Decision Support System for Green Real-life Field Scheduling Problems

Yizi Zhou[1][0000−0003−2140−9801] ✉, Anne Liret[2][0000−0003−0620−7240], Jiyin Liu[1][0000−0002−2752−5398], Emmanuel Ferreyra[3][0000−0003−3381−5380], Rupal Rana[1][0000−0002−9025−0569] and Mathias Kern[2][0000−0003−3734−6542]

[1] Loughborough University, UK
y.zhou2@lboro.ac.uk
[2] British Telecommunications plc, Ipswich, UK
[3] University of Essex, Colchester, UK

Abstract. A decision support system is designed in this paper for supporting the adoption of green logistics within scheduling problems, and applied to real-life services cases. In comparison to other green logistics models, this system deploys time-varying travel speeds instead of a constant speed, which is important for calculating the CO_2 emission accurately. This system adopts widely used instantaneous emission models in literature which can predict second-by-second emissions. The factors influencing emissions in these models are vehicle types, vehicle load and traffic conditions. As vehicle types play an important role in computing the amount of emissions, engineers vehicles number plates are mapped to specified emission formulas. This feature currently is not offered by any commercial software. To visualise the emissions of a planned route, a Heat Map view is proposed. Furthermore, the differences between minimising CO_2 emission compared to minimising travel time are discussed under different scenarios. The field scheduling problem is formulated as a vehicle routing and scheduling problem, which considers CO_2 emissions in the objective function, heterogeneous fleet, time window constraints and skill matching constraints, different from the traditional time-dependent VSRP formulation. In the scheduler, this problem is solved by metaheuristic methods. Three different metaheuristics are compared. They are Tabu search algorithms with random neighbourhood generators and two variants of Variable Neighbourhood search algorithms: variable neighbourhood descent (VND) and reduced variable neighbourhood search (RVNS). Results suggest that RVNS is a good trade-off between solution qualities and computational time for industrial application.

Keywords: Green Logistic Scheduling System, Speed Profile, Metaheuristics Comparison, Heat Map.

1 Introduction

Green logistics has gained increasing awareness from the governments and service provisioning companies, recognizing that the traditional distribution and

production logistics is not sustainable in the long term. However challenges in such transformation are firstly the design of operationally efficient and accurate approach of CO_2 emissions calculation and secondly the smooth adoption of CO_2 awareness in planning decisions across actors. Transportation is a major source of carbon emissions, and the main task in this paper is to design a decision support system that provides engineers guidance on their daily schedule and routine based on the lowest emission or fuel consumption efficient routes. Field scheduling problems aim at scheduling field tasks to the right mobile engineers at the right time. An integrated system of a scheduling and routing engine and a vehicle specified instantaneous emission model is developed. The unique features of the system are: 1) Automated vehicles details mapping to emission formula, which is not provided in any commercial emissