{ij} = 1$) or not ($X_{ij} = 0$)

x_{ijk} : binary variable indicating whether arc (i, j) is traveled in time period k ($x_{ijk} = 1$) or not ($x_{ijk} = 0$)

d_{ijk} : traveled distance of arc (i, j) in time period k

t_{ijk} : travel time of arc (i, j) in time period k

l_i : departure time from node i

a_i : arrival time at node i

L_i : tardiness of node i

Three hierarchical objectives are considered in the GVRSP. The primary objective is to minimize the total CO₂ emissions. The second level objective is to minimize the total arrival time (TAR). The second level objective requires serving the customers as early as possible. The third level objective is to minimize the total time on road (TOR). All three objectives also include a weighted tardiness penalty. Three levels of TOR can be defined as follows: (1) TOR1—the total time between departing from and returning back to the depot, i.e., $a_0 - l_0$, (2) TOR2—the total time on road excluding the idle time on the customer's premise (but including the scheduled idle time on road), i.e., $\sum_{i=0}^n (a_i - l_i)$, and (3) TOR3—only the travel time on road excluding all scheduled idle times, i.e., $\sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m t_{ijk}$. Note that the proposed formulation and solution approach herein are independent from the objective functions. These three definitions of TOR are used to demonstrate how transportation companies can use the formulation in real-life scenarios to determine a time-based wage system for drivers. In addition, for the cases where tardiness is not a concern, the tardiness penalty coefficient can be set zero. The MILP formulation for the GVRSP is given as follows.

$$\text{Min CO}_2\text{-Emission (CO}_2\text{-E)} = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ijk} \cdot c_{ijk} + \sum_{i=1}^n \omega_i \cdot L_i \quad (1)$$

$$\text{Min Total arrival time (TAR)} = \sum_{i=1}^n a_i + \sum_{i=1}^n \omega_i \cdot L_i \quad (2)$$

$$\text{Min Total time on road (TOR3)} = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m t_{ijk} + \sum_{i=1}^n \omega_i \cdot L_i \quad (3)$$

Subject to

$$\sum_{(i,j) \in A} X_{ij} = 1 \quad \forall i \in N' \quad (1)$$

$$\sum_{(i,j)} X_{ij} = 1 \quad \forall j \in N' \quad (2)$$

$$\sum_{j=1}^n X_{0j} \leq q \quad (3)$$

$$X_{ij} D_{ij} = \sum_{k=1}^m d_{ijk} \quad \forall (i, j) \in A \quad (4)$$

$$X_{ij} \geq x_{ijk} \quad \forall (i, j) \in A, \quad k \in K \quad (5)$$

$$X_{ij} \leq \sum_{k=1}^m x_{ijk} \quad \forall (i, j) \in A \quad (6)$$

$$d_{ijk} \leq D_{ij} \cdot x_{ijk} \quad \forall (i, j) \in A, \quad k \in K \quad (7)$$

$$x_{ijk} \leq M \cdot d_{ijk} \quad \forall (i, j) \in A, \quad k \in K \quad (8)$$

$$l_i \leq e_k - t_{ijk} + e_m \cdot (1 - x_{ijk}) \quad \forall (i, j) \in A, \quad k \in K \quad (9)$$

$$a_j \geq b_k + t_{ijk} - e_m \cdot (1 - x_{ijk}) \quad \forall (i, j) \in A, \quad k \in K \quad (10)$$

$$a_i \leq l_i \quad \forall i \in N' \quad (11)$$

$$a_0 \leq e_m \quad (12)$$

$$a_j \geq l_i + \sum_{k=1}^m t_{ijk} - e_m \cdot (1 - X_{ij}) \quad \forall (i, j) \in A \quad (13)$$

$$Z_k \geq \sum_{(i,j) \in A} t_{ijk} \quad \forall k \in K \quad (14)$$

$$L_i \geq a_i - T_i \quad \forall i \in N' \quad (15)$$

$$t_{ijk} = 60 \cdot d_{ijk} / v_{ijk} \quad \forall (i, j) \in A, \quad k \in K \quad (16)$$

$$X_{ij} \in \{0, 1\}, x_{ijk} \in \{0, 1\}, d_{ijk} \geq 0, t_{ijk} \geq 0, l_i \geq 0, a_i \geq 0, L_i \geq 0 \quad \forall (i, j) \in A, \quad k \in K \quad (17)$$

In the model above, Constraints (1) and (2) ensure that the vehicles visit all customers only once. Constraint (3) ensures at most q vehicles are used. Constraint (4) guarantees that if an arc is selected, its full distance must be traveled. Constraints (5) and (6) make variable X_{ij} consistent with variables x_{ijk} , and Constraints (7) and (8) make variables x_{ijk} and d_{ijk} consistent with one another. The large number M requires that a minimum distance, i.e., $1/M$, must be traveled in time period k if arc (i, j) is selected to be traveled in that period. Constraints (9) and (10) bound the departure and arrival times of the nodes within proper time intervals. Constraint (9) ensures that if arc (i, j) is traveled in period k , the departure time of node i must be ahead of the end time of period k by the travel time of arc (i, j) in period k . Similarly, Constraint (10) states that a vehicle cannot enter node j before the start time of period k plus the travel time of arc (i, j) in that period. Note that for $x_{ijk} = 0$, Constraints (9) and (10) still hold because e_m is always greater than or equal to any of the departure and arrival times. Constraint (11) ensures that a vehicle's departure time must be after its arriving time. If a customer service time g_i is required, then Constraint (11) can be modified as $a_i + g_i \leq l_i \quad \forall i \in N'$. Constraint (12) requires that the return time to the depot must not exceed the end of the planning horizon. Constraint (13) is a disjunctive constraint to calculate the arriving time at node j by cumulatively considering the travel times of all arcs and the leaving times of all nodes that would have been traveled before node j . Note that Constraint (13) is inactive for an arc which is not used in the tour. Moreover, Constraints (11) and (13) prevent any sub-tours of the customers. Constraint (14) prevents the time periods from being over used. Constraint (15) is a linear expression for calculating the tardiness. Constraint (16) calculates the travel time of arc (i, j) in period k . Note that Constraint (16) is a linear constraint because the values of v_{ijk} are all pre-calculated parameters. Thereby, it is possible to express the non-linear relationship between the CO₂ emissions and the travel-speed as linear

constraints without requiring auxiliary binary variables for linearization. To achieve this objective, it is assumed that the travel speed of an arc is constant within a time period. This assumption is suitable for real-life operations of commercial fleets that must follow speed limits strictly. In addition, all mathematical expressions in the model are linear. Therefore, it is possible to optimally solve the formulation using MIP solvers.

In addition to the constraints above, Constraints (18) and (19) given below are useful valid inequalities to tighten the formulation and for developing heuristics to solve the problem.

$$x_{jlk_1} \leq 2 - x_{ijk_2} - x_{ij} \quad \forall i, j, l \in N'; k_1, k_2 \in K; k_1 < k_2 \quad (18)$$

$$x_{jlk_1} + x_{jik_2} \leq 1 \quad \forall i, j \in N; k_1, k_2 \in K \quad (19)$$

Constraint (18) states that if arc (i, j) is traveled in time period k_2 , all arcs emanating from node j cannot be traveled in a time period k_1 such that $k_1 < k_2$. Constraint (19) prevents the vehicle from going back on the same arc.

3.2. Solving the MILP model with hierarchical objectives

In this section, a three-stage approach is introduced to solve the MILP model with hierarchical objectives and Pareto optimality. Two main motivations of solving the problem in three stages with hierarchical objectives are to demonstrate how the formulation can be used in practice and to illustrate the differences in the optimal solutions found by various objective functions. In the first stage, the problem is solved using the CO_2 -E objective function only, and the resulting optimal objective function value is denoted as CO_2E^* . The first stage MILP model consists of Eq. (1) and Constraints (1)–(17). In the second stage, the objective function is changed to TAR, and Constraint (20) is added to the model. The resulting optimal objective function value of the second stage is denoted as TAR^* . Parameter λ_1 in Constraint (20) is a non-negative relaxation parameter on the CO_2 emissions constraint. A positive λ_1 represents pursuing better customer satisfaction at the expense of higher CO_2 emissions in the second stage. When CO_2E^* is less than the allowance amount, and the company is not intending to sale the surplus emissions in the carbon trading market, Constraint (20) can be relaxed to achieve better business performance in the second stage. On the other hand, if CO_2E^* already exceeds the allowance amount, then Constraint (20) can allow the company to buy a controllable amount of emissions from the market. In the third stage, the objective is changed to TOR3, and Constraint (21) with another relaxation parameter λ_2 is added to the MILP model. Parameter λ_2 represents the extent to sacrifice customer satisfaction in order to reduce the operational cost. After all three stages are solved, the final solution for the objectives CO_2E , TAR and TOR3 is a Pareto optimal with respect to parameters λ_1 and λ_2 .

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ijk} \cdot c_{ijk} \leq CO_2E^* \times (1 + \lambda_1) \quad (20)$$

$$\sum_{i=1}^n a_i \leq TAR^* \times (1 + \lambda_2) \quad (21)$$

The process of solving the problem with the hierarchical objectives is shown in Fig. 2.

Note that three objectives CO_2 -E, TAR, and TOR can also be used alone as independent objectives. In the next section, we compare solutions that are found using different objective functions. In addition, these three objectives, a distance-oriented objective is considered as follows:

$$\text{Min Total_travel_distance (TTD)} = \sum_{i=0}^n \sum_{j=0}^n x_{ij} \cdot D_{ij} + \sum_{i=1}^n \omega_i \cdot L_i \quad (4)$$

- 1) Set CO_2 -E as the objective;
- 2) Solve the MILP model to obtain optimal value CO_2E^* ;
- 3) Set TAR as the objective;
- 4) Select a value for λ_1 and add Constraint (20) to the MILP model;
- 5) Solve the MILP model to obtain the optimal value TAR^* ;
- 6) Set TOR3 as the objective;
- 7) Select a value for λ_2 and add Constraint (21) to the MILP model;
- 8) Solve the MILP model to obtain the optimal value TOR^* ;
- 9) Return vehicles' route and schedule.

Fig. 2. A three-stage MILP approach for solving the problem with multiple objectives.

We refer to the solutions found using only CO_2 -E, TAR, TOR, or TTD objectives as CO_2 -oriented, TAR-oriented, TOR-oriented, distance-oriented solutions, respectively.

3.3. Computational studies of the MILP model for real-life applications

We first investigated differences between CO_2 -oriented and distance-oriented solutions in terms of the total CO_2 emissions, travel time, and travel distance. Problem instances were generated to simulate the morning rush time in a 10×10 mile urban area. A V-shape pattern with five time periods was used to generate time-dependent traffic conditions as shown in Table 2. In each period, the average travel speed was randomly determined within the corresponding speed ranges. A set of customers was randomly located in the target area, and all customers were to be served until 10:00am by a single vehicle.

We used the fuel consumption function developed by Franceschetti et al. [16], which is based on the comprehensive modal emissions model (CMEM) by [3] and [2], to estimate CO_2 emissions based on the average travel speed. In Eq. (5), the relationship between the CO_2 emissions rate (Lb per mile, denoted as e) and the vehicle's travel speed (Mile per hour, denoted as v) is given for an unloaded vehicle with a gross weight of 6.35 t. The coefficients of the equation were partially calculated based on the data provided by Franceschetti et al. [16], considering the gasoline density (e.g., 1.62 Lb/l) and converting the rate from gasoline to CO_2 (largely 3.3 Lb/Lb). The curve of CO_2 emissions with respect to vehicle's travel speed is plotted in Fig. 3.

$$e = 19.585074626866 \times v^{-1} + 0.00024281605015226 \times v^2 + 0.4590900784476 \quad (5)$$

We generated three groups of ten random problem instances with a single vehicle and the number of customers $n=5, 10$, and 15 (a total of 30 problem instances). These instances were optimally solved using AMPL/CPLEX (version 12.4.0.1) in an iMac with 2.50 GHz Intel® Core™ i5-2400S CPU. Distance-oriented solutions were also determined optimally for comparison. The tardiness penalty was not considered in this set of experiments so that the optimal solutions would not depend in the tardiness weight. We present the summary of the CO_2 -oriented and distance-oriented solutions in Table 3 and the detailed results in Table 4.

As it can be observed in Tables 3 and 4, the CO_2 -oriented optimal solutions resulted in significantly lower levels of CO_2 emissions compared to the distance-oriented optimal solutions. The maximum CO_2 emissions reduction was 18.6% with an average of 14.2%. The average CO_2 emissions reduction was also increased with the

Table 2
Speed changing pattern in the morning of an urban region.

Period	Time range	Period length (m)	Avg. speed	Speed range (miles/h)
1	7:00 AM–7:30AM	30	Normal	35–45
2	7:30 AM–8:00AM	30	Low	15–25
3	8:00 AM–8:30AM	30	Very low	5–15
4	8:30 AM–9:00AM	30	Low	15–25
5	9:00 AM–10:00AM	60	Normal	35–45

Table 3
Summary comparisons of CO₂-oriented and distance-oriented solutions.

Group	Ins.	CO ₂ decrease (%)			Travel time change (%)			Travel distance change (%)		
		Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.
5 × 5	10	−6.0	0.0	−2.9	−16.9	14.2	−6.2	0.1	0.0	0.0
10 × 5	10	−18.2	−0.5	−9.0	−44.0	2.8	−26.6	0.0	0.0	0.0
15 × 5	10	−18.6	−8.9	−14.2	−45.9	−26.8	−39.2	0.1	0.0	0.0

Table 4
Detailed results of the distance-oriented and CO₂-oriented models.

Group	No.	Distance-oriented			CO ₂ -oriented					
		TTD	CO ₂	TOR1	TTD	ΔTTD%	CO ₂	ΔCO ₂ %	TOR1	ΔTOR1%
5 × 5	1	19.9	26.6	31.3	19.9	0.0	26.4	−0.7	31.0	−0.8
	2	22.5	30.8	39.3	22.5	0.0	29.9	−2.8	35.9	−8.7
	3	28.6	40.2	54.3	28.6	0.0	38.1	−5.2	45.1	−16.9
	4	23.1	32.5	42.6	23.1	0.0	30.6	−6.0	37.0	−13.2
	5	26.6	37.1	46.0	26.6	0.1	35.3	−4.6	43.0	−6.4
	6	19.8	26.9	28.5	19.8	0.0	26.2	−2.5	32.6	14.2
	7	22.8	31.5	39.9	22.8	0.0	30.5	−3.0	34.7	−12.9
	8	17.7	23.5	27.5	17.7	0.0	23.5	0.0	27.5	0.0
	9	26.3	36.5	48.1	26.3	0.0	35.0	−4.2	41.4	−14.1
	10	20.8	28.0	32.3	20.8	0.0	27.9	−0.4	31.3	−3.1
	Avg.	22.8	31.4	39.0	22.8	0.0	30.3	−2.9	36.0	−7.8
10 × 5	1	32.1	48.8	75.8	32.1	0.0	43.1	−11.8	48.6	−36.0
	2	37.1	60.7	100.9	37.1	0.0	49.7	−18.2	56.5	−44.0
	3	29.4	43.3	67.1	29.4	0.0	39.2	−9.4	45.8	−31.7
	4	35.0	56.1	94.4	35.0	0.0	46.5	−17.1	55.8	−40.9
	5	28.2	39.5	53.2	28.2	0.0	37.8	−4.4	42.9	−19.2
	6	29.7	42.0	58.7	29.7	0.0	39.9	−5.0	44.5	−24.1
	7	26.3	36.0	43.6	26.3	0.0	35.2	−2.4	40.7	−6.7
	8	29.7	42.7	63.4	29.7	0.0	39.5	−7.4	45.7	−27.9
	9	19.4	26.0	29.3	19.4	0.0	25.8	−0.5	30.1	2.8
	10	33.2	51.7	82.1	33.2	0.0	44.6	−13.9	50.7	−38.2
	Avg.	30.0	44.7	66.8	30.0	0.0	40.1	−9.0	46.1	−31.0
15 × 5	1	36.9	60.5	105.5	36.9	0.0	49.3	−18.6	57.0	−45.9
	2	31.0	47.3	76.6	31.0	0.0	41.3	−12.6	48.7	−36.5
	3	35.7	55.1	88.5	35.7	0.0	47.5	−13.8	56.8	−35.8
	4	36.3	57.5	101.0	36.4	0.1	48.5	−15.6	57.0	−43.6
	5	40.1	63.7	106.0	40.1	0.0	53.6	−15.8	61.3	−42.1
	6	34.0	51.7	84.0	34.0	0.0	45.5	−12.1	53.0	−36.9
	7	32.8	47.9	71.2	32.8	0.0	43.7	−8.9	52.2	−26.8
	8	34.4	54.5	94.5	34.4	0.0	45.7	−16.2	53.9	−43.0
	9	39.0	60.5	102.8	39.0	0.0	52.3	−13.6	59.2	−42.4
	10	36.0	56.7	92.4	36.0	0.0	48.1	−15.2	56.3	−39.0
	Avg.	35.6	55.6	92.3	35.6	0.0	47.5	−14.2	55.5	−39.8

Note: Bold face indicates optimized value.

problem size. However, the reduction rates for individual instances varied significantly even within the same problem group. For problem groups 5 × 5, 10 × 5, and 15 × 5, the reduction rates were in the ranges of [0.0%, 6.0%], [0.5%, 18.2%], and [8.9%, 18.6%], respectively.

The CO₂-oriented and distance-oriented solutions had virtually identical travel distances. Therefore, the observed reductions in the CO₂-oriented solutions can be attributed to the effective scheduling of the vehicle under time dependent traffic conditions. The

Table 5
Constants and coefficients with respect to different gross weight.

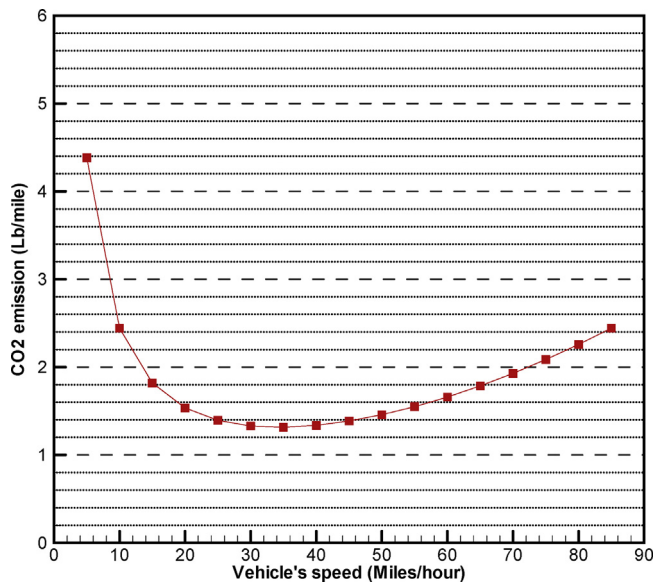
Gross weight (t)	K	A	B	C	D	E	F
3.5–7.5	110	0	0	0.000375	8702	0	0
7.5–16	871	−16.0	0.143	0	0	32,031	0
16–32	765	−7.04	0	0.000632	8334	0	0
32–40	1576	−17.6	0	0.00117	0	36,067	0

Table 6

Examples of detailed results solved by the three-stage approach with relaxation.

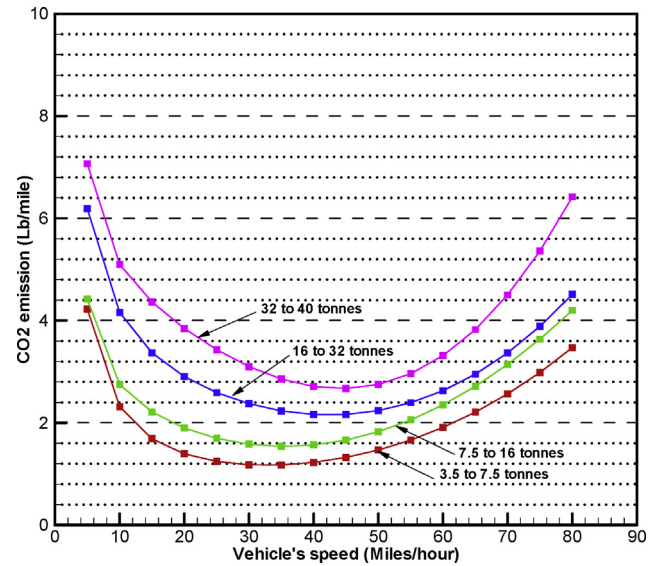
P. class $n \times m$	Relax. λ_1, λ_2	Stage	Objective	TTD	CO ₂ -E	TAR	TOR3	CPU Time (s)
5 × 3	$\lambda_1 = \lambda_2 = 0$	1	CO ₂ -E	45.7	56.8	183.1	84.9	<1
		2	TAR	45.7	56.8	173.9	84.9	<1
		3	TOR3	45.7	56.8	173.9	84.9	<1
	$\lambda_1 = \lambda_2 = 0.1$	1	CO ₂ -E	45.7	56.8	183.1	84.9	<1
		2	TAR	45.7	60.6	140.7	69.2	<1
		3	TOR3	45.7	62.5	154.8	67.7	<1
	$\lambda_1 = \lambda_2 = 0$	1	CO ₂ -E	33.0	45.1	347.4	56.4	2
		2	TAR	33.0	45.1	335.9	56.4	7
		3	TOR3	33.0	45.1	335.9	56.4	4
10 × 5	$\lambda_1 = \lambda_2 = 0.1$	1	CO ₂ -E	33.0	45.1	374.4	56.4	2
		2	TAR	31.8	49.6	310.2	52.4	80
		3	TOR3	31.8	49.6	341.2	50.8	4

Note: Bold faces indicate optimum values.

**Fig. 3.** The relationship between the CO₂ emissions and the vehicle travel speed for a heavy-duty vehicle with 6.35 t gross weight.

CO₂-oriented and distance-oriented solutions had more significant differences in the total travel time. Since the vehicle speeds during the rush hours in an urban area are far below the optimum levels, pursuing lower travel times (i.e., higher speed) would normally result in lower CO₂ emissions. Although a few instances required longer travel times for reducing CO₂ emissions, most of the CO₂-oriented solutions resulted in decreased travel times. On the average, for problem groups 5 × 5, 10 × 5, and 15 × 5, the travel time changes were −6.2%, −26.6%, and −39.2%, respectively. It should be noted that this observation does not necessarily mean that shorter travel times will always lead to lower CO₂ emissions. On the contrary, simulation experiments with more general cases, which are summarized in Tables 6 and 7, showed that TAR-oriented solutions might also result in a sharp increase in CO₂ emissions.

We generated another set of 240 problem instances to simulate a more general case of time-dependent road conditions and different levels of vehicle weight. The customers were located randomly in a 10 × 10 or 20 × 20 mile region, where the travel speeds on roads were subject to different time-dependent traffic patterns as shown in Fig. 1. Problem instances were generated with number of customers with $n = \{5, 10, 15\}$ and the number of periods $m = \{3, 5\}$. In addition, four levels of vehicle's gross weight (with different CO₂ emission rates) were simulated. Ten instances were randomly generated for each combination of the number of

**Fig. 4.** Relationships between CO₂ emission rates and travel speeds of heavy-duty vehicles.

customers, the number of periods, and the vehicle gross weight, resulting in $3 \times 2 \times 4 \times 10 = 240$ test problem instances. The due-times of customers were randomly generated within the planning horizon. The tardiness penalty weights were randomly generated in interval [5,8] to ensure customers being served in their due-times as much as possible. Each arc (i, j) was randomly associated with a speed pattern in Fig. 1.

Hickman [19] developed a regression function of CO₂ emissions based on the speed of the vehicle and considered the vehicle's gross weight as follows:

$$e = K + Av + Bv^2 + Cv^3 + \frac{D}{v} + \frac{E}{v^2} + \frac{F}{v^3}, \quad (6)$$

where e is the rate of emissions in g/km for an unloaded vehicle, or for a bus or a coach carrying an average load, on a road with a slope of 0%, and v is the mean speed of the vehicle in km/h. Hickman [19] also provided the constants and coefficients of the function for the four ranges of gross weight of heavy-duty vehicles as listed in Table 5. Fig. 4 illustrates the relationship between CO₂ emissions (Lb/mile) and vehicle's travel speed (mile/h) for the four types of vehicle gross weights. These fitted curves were used to estimate the amount of CO₂ emissions under different speeds and different vehicles.

The 240 instances were also optimally solved using the MILP formulation. We use the No.1 instance of 5 × 3 group and No.1 instance

Table 7
Same customers visited by vehicle with different gross weights ($\lambda_1 = \lambda_2 = 1$).

Ins. no.	Gross weight (t)	Distance-oriented		CO ₂ -oriented						Route
		TTD	CO ₂ -E	TTD	Δ TTD%	CO ₂ -E	Δ CO ₂ %	TAR	TOR3	
1	3.5–7.5	39.8	84.2	45.7	14.8	56.8	–32.5	173.9	84.9	0→1→3→5→4→2→0
2	7.5–16	39.8	105.4	45.7	14.8	74.1	–29.7	173.9	84.9	0→1→3→5→4→2→0
3	16–32	39.8	124.7	40.4	1.5	101.2	–18.8	447.2	50.5	0→2→4→3→1→5→0
4	32–40	39.8	164.6	40.4	1.5	129.2	–21.5	447.2	50.5	0→2→4→3→1→5→0
Avg.					8.2		–25.6			

Note: Bold faces indicate optimum values.

of 10 × 5 group to demonstrate the three-stage MILP approach. The results are given in Table 6. First, λ_1 and λ_2 were set to zero (without relaxations), and the three objectives CO₂-E, TAR and TOR3 were optimized hierarchically by the three-stage MILP approach. It can be observed that the optimal objective value of each stage remained the same to the end. Therefore, the final solution is a Pareto optimal solution in terms of minimum CO₂-E, TAR and TOR3 objectives. Next, λ_1 and λ_2 were set to 0.1, i.e. additional 10% amount of CO₂ emissions was allowed in order to seek better TAR, and additional 10% TAR was sacrificed to seek better TOR3. It can be observed that the objective value of CO₂-E became larger (below 10%) in the second and third stages but the objectives TAR and TOR3 were improved in comparison to those without relaxations (i.e., $\lambda_1 = \lambda_2 = 0$). In Table 7, we show the results of the identical instance (i.e., No.1 instance of 5 × 3 group) with the same customers served by different gross weight vehicles. Column TTD and CO₂-E represent the traveled distance and the amount of CO₂ emissions, respectively. Columns Δ TTD% and Δ CO₂% indicate the percent deviations in the total travel distance and CO₂ emissions from the distance-oriented optimal solutions, respectively. The distance deviation increased by 8.2% on the average but the CO₂ emissions declined by 25.6% on the average with a maximum value of 32.5%. It can also be observed that in terms of minimizing CO₂ emissions, vehicles with different gross weights may yield different optimal routes for the same set of customers.

In Table 8, the optimal solutions found by the three-stage hierarchical MILP approach, and the distance-oriented results for 240 instances with the four-vehicle gross weight types are compared with respect to CO₂ emissions reduction (%), the rate of travel time change (%), and the rate of travel distance change (%). In terms of CO₂ emissions reduction, all problem groups had a reduction of 12% on the average compared to the distance-oriented solutions. For individual instances, the reduction varied even within the same group, ranging from 0% to 39.3%. Generally, the amount of CO₂ emissions reduction depended on the problem parameters such as customers' location, due-time, and roads' speed changing patterns. The problem size had limited effect on the average CO₂ emissions reduction. A larger vehicle gross weight led to a lower rate of CO₂ emissions reduction, but the total reduction amount was increased. If all instances are considered together, the CO₂-oriented MILP model provided 12.0% CO₂ emissions reduction compared to the distance-oriented approach. These results justify the significance of using the CO₂-oriented model under time-dependent traffic congestions. For problem instance 5 × 5 with 32–40(t) vehicle weight, the distance-oriented approach yielded solutions with longer travel distances. The reason for this abnormality is that the tardiness penalty of this instance is relatively larger in distance-oriented solutions than in CO₂-oriented solutions. The travel distance change (%) column shows the rate of increased travel distance compared to the distance-oriented solutions. In total, 156 instances (65%) out of the 240 instances had longer travel distances to attain lower levels of CO₂ emissions, and for the other 84 instances, the optimum routes to lower CO₂ emissions were also the shortest distance routes with optimized period

selections. On the average, the traveled distance increased by 3.2% for all instances. The Travel Time Change (%) column in Table 8 shows the travel time difference between the CO₂-oriented and distance-oriented solutions. Generally, 115 of the 240 instances (48%) had an increased travel time (while the others had reduced travel time) to achieve lower CO₂ emissions. It is because different vehicles may have different economic travel speed levels, and traveling with a speed closer to the economic level will help to lower CO₂ emissions. The comparisons in Table 8 support the merits of the proposed CO₂-oriented model. First of all, CO₂ emissions cannot be minimized by a pure distance-oriented approach in the presence of time-dependent traffic congestion. Even the CO₂-oriented and distance oriented approach yield the same optimal route, CO₂ emissions can be reduced by optimizing the schedule of the service on the route. These results support the main contribution of the paper, i.e. vehicle routing and scheduling should be considered together to reduce CO₂ emissions under time-dependent traffic conditions.

In Table 9, CO₂-oriented solutions and travel-time-oriented solutions are compared with respect to the three performance metrics, CO₂ reduction (%), travel time change (%), and travel distance change (%). It can be observed that the CO₂ reductions reached an average of 28% for all problem groups, ranging from 2.4% to 50.1%. Clearly, the CO₂ reductions were more significant compared to the distance-oriented solutions in Table 8. This observation may imply that the route with the shortest travel time (with higher speed) leads to emit more CO₂ (42% on the average) than the optimal amount emitted under the CO₂-oriented model. Because the heavy-duty vehicles in this experiment have economic speed levels of 30–45 m/h, the travel time under the CO₂-oriented model may have also increased (50% on the average) in comparison to the travel-time-oriented model. In terms of travel distance change, only 34 instances had longer travel distances, and the other 206 instances resulted in shorter. On the average, the CO₂-oriented solutions had 7.2% shorter travel distance than the travel-time-oriented solutions. As the results in Table 9 indicate, there is a clear trade-off between the travel time and CO₂ emissions. As a result, delaying customer service until the time periods with favorable traffic patterns can reduce CO₂ emissions.

Finally, we tested the computational efficiency of the MILP model by solving different sizes of problem instances. The CPU times reported in Table 10 were obtained by CPLEX (version 12.4.0.1) on the same computer using four CPU cores at full extent. We generated new instances with the number of customers from 5 to 20 and randomly located customers in regions of 10 × 10, 15 × 15, and 25 × 25 square miles, respectively. All instances had five time periods of 30 minute. The travel speed patterns were randomly selected among the six patterns given in Fig. 3. The CO₂ emission rate was calculated by Eq. (6) with random vehicle gross weights. For each problem group, 10 instances were generated, resulting in 16 × 3 × 10 = 480 test instances. Table 10 summarizes the CPU times to solve the test instances. It can be observed that the solution time exponentially increased with the problem size. For example, adding even one additional customer resulted in a significant increase in the CPU times of the problem sizes of 13 × 5, 14 × 5, and 15 × 5

Table 8Summarized comparison of CO₂-oriented and distance-oriented solutions.

Group	Gross weight (t)	Ins. Num.	CO ₂ reduction (%).			Travel distance change (%)			Travel time change (%)			
			Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.	
5 × 3	[3.5, 7.5]	10	−33.4	−4.7	−21.0	14.9	0.0	3.6	125.5	−19.3	28.3	
	[7.5, 16]	10	−30.4	−3.0	−18.9	14.9	0.0	3.6	102.2	−19.3	26.0	
	[16, 32]	10	−19.7	−1.4	−11.9	6.7	0.0	1.7	47.5	−32.3	3.0	
	[32, 40]	10	−22.9	−1.5	−13.1	6.7	0.0	1.7	47.5	−32.3	3.0	
5 × 5	[3.5, 7.5]	10	−28.7	0.0	−8.6	6.1	0.0	0.7	15.5	−14.3	2.5	
	[7.5, 16]	10	−27.0	0.0	−7.9	6.1	0.0	0.7	12.3	−20.3	−0.4	
	[16, 32]	10	−22.2	0.0	−6.5	12.2	0.0	1.3	11.9	−26.3	−8.8	
	[32, 40]	10	−25.8	−0.8	−10.1	12.2	−27.9	−1.5	6.3	−26.3	−11.5	
10 × 3	[3.5, 7.5]	10	−39.3	−14.1	−22.8	13.7	0.0	4.5	41.8	−11.7	12.7	
	[7.5, 16]	10	−36.8	−12.3	−20.6	13.7	0.0	4.0	41.8	−11.7	13.2	
	[16, 32]	10	−22.1	−9.0	−14.8	13.0	0.0	2.5	32.8	−42.2	−11.5	
	[32, 40]	10	−24.8	−10.3	−16.2	13.1	0.0	3.3	32.8	−44.5	−10.6	
10 × 5	[3.5, 7.5]	10	−22.5	−1.3	−10.2	18.0	0.0	7.3	54.9	−12.8	14.4	
	[7.5, 16]	10	−19.8	−2.1	−8.3	18.0	0.0	7.4	47.0	−12.8	11.9	
	[16, 32]	10	−17.6	−0.9	−5.7	6.2	0.0	0.9	11.0	−33.6	−8.6	
	[32, 40]	10	−18.3	−1.0	−6.4	8.0	0.0	1.8	13.8	−33.6	−9.5	
15 × 3	[3.5, 7.5]	10	−28.6	−0.8	−13.4	10.9	0.6	6.0	74.1	−13.1	14.6	
	[7.5, 16]	10	−26.3	−0.9	−11.7	10.9	0.0	5.0	74.1	−13.1	14.4	
	[16, 32]	10	−16.6	−1.7	−7.5	9.5	0.0	3.2	30.9	−42.1	−10.4	
	[32, 40]	10	−17.5	−2.1	−8.2	7.7	0.0	3.3	22.9	−42.1	−11.4	
15 × 5	[3.5, 7.5]	10	−21.8	−7.8	−14.9	12.6	0.2	5.0	58.6	−18.5	9.6	
	[7.5, 16]	10	−19.8	−6.1	−12.3	9.9	0.2	3.8	21.6	−28.4	−0.6	
	[16, 32]	10	−13.7	−1.6	−8.3	5.6	0.0	2.0	3.7	−28.4	−12.5	
	[32, 40]	10	−15.9	−3.0	−9.3	12.0	0.0	4.0	0.6	−38.4	−13.6	
Avg.					−12.0				3.2			1.8

under 15×15 square mile. Notably, the size of the service region had also an effect on the computational efficiency because of the continuous variables of the MILP model. When the service region increased from 10×10 miles, 15×15 miles, to 25×25 miles, the

CPU time also increased and instances larger than 13×5 could not be solved in reasonable CPU times. Moreover, individual problem instances of the same group may have also required very different CPU times as seen in Table 10.

Table 9Summarized comparison of CO₂-oriented and travel-time-oriented solutions (TOR3).

Class	Gross weight (t)	Ins. Num.	CO ₂ Reduction (%).			Travel distance change (%)			Travel time change (%)		
$n \times m$			Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.
5×3	[3.5, 7.5]	10	−43.9	−23.8	−34.4	3.3	−12.1	−2.2	137.2	29.1	81.4
	[7.5, 16]	10	−41.2	−20.3	−31.3	3.3	−12.1	−2.2	112.7	29.1	78.9
	[16, 32]	10	−30.6	−5.5	−17.6	4.1	−12.1	−4.0	84.9	20.6	47.2
	[32, 40]	10	−33.0	−4.4	−18.4	4.1	−12.1	−4.0	84.9	20.6	47.2
5×5	[3.5, 7.5]	10	−41.6	−8.6	−27.1	0.6	−14.0	−4.5	138.3	6.0	39.8
	[7.5, 16]	10	−39.3	−7.5	−25.1	0.6	−14.0	−4.5	138.3	6.0	35.9
	[16, 32]	10	−29.6	−2.4	−16.5	6.3	−14.0	−4.0	56.5	6.0	21.3
	[32, 40]	10	−32.5	−6.3	−20.6	6.3	−27.9	−6.8	56.5	−13.0	18.3
10×3	[3.5, 7.5]	10	−45.3	−18.4	−32.0	8.8	−15.0	−3.1	122.2	50.1	81.7
	[7.5, 16]	10	−42.4	−15.9	−29.0	8.8	−15.0	−3.6	122.2	50.1	82.6
	[16, 32]	10	−29.8	−5.2	−16.0	2.4	−20.7	−4.9	64.9	5.6	38.8
	[32, 40]	10	−29.8	−5.8	−16.9	8.3	−20.7	−4.2	64.2	19.4	40.2
10×5	[3.5, 7.5]	10	−43.4	−30.4	−36.1	6.4	−11.2	−2.2	100.7	43.5	68.5
	[7.5, 16]	10	−40.9	−28.3	−33.2	6.4	−11.2	−2.1	100.7	41.5	65.4
	[16, 32]	10	−29.6	−15.2	−21.8	−4.2	−14.0	−8.1	59.6	11.3	35.4
	[32, 40]	10	−31.4	−16.0	−23.8	−1.0	−14.0	−7.3	59.6	8.0	33.8
15×3	[3.5, 7.5]	10	−49.9	−34.8	−40.9	−3.4	−19.0	−12.3	144.1	40.0	75.2
	[7.5, 16]	10	−47.4	−32.3	−38.2	−3.4	−20.4	−13.1	141.2	40.8	74.9
	[16, 32]	10	−34.0	−19.2	−25.7	−4.8	−24.3	−14.5	73.0	10.3	36.6
	[32, 40]	10	−34.6	−20.8	−26.7	−4.8	−21.1	−14.5	73.0	10.3	35.3
15×5	[3.5, 7.5]	10	−50.1	−33.8	−42.5	−3.3	−18.8	−11.7	94.9	38.9	66.5
	[7.5, 16]	10	−47.6	−30.2	−39.7	−4.2	−18.8	−12.7	94.9	27.1	51.4
	[16, 32]	10	−36.2	−18.2	−28.0	−6.7	−19.5	−14.3	47.1	15.3	33.8
	[32, 40]	10	−39.2	−19.2	−30.0	−4.2	−19.5	−12.6	47.1	15.3	31.7
Avg.					−28.0	−7.2			50.9		

Table 10
Computational efficiencies of MILP model for different problem classes.

P.	Ins.	10 × 10 square mile			15 × 15 square mile			25 × 25 square mile		
		Max.	Min.	Avg.	Max.	Min.	Avg.	Max.	Min.	Avg.
5 × 5	10	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
6 × 5	10	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s
7 × 5	10	<1 s	<1 s	<1 s	<1 s	<1 s	<1 s	7.4 s	<1 s	1.4 s
8 × 5	10	<1 s	<1 s	<1 s	9.2 s	<1 s	1.8	11 s	<1 s	2.9 s
9 × 5	10	<1 s	<1 s	<1 s	3.4 s	<1 s	1.4	27 m	<1 s	6.1 s
10 × 5	10	8.9 s	<1 s	1.6 s	6.9 m	1 s	59 s	13 m	1.0 s	2.4 m
11 × 5	10	104 s	<1 s	19 s	12 m	2 s	125 s	12 m	5.8 s	3.0 m
12 × 5	10	32 s	1.6 s	9.8 s	2.3 m	3 s	44.8 s	67 m	3.0 s	7.6 m
13 × 5	10	6.9 m	0.9 s	62 s	1.6 h	35 s	14.6 m	>10 h	–	–
14 × 5	10	3.8 m	4.4 s	88 s	10.2 h	7.2 s	1.4 h	–	–	–
15 × 5	10	40 m	4.9 s	4.7 m	>24 h	–	–	–	–	–
16 × 5	10	41 m	2.4 s	5.7 m	–	–	–	–	–	–
17 × 5	10	56 m	21 s	8.2 m	–	–	–	–	–	–
18 × 5	10	2.2 h	1.8 m	31 m	–	–	–	–	–	–
19 × 5	10	4.2 h	2.0 m	1.1 h	–	–	–	–	–	–
20 × 5	10	>24 h	–	–	–	–	–	–	–	–

Note: The 4 cores of the CPU were fully used by CPLEX.

4. Solving large-sized GVRSP by an existing meta-heuristics algorithm

Theoretically, the proposed MILP model can be optimally solved only for small-sized problems. The GVRSP is a generalization of the VRP, which is an NP-hard problem. Therefore, solving large-sized GVRSPs optimally with multiple vehicles and a large number of customers is not computationally feasible. However, it is practical to find near-optimal solutions for large-sized GVRSPs by adopting existing meta-heuristic algorithms. In this section, we introduce a Simulated Annealing (SA) algorithm to find good solutions for large-sized problems. In the SA, the continuous variables of the model are not encoded and searched, but determined by a straightforward heuristic procedure that provides near-optimal detailed schedules for a given set of routes (i.e., X_{ij} decision variables) and rough schedules (i.e., x_{ijk} decision variables). This aspect of the SA algorithm ensures the tractability and effectiveness of the solution approach.

4.1. Solution representation

Let string vector $S = \{s_p\}$ represent the sequence in which nodes (customer IDs plus depot 0) are visited, and binary matrix $\mathbf{X} = \{x_{pk}\}$ represent the time period when the arcs in the sequence are traveled. Additional notation is as follows:

- p : position index of the string vector
- s_p : the index of the p th visited node, that is $p = 1, 2, \dots, n, n + 1$, such that the route starts from and ends at the depot ($s_1 = s_{n+1} = 0$)
- x_{pk} : binary variable indicating if arc (s_p, s_{p+1}) is traveled in time period k or not

Fig. 5 illustrates an example of the solution encoding for a small problem with nine customers visited by a single vehicle in five time

periods. The vehicle starts from the depot (represented as 0), visits customer 5 first, and then customers 3, 4, and so forth. After visiting customer 7, the vehicle returns back to the depot. Binary matrix \mathbf{X} represents the travel time periods of the corresponding arcs. If $x_{pk} = 1$ then arc $(p, p + 1)$ uses time period k to travel; otherwise, time period k is not used by arc $(p, p + 1)$. It can be observed that cells with 1 in matrix \mathbf{X} form a route and a schedule path starting from the left-top corner toward the right-bottom corner, and more paths will appear when multiple vehicles are used. This representation is referred to as an *R&S map*.

A feasible R&S map (\mathbf{S}, \mathbf{X}) must satisfy the following Constraints (22)–(25). Constraint (22) ensures that all customers must be visited only once, starting from and ending at the depot. Constraint (23) ensures arc (s_p, s_{p+1}) must be traveled in at least one time period. Constraint (24) ensures that if $x_{pk} = 1$, the left-bottom corner of \mathbf{X} (namely, $p' > p$ and $k' < k$) must be zero, and Constraint (25) ensures that if $x_{pk} = 1$, the right-top corner of \mathbf{X} (namely, $p' < p$ and $k' > k$) must be zero. A feasible R&S map forms a single path traveling through \mathbf{X} from the left-top corner toward the right-bottom corner. Note that the path does not necessarily start at the first period and end at the last period. The path must also be contiguous in the columns but may be noncontiguous in the rows.

$$S \cap N = N, s_1 = 0, s_{n+1} = 0 \tag{22}$$

$$\sum_{k=1}^m x_{pk} \geq 1 \quad \forall p = 1, 2, \dots, n \tag{23}$$

$$1 - x_{pk} \geq x_{p'k'}, \quad \forall p, k, p' > p, \quad k' < k \tag{24}$$

$$1 - x_{pk} \geq x_{p'k'}, \quad \forall p, k, p' < p, \quad k' > k \tag{25}$$

The continuous variables are defined as follows.

- d_{pk} : distance of arc (s_p, s_{p+1}) traveled in period k
- τ_{pk} : time of arc (s_p, s_{p+1}) traveled in period k
- l_p : starting time of traveling arc (s_p, s_{p+1})
- a_p : ending time of traveling arc (s_p, s_{p+1})

In the R&S representation, the total time spent to traverse arcs in a time period can be longer than the length of that time period. Therefore, a penalty term is added to the objective function to

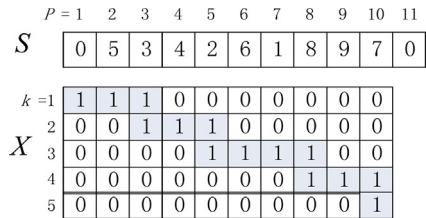


Fig. 5. The R&S map for a solution example.

penalize any overuse of the time periods as follows:

$$\text{Min. CO}_2\text{-E} = \sum_{p=1}^n \sum_{k=1}^m d_{pk} \cdot c_{s_p, s_{p+1}, k} + \sum_{i=0}^n \omega_i \cdot L_i + \beta \cdot \sum_{k=1}^m \max \left\{ 0, Z_k - \sum_{p=1}^n \tau_{pk} \right\}, \quad (7)$$

where β represents the penalty term for any overuse of a time period. Note that the objective function in Eq. (7) is for minimizing CO₂ emissions. The TAR or TOR objectives can be also used.

In the following, we present tight bounds for the continuous variables for given \mathbf{S} and \mathbf{X} . Eqs. (8)–(10) are designed to assign near-optimal values to d_{pk} , τ_{pk} , and l_p as follows:

$$d_{pk} = \begin{cases} x_{pk} \cdot \min \left\{ D_{s_p, s_{p+1}} - \sum_{k'=1}^{k-1} d_{pk'}, \left(Z_k - \sum_{p'=1}^{p-1} \tau_{p'k} \right) \cdot v_{s_p, s_{p+1}, k} \right\}, & \text{if } \sum_{k'=k+1}^m x_{pk'} > 0 \\ x_{pk} \cdot \left(D_{s_p, s_{p+1}} - \sum_{k'=1}^{k-1} d_{pk'} \right), & \text{if } \sum_{k'=k+1}^m x_{pk'} = 0 \end{cases}, \quad \forall p, k \quad (8)$$

$$\tau_{pk} = \frac{60 \cdot d_{pk}}{v_{s_p, s_{p+1}, k}} \quad (9)$$

$$l_p = \begin{cases} 0 & \text{if } p = 0 \\ \max\{a_{p-1}, b_{k'}\} & \text{if } p > 0, \quad k' = \min\{k | k = 1, 2, \dots, m; d_{pk} > 0\}, \quad \forall p \end{cases} \quad (10)$$

Eq. (8) determines the distance traveled on arc (s_p, s_{p+1}) in period k . If $x_{pk} = 0$, then d_{pk} is assigned to zero; otherwise, d_{pk} is assigned to the minimum of the *untraveled-distance* of arc (s_p, s_{p+1}) before period k , calculated by $D_{s_p, s_{p+1}} - \sum_{k'=1}^{k-1} d_{pk'}$, and the *can-travel-distance* of arc (s_p, s_{p+1}) in period k which is calculated by $(Z_k - \sum_{p'=1}^{p-1} \tau_{p'k}) \cdot v_{s_p, s_{p+1}, k}$. However, if period k is the last period that arc (s_p, s_{p+1}) can be traveled (indicated by $\sum_{k'=k+1}^m x_{pk'} = 1$), then all *untraveled-distance* must be assigned to d_{pk} in order to ensure the arc is fully traveled. This way of assignment may lead to an overuse of period k . Any overused time period is penalized in the objective function as given in Eq. (7). In the computer implementation of the SA, d_{pk} and τ_{pk} are calculated mutually by Eqs. (8) and (9) as p increasing from 1 to n .

The starting and ending time of traveling arc (s_p, s_{p+1}) , i.e., l_p and a_p , are determined by the early starting principle. Eq. (10) is used to determine l_p based on a_{p-1} and $b_{k'}$, where k' is the first period selected to travel arc (s_p, s_{p+1}) . To determine a_p , however, we first need to introduce a temporary variable a_{pk} representing the ending time for traveling d_{pk} which is calculated by Eq. (11), where k' is the first period selected to travel arc (s_p, s_{p+1}) . Then a_p can be determined by Eq. (12).

$$a_{pk} = \begin{cases} \max\{l_p, b_k\} + t_{pk} & \text{if } k = k' \\ \max\{a_{p, k-1}, b_k\} + t_{pk} & \text{if } k > k' \end{cases}, \quad \forall p, k, d_{pk} > 0 \quad (11)$$

$$a_p = \max\{a_{pk}\} \quad \forall p, k, d_{pk} > 0 \quad (12)$$

Thus, once d_{pk} is fixed, the starting time l_p and ending time a_p of arc (s_p, s_{p+1}) can be mutually calculated through Eqs. (10)–(12) by increasing p from 1 to n . The process to determine a_p is given in Fig. 6.

Note that Eq. (8) may assign non-optimal values to d_{pk} for those arcs that require more than one period to travel. For example, when the vehicle travels arc (s_p, s_{p+1}) in periods k and $k+1$, i.e., $x_{pk} = 1$, and $x_{p, k+1} = 1$, Eq. (8) will require the vehicle to make full use of period k first, and then use period $k+1$ to travel the remaining distance. This assignment is not optimal because traveling in period k may not be superior to traveling in period $k+1$ in terms of minimizing

the objective function. Nevertheless, for those arcs that have only one period to travel, Eq. (8) will be optimal because it has to assign the arc's whole distance to d_{pk} in order to complete the tour. Fortunately, there are only at most $m-1$ arcs that are possible to have multiple periods, and all other arcs will be assigned with optimal values to d_{pk} . For the arcs that may use multiple periods, an *Upper Bound* is developed to estimate the maximum gap in the objective values between the optimal and near-optimal d_{pk} as follows:

Upper Bound:

$$UB = E^* + L^* \quad (13)$$

The *Upper Bound* consists of three parts: E^* - total CO₂ emissions, L^* - tardiness penalty, and O^* - penalties for over using time periods, which are calculated as

$$E^* = \sum_{p=1}^n e_p^* \text{ and } e_p^* = \max\{D_{s_p, s_{p+1}} \cdot c_{s_p, s_{p+1}, k} | x_{pk} = 1\} - \min\{D_{s_p, s_{p+1}} \cdot c_{s_p, s_{p+1}, k} | x_{pk} = 1\} \quad (14)$$

$$T^* = \sum_{p=1}^n t_p^* \text{ and } t_p^* = \max\{D_{s_p, s_{p+1}} / v_{s_p, s_{p+1}, k} | x_{pk} = 1\} - \min\{D_{s_p, s_{p+1}} / v_{s_p, s_{p+1}, k} | x_{pk} = 1\} \quad (15)$$

$$L^* = \sum_{i=1}^n (L_i - \omega_i \cdot \max\{-T^* + a_i - T_i, 0\}) \quad (16)$$

$$O^* = \min\{O, \beta \cdot T^*\} \quad (17)$$

In the equations above, e_p^* is the maximum gap of CO₂ emissions for arc $(p, p+1)$ when it is traveled in time periods with the highest and the lowest CO₂ emissions rate. Particularly, if arc $(p, p+1)$ has only one time period to travel, then $e_p^* = 0$. Similarly, t_p^* is the maximum gap of travel time on arc $(p, p+1)$ when it is traveled in time periods with the highest and the lowest speed, and T^* is the sum of t_p^* . Thus, L^* and O^* can consequently be estimated

```

1) For k=1 to m Do
2)   If dpk > 0 Then
3)     IF k is the first period that satisfies dpk>0 then
4)       ap ← max{lp, sk} + tpk
5)     Else
6)       ap ← max{ap, sk} + tpk
7)   End If
8) End

```

Fig. 6. The procedure for determining a_p using Eqs. (11) and (12).


```

1) For each arc  $(p, p+1)$  that are assigned to multiple time periods Do
2)   For each pair of  $(k_1, k_2)$  that satisfies  $x_{pk_1}=1$ ,  $x_{pk_2}=1$ ,  $d_{pk_2}>0$ , and  $r_{k_1}<1$  Do
3)     Calculate the rate that the objective decreases if  $\Delta$  distance is transferred from  $d_{pk_1}$  to
          $d_{pk_2}$ , and find  $p^*$ ,  $k_1^*$  and  $k_2^*$  that lead to the maximum decreasing rate.
4)   End
5) End
6) If  $p^*$ ,  $k_1^*$ , and  $k_2^*$  exist Then
7)   Let  $d_{p^*k_1^*} \leftarrow d_{p^*k_1^*} - \Delta$ ,  $d_{p^*k_2^*} \leftarrow d_{p^*k_2^*} + \Delta$ , and recalculate the objective value.
8)   Go to Line 1).
9) End
10) Return

```

Fig. 7. An objective-oriented detailed scheduling procedure for arcs used in multiple periods.

based on T^* , which are the *can-decreased* tardiness penalty and the *can-decreased* over-length penalty, respectively. It is notable that L^* and O^* should not exceed the existing tardiness penalties (noted by L_i) and over-length penalties (noted by O) of the current solution. The near-optimal solution can be further improved by the following optimization procedure in Fig. 7, which tries distance reassignments within the arcs that have multiple periods to decrease the objective function value as much as possible.

For a pair of given S and X , let f^* represent the optimal objective value (which is unknown), f represent the near-optimal objective value obtained by Eqs. (8)–(12), and f' be the objective function value optimized by the objective-oriented procedure depicted in Fig. 7. Thus, we have $f - f' \leq f^* \leq UP$. Particularly, if f does not contain any penalties (i.e., $L_i = 0$ and $O = 0$), we can remove the penalty terms from the objective function and use the procedure in Fig. 7 to obtain a new objective value, noted as f' . Thereby, we have the following Proposition to determine whether f is optimal for (S, X) or is still a near-optimal solution within the *Upper Bound* defined in Eq. (13).

Proposition. If $f = f'$ holds, then $f = f^*$ will hold; otherwise $f \geq f^*$.

Proof. Let $g_p = \sum_{k=1}^m x_{pk}$ indicate the number of time periods used by arc (s_p, s_{p+1}) . For the arcs with $g_p = 1$, Eq. (8) can be simplified as $d_{pk} = x_{pk} \cdot D_{s_p, s_{p+1}}$, which is optimal because the whole distance of the arc has to be traveled in the only one period. For the other arcs with $g_p > 1$, if the objective function does not contain any penalties,

then the arcs contribute only CO₂ emissions to the objective and their contributions are independent from each other. If those arcs with $g_p > 1$ can also be optimized independently without adding new penalties to the objective function value (indicated by $f = f'$ after removing the penalty terms from the objective function to check whether arcs are independent or not), then f' is the optimal objective value for given S and X .

Nevertheless, if the objective value contains penalties, then the procedure in Fig. 7 will not guarantee an optimal objective value, as the penalties may be non-linear. The Upper Bound calculated in Eq. (13) is still the maximum possible gap between f and f^* .

4.2. A basic Simulated Annealing (SA) algorithm for large-sized GVRSP

A canonical SA procedure was used, as shown in Fig. 8, to demonstrate the application of meta-heuristics to solve the large-sized GVRSP with near-optimal solutions. SA is one of most commonly used meta-heuristic algorithms to solve combinatorial optimization problems. SA is a probabilistic algorithm to search for global optima in problems with complex search spaces. At each step of SA, a neighbor solution (candidate solution) is randomly generated from the current solution, and the candidate is accepted as the new current solution with a probability based on the energy value (fitness) of the candidate and current solutions. In SA, the objective is to move from high energy regions to low energy regions in the

```

Procedure SA ( $\tau_{end}$ ,  $\tau$ ,  $N$ )
1) Initialize  $(S, X)$ : Let  $S = \{0, 1, 2, \dots, n, 0\}$ ,  $x_{pk}=1$  ( $\forall k=1$ ) and  $x_{pk}=0$  ( $\forall k>1$ )
2) Heating: repeat to change randomly  $S$  or  $X$  to a neighboring status for  $10n$  times,
   take the maximum objective deviation of two neighboring solutions as the
   initial temperature, i.e.,  $\tau_0 = \max\{\Delta H\}$ 
3)  $\tau = \tau_0$ 
4) While ( $\tau > \tau_{end}$ ) do begin
5)   For  $l=1$  to  $N$  do begin
6)     Select  $S$  or  $X$  uniformly and randomly to generate a candidate solution
7)     Calculate  $\Delta H = H^{new} - H$ 
8)     Accept the candidate as the new current solution with probability  $\exp(-\Delta H/\tau)$ 
9)     Update the best solution found
10)   End For
11)    $\tau = \alpha \cdot \tau$ 
12) End While
13) Output the best solution.

```

Fig. 8. The procedure of the SA algorithm for solving large-sized GVRSP.

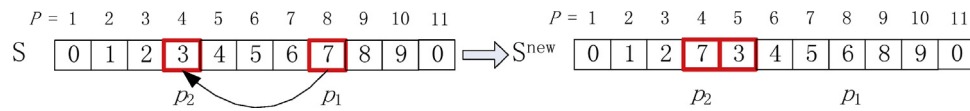
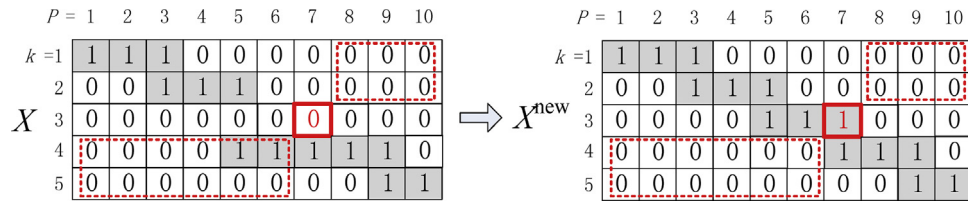
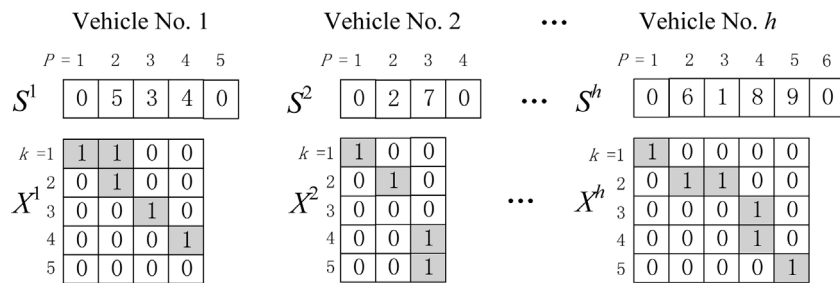
Fig. 9. An insert operation on S .Fig. 10. A reversing operation on X .

Fig. 11. A solution representation example with multiple vehicles.

Table 11
Optimality examination of the SA algorithm on small-sized instances.

Prob.	Instance		Optimal CO ₂ by CPLEX	Near-optimal CO ₂ by SA in 10 runs							
Class	ID	Vehicles		Fs	Avg.	Best	Dev.%	UP	R.%	St.dev.%	CPU time (s)
5 × 5	1	1	72.0	10	72.0	72.0	0	12.4	100	0	2.5
	2	1	108.3	10	108.3	108.3	0	32.7	100	0	2.4
	3	1	122.6	10	124.7	124.7	1.7	4.0	0	0	2.3
	4	1	54.3	10	54.3	54.3	0	10.8	100	0	2.4
	5	1	64.4	10	64.6	64.6	0.3	18.8	94	0	2.4
	6	1	47.7	10	48.3	48.3	1.2	3.5	0	0	2.8
	7	1	46.5	10	46.5	46.5	0	0.1	–	0	2.5
	8	1	52.5	10	52.5	52.5	0	16.7	100	0	2.8
	9	1	67.8	10	67.8	67.8	0	23.2	100	0	2.8
	10	1	65.5	10	69.9	69.9	6.3	12.3	41	0	2.7
Avg.			70.2		70.9	70.9	1.0	13.5	71	0	
10 × 5	1	1	45.1	10	45.2	45.1	0	2.8		0.69	8.2
	2	1	47.3	10	47.3	47.3	0	3.2	100	0	8.5
	3	1	50.0	10	50.4	50.4	0.8	6.2	90	0	8.0
	4	1	46.7	10	47.0	47.0	0.6	4.7	88	0	7.6
	5	1	49.7	10	49.7	49.7	0	8.8	100	0	8.5
	6	1	51.4	10	51.4	51.4	0	11.0	100	0	8.4
	7	1	42.5	10	42.5	42.5	0	4.7	100	0	8.5
	8	1	55.2	10	55.5	55.3	0.2	0.9	50	0.28	7.0
	9	1	45.6	10	45.6	45.6	0	7.8	–	0	7.9
	10	1	37.1	10	37.6	37.6	1.3	4.4	0	0	8.7
Avg.			47.1		47.2	47.2	0.3	5.4	78	0.1	
15 × 5	1	1	44.2	10	44.5	44.2	0	5.0	100	0.95	17
	2	1	45.2	10	45.7	45.7	1.1	0.3	17	0.00	15
	3	1	52.3	10	52.3	52.3	0	3.4	–	0.11	16
	4	1	50.3	10	50.3	50.3	0	3.2	100	0.27	17
	5	1	51.4	10	51.6	51.4	0	2.7	44	0.85	17
	6	1	57.6	10	59.5	59.2	2.7	12.5	79	0.33	17
	7	1	51.2	10	51.2	51.2	0	4.2	100	0.00	16
	8	1	40.4	10	41.7	40.4	0	1.1	0	3.32	15
	9	1	54.3	10	54.3	54.3	0	8.1	100	0.00	17
	10	1	43.5	10	43.5	43.5	0	2.9	100	0.00	17
Avg.			49.0		49.4	49.3	0.4	4.3	71	0.58	

Note: Bold face indicates optimal solutions.

Table 12
Performances of the SA algorithm on medium & large-sized instances.

Prob.	Instance		Near-optimal CO2 by SA				
	ID	Vehicles	Fs	Avg.	Best	St.dev. %	CPU time
30 × 3	1	3	10	339.8	303.5	6.82	22 s
	2	3	10	246.4	202.5	8.36	21 s
	3	3	10	202.9	190.2	5.00	22 s
	4	3	10	194.1	184.4	2.99	22 s
	5	3	10	294.9	288.1	2.93	22 s
	6	3	10	215.0	204.0	2.94	24 s
	7	3	10	352.1	331.2	3.59	22 s
	8	3	10	262.9	256.8	3.43	22 s
	9	3	10	259.6	246.8	3.11	22 s
	10	3	10	279.5	268.9	2.12	22 s
Avg.					247.6	4.13	22 s
30 × 5	1	2	10	152.7	145.8	3.91	32 s
	2	2	10	200.8	194.2	3.40	33 s
	3	2	10	123.6	115.0	3.37	31 s
	4	2	10	214.6	208.6	2.12	33 s
	5	2	10	191.5	187.5	1.63	32 s
	6	2	10	118.5	106.6	8.93	30 s
	7	2	10	192.9	183.7	4.71	32 s
	8	2	10	190.3	171.1	9.68	29 s
	9	2	10	216.2	205.2	5.52	32 s
	10	2	10	130.5	122.3	5.80	32 s
Avg.					164.0	4.68	31 s
50 × 3	1	6	10	443.8	429.0	1.96	1.7 m
	2	6	10	275.7	270.0	2.51	1.7 m
	3	6	10	303.5	277.9	4.56	1.7 m
	4	6	10	291.1	265.2	6.09	1.5 m
	5	6	10	359.9	335.4	3.46	1.6 m
	6	6	10	173.0	165.1	3.14	1.7 m
	7	6	10	199.7	193.0	3.10	1.7 m
	8	6	10	227.5	207.5	6.15	1.7 m
	9	6	10	368.5	356.0	3.93	1.7 m
	10	6	10	170.3	160.8	3.15	1.8 m
Avg.					266.0	3.71	1.7 m
50 × 5	1	4	10	203.7	194.1	3.63	2.6 m
	2	4	10	254.6	243.0	2.25	2.6 m
	3	4	10	177.7	165.0	5.21	2.6 m
	4	4	10	294.5	286.6	2.00	2.6 m
	5	4	10	353.6	329.1	4.01	2.6 m
	6	4	10	212.0	201.9	2.63	2.6 m
	7	4	10	160.4	153.0	3.55	2.5 m
	8	4	10	178.9	169.4	3.21	2.5 m
	9	4	10	252.0	239.9	3.68	2.6 m
	10	4	10	388.0	362.4	4.56	2.3 m
Avg.					234.4	3.48	2.5 m
100 × 10	1	8	1	–	500.9	–	54.1 m
	2	8	1	–	277.7	–	57.0 m
	3	8	1	–	270.1	–	61.3 m
	4	8	1	–	523.5	–	55.1 m
	5	8	1	–	296.6	–	58.9 m
	6	8	1	–	218.2	–	62.5 m
	7	8	1	–	300.5	–	64.5 m
	8	8	1	–	472.9	–	53.2 m
	9	8	1	–	557.9	–	53.7 m
	10	8	1	–	239.2	–	50.0 m
Avg.					365.7		57.0 m

search space. By accepting neighbors that raise the current energy value, SA may avoid being trapped in local optima. The convergence process in SA is facilitated by setting a temperature parameter to a high value in earlier stages of the search (i.e., the probability of accepting inferior solutions is higher) and then slowly reducing it as the search progresses. Being a straightforward algorithm, SA has been applied to many combinatorial problems. The SA algorithm herein have three parameters τ_{end} , α , and N , representing the stopping condition, the cooling rate, and the thermal equilibrium requirement for each temperature level, respectively, to control the trade-off between solution quality and computational efficiency. The procedure of the SA algorithm is given in Fig. 8.

In the SA algorithm, S or X is randomly and uniformly selected to perturb the current solution to generate a new candidate. If S is selected, an operator *insert*, as shown in Fig. 9, is applied to generate a new solution. The insert operator reinserts the customer ID position p_1 into position p_2 by shifting the customers between positions p_1 and p_2 accordingly. A heuristic based on distance ranking [37] is used to select positions p_1 and p_2 . The insert operator with distance ranking has a bias to select positions p_1 and p_2 which include closer customers. After performing insert operation on S , X remains unchanged.

When X is selected to generate a candidate solution, a *flip* operator operates on a randomly selected point (p, k) in X to change

the binary value of x_{pk} as $1 - x_{pk}$. However, changing a point x_{pk} from 1 to 0 may violate Constraint (23); therefore, such changes are not permissible to maintain the feasibility of \mathbf{X} . Meanwhile when a point x_{pk} is changed from 0 to 1, both the right-top zone of x_{pk} and the left-bottom zone of x_{pk} must be filled with zero (or shift the “1” to the same row as x_{pk}), as shown in Fig. 10, in order to comply with Constraints (24) and (25).

To apply the SA algorithm to the GVRSP with multiple vehicles, multiple pairs of decision variables, e.g., (S^h, X^h) $h = 1, 2, \dots, H$, are added to the representation, where h is the vehicle ID and H is the number of available vehicles. The objective function is also updated considering CO₂ emissions of all vehicles and related penalties. Fig. 11 shows a solution example with multiple vehicles. Note that when multiple vehicles are used, the insert operator is applied to one vehicle or two different vehicles selected randomly, and the flip operator is applied to a randomly selected vehicle.

4.3. Computational experiments

The SA algorithm was used to solve problem instances with different sizes in order to study its performance for the GVRSP. First, 30 small-sized instances generated in Section 3.3 were solved to compare the solutions found by the SA algorithm with the optimal solutions found by the MILP formulation. The small-sized instances include 10 instances of each group 5×5 , 10×5 , and 15×5 . Next, 40 medium-sized instances, including 10 instances of each group 30×3 , 30×5 , 50×3 , and 50×5 , were studied. Finally, 10 large-sized instances of group 100×10 were generated and tested. Multiple vehicles were used for these test problem instances to ensure that all deliveries could be completed within the planning horizon. For the instances of small-sized and medium-sized, the SA procedure was run for 10 random replications. For large-sized instances, the SA procedure was run only once. The control parameters of SA were set as $\tau_{\text{end}} = 0.1$, $\tau = 0.99$, and $N = 100 \times n$ for all instances. The SA procedure was coded by Visual C++ 6.0 and run in a PC computer equipped with CPU Intel® Core™ i5-3571 @ 3.4 GHz and MS Window 7 system.

In Table 11, the solutions found by the SA procedure for small-sized instances are compared to the optimal solutions found by AMPL/CPLEX. The tardiness penalty part is not considered for a better comparison (only No.2 instance of group 5×5 has tardiness penalty). Column *Fs* indicates how many feasible solutions were found in 10 runs. Columns *Avg.* and *Best* represent the average solution and the best solution found in 10 runs, respectively. Column *Dev.%* is the percentage deviations of the best solutions from the optimal solutions found by AMPL/CPLEX. Column *UP* is the Upper Bound calculated by Eq. (13). Column *R.%* represents to what extents the optimizing procedure in Fig. 7 adjusted non-optimal solutions toward optimality. *R.%* is calculated as $R.\% = (F' - F)/(F' - F^*) \times 100\%$, where F^* is the optimal objective function value, and F and F' are the objective function values obtained by the SA algorithm with and without using the optimizing procedure, respectively. Column *St.dev. %* is the ratio of the standard deviation of 10 runs to the best solution. It can be observed that the SA procedure found the optimal solution for 21 of the 30 small instances. The optimal solutions were found in 186 (89%) out of 210 random replications. In general, the average deviation from optimal solutions is 1.0% for group 5×5 , 0.3% for group 10×5 and 0.4% for group 15×5 , which are relatively small for real-life applications. The optimizing procedure performed well in adjusting non-optimal solutions toward optimality as 14 of the 30 instances were successfully optimized by the optimizing procedure. The difference between the optimal solutions before and after applying the optimizing procedure (i.e., *R.%* in Table 11) exceeds 70% on the average.

The results for medium and large-sized instances are listed in Table 12. The average and best solutions of 10 runs are

displayed in columns *Avg.* and *Best*, respectively. The computational time also increased considerably with increasing problem size, but the largest problem could be solved less than an hour. Since the GVRSP is NP-hard, the optimal solutions for medium and large-sized instances are difficult to find. The SA provides a basic benchmark results for comparison with other competitive heuristic meta-methods such as Genetic Algorithms, Tabu Search, and Ant Colony Optimization, and Particle Swarm Optimization.

5. Conclusions

We have studied the green vehicle routing and scheduling problem (GRVSP) considering a general case of time-dependent traffic conditions under multiple time periods. In the GRVSP, vehicles need to optimize their travel plans by both arc routing and time scheduling in order to achieve the minimum amount of greenhouse gas emissions and tardiness. A mixed integer linear programming (MILP) model with hierarchical objectives and linear constraints is developed for the GRVSP. The proposed formulation is different from the literature where the schedule of a vehicle is generally determined by the time point when the vehicle departs from each node on its route. In the proposed formulation, the schedule of the vehicle depends on the distance that the vehicle travels on each arc in each period. The main advantage of the proposed formulation is that more general time-dependent traffic patterns can be considered. In addition, the complexity of calculating CO₂ emissions within the mathematical model is eliminated. To achieve these objectives, the average travel speed of a vehicle is assumed to be given and depend on the time period when the vehicle actually travels the arc. Although assuming pre-determined travel step can be considered a drawback of the formulation, in many real-life cases like urban roads, the vehicle speed is not a decision variable, but a parameter dictated by traffic rules and conditions. The proposed formulation is more appropriate for urban transportation networks where the travel speed of a vehicle depends on the road conditions and the time of the day.

The MILP model can either be optimally solved with single objective optimization, such as the total CO₂ emissions, the total arrival time, the total travel distance, and the total travel time, or be solved with hierarchical objectives under Pareto optimality to assist transportation companies in making trade-off decisions according to a carbon trading system in the future.

Since the formulation is linear, it can be solved to optimality by most of the existing MIP solvers like CPLEX. In our experiments, the maximum problem size can be solved by CPLEX in reasonable computational time is 19×5 within a region of 10×10 square miles, which is realistic for small-sized problems in real-life applications. We performed simulation experiments based on two existing models for estimating CO₂ emissions of heavy-duty vehicles with different levels of gross weight and find that the MILP model for the GRVSP can reduce a various range of CO₂ emissions (also for fuel cost) from 0% to 50%, (14.2% on the average) compared to distance-oriented solutions and 28% on the average compared to travel-time-oriented solutions. Compared to the reported CO₂ reduction results of the existing models in the literature (such as [24] by 22.69% and 24.61%, Miguel [15] by 14.5%, [28] by 7%, and [22] by 4.3% on the average with maximum of 11.4%), we obtained slightly higher levels of CO₂ emissions reductions when the CO₂ emissions-based objective was used instead of the traditional distance or time-based objectives. One of the reasons for this result is that we were able to use more complex and more general time-dependent traffic congestion scenarios in the model. Under time-dependent traffic conditions, vehicle routing and scheduling should be considered together to reduce CO₂ emissions, and the

impact of vehicle scheduling decisions on reducing CO₂ emissions becomes more critical with the increasing complexity of traffic congestion patterns.

To solve large size problems, we represented the problem as an R&S map and developed a simulating annealing algorithm to solve medium & large-sized problems with near-optimal solutions. Computational experiments on 30 small-sized problem instances show that over two-third of the GVRSPs can be solved with optimal solutions, with less than 1% of average deviation from the optimal solutions. Medium & large-sized GRVSP instances have been tested by the SA algorithm, providing benchmarks for the future research, which may involve developing and testing other meta-heuristics (single or multi-objective) based-on the modeling framework proposed in this paper. In addition, the model can be expanded by considering other factors such as the changes in the weight of the vehicle and load during the service as a further research topic.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China under Grant no. 71271009.

References

- [1] B.-H. Ahn, J.-Y. Shin, Vehicle-routing with time windows and time-varying congestion, *J. Oper. Res. Soc.* 42 (5) (1991) 393–400.
- [2] M. Barth, K. Boriboonsomsin, Real-world CO₂ impacts of traffic congestion, *Transp. Res. Rec.: J. Trans. Res. Board* 2058 (2008) 163–171.
- [3] M. Barth, T. Younglove, G. Scora, Development of a heavy-duty diesel modal emissions and fuel consumption model. Technical Report. UCB-ITSPRR-2005-1, California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2005.
- [4] T. Bektas, G. Laporte, The pollution-routing problem, *Transp. Res. B* 45 (2011) 1232–1250.
- [5] H.K. Chen, C.F. Hsueh, M.S. Chang, The real-time time-dependent Vehicle Routing Problem, *Transp. Res. E* 42 (2006) 383–408.
- [6] N. Cohen, P. Robbins, *Green Cities: An A-to-Z Guide*, SAGE Publication, Inc., 2011, pp. 55–56.
- [7] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, *Manage. Sci.* 6 (1) (1959) 80–91.
- [8] E. Demir, T. Bektas, G. Laporte, A review of recent research on green road freight transportation, *Eur. J. Oper. Res.* 237 (2014) 775–793.
- [9] E. Demir, T. Bektas, G. Laporte, An adaptive large neighborhood search heuristic for the Pollution-Routing Problem, *Eur. J. Oper. Res.* 223 (2012) 346–359.
- [10] B. Eksioglu, A.V. Vural, A. Reisman, The vehicle routing problem: a taxonomic review, *Comput. Ind. Eng.* 57 (2009) 1472–1483.
- [11] Environmental Protection Agency, Motor Vehicle Emission Simulator (MOVES), 2012, June.
- [12] S. Erdogan, E. Miller-Hooks, A Green Vehicle Routing Problem, *Transp. Res. E* 48 (2012) 100–114.
- [13] E. Ericsson, H. Larsson, K. Brundell-Freij, Optimizing route choice for lowest fuel consumption—potential effects of a new driver support tool, *Transp. Res. C—Emerg. Technol.* 14 (2006) 369–383.
- [14] M.A. Figliozzi, The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics, *Transp. Res. E* 48 (2012) 616–636.
- [15] M.A. Figliozzi, Vehicle routing problem for emissions minimization, *Transp. Res. Rec.: J. Transp. Res. Board* 2197 (2010) 1–7.
- [16] A. Franceschetti, D. Honhon, T.V. Woensel, G. Laporte, T. Bektas, The time-dependent pollution-routing problem, *Transp. Res. B* 56 (2013) 265–293.
- [17] D.R. Gaur, A. Mudgal, R.R. Singh, Routing vehicles to minimize fuel consumption, *Oper. Res. Lett.* 41 (2013) 576–580.
- [18] B.L. Golden, S. Raghavan, E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, Berlin, 2008.
- [19] A.J. Hickman, Methodology for calculating transport emissions and energy consumption, *Transp. Res. Lab.* (1999) 69–73.
- [20] S. Ichoua, M. Gendreau, J.-Y. Potvin, Vehicle dispatching with time-dependent travel times, *Eur. J. Oper. Res.* 144 (2003) 379–396.
- [21] International Road Transport Union, CO Emission, 2012, <http://www.iru.org/en-policy.co2.home> (accessed 16.02.14).
- [22] O. Jabali, T. Woensel, A.G. de Kok, Analysis of travel times and CO₂ emissions in time-dependent vehicle routing, *Prod. Oper. Manage.* 21 (6) (2012) 1060–1074.
- [23] I. Kara, B.Y. Kara, M.K. Yetis, Energy minimizing vehicle routing problem, in: A. Dress, Y. Xu, B. Zhu (Eds.), *Combinatorial Optimization and Applications: FIRST International Conference, LNCS 4616*, Springer, Berlin/Heidelberg, 2007, pp. 62–71.
- [24] Y. Kuo, Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem, *Comput. Ind. Eng.* 59 (2010) 157–165.
- [25] Y.J. Kwon, Y.J. Choi, D.H. Lee, Heterogeneous fixed fleet vehicle routing considering carbon emission, *Transp. Res. D* 23 (2013) 81–89.
- [26] G. Laporte, M. Gendreau, J.Y. Potvin, F. Semet, Classical and modern heuristics for the vehicle routing problem, *Int. Trans. Oper. Res.* 7 (4–5) (2000) 285–300.
- [27] C. Lin, K.L. Choy, G.T.S. Ho, S.H. Chung, H.Y. Lam, Survey of green vehicle routing problem: past and future trends, *Expert Syst. Appl.* 41 (2014) 1118–1138.
- [28] W. Maden, R. Eglese, D. Black, Vehicle routing and scheduling with time-varying data: a case study, *J. Oper. Res. Soc.* 61 (3) (2010) 515–522.
- [29] C. Malandraki, M.S. Daskin, Time-dependent vehicle-routing problems: formulations, properties and heuristic algorithms, *Transp. Sci.* 26 (3) (1992) 185–200.
- [30] Research Papers Download Center in China, 2008, <http://www.studa.net/chengben/080715/09130279-2.html> (accessed 21.02.14).
- [31] B. Sahin, H. Yilmaz, Y. Ust, A.F. Guneri, B. Gulsun, An approach for analysing transportation costs and a case study, *Eur. J. Oper. Res.* 193 (2009) 1–11.
- [32] G. Tavares, Z. Zsigraiova, V. Semiao, M. da Grac, A case study of fuel savings through optimization of MSW transportation routes, *Manage. Environ. Qual.: Int. J.* 19 (4) (2008) 444–454.
- [33] The International Energy Agency, CO₂ emissions from fuel combustion—highlights (2012 Edition), 2012, <https://www.iea.org/co2highlights/co2highlights.pdf> (accessed 16.02.14).
- [34] Thomson Reuters Point Carbon, Value of world's CO₂ markets to rise two-thirds in 2014- Point Carbon, 2014, <http://www.pointcarbon.com/news/reutersnews/14309323> (accessed 21.02.14).
- [35] P. Toth, D. Vigo, An overview of vehicle routing problems, in: P. Toth, D. Vigo (Eds.), *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*, vol. 9, Siam, 2002, pp. 1–26.
- [36] Wikimedia, Kyoto Protocol, 1997, http://en.wikipedia.org/wiki/Kyoto_Protocol (accessed 21.02.14).
- [37] Y. Xiao, Q. Zhao, I. Kaku, M. Nenad, Variable neighborhood simulated annealing algorithm for capacitated vehicle routing problems, *Eng. Opt.* 46 (4) (2014) 562–579.
- [38] Y. Xiao, Q. Zhao, I. Kaku, Y. Xu, Development of a fuel consumption optimization model for the capacitated vehicle routing problem, *Comput. Oper. Res.* 39 (2012) 1419–1431.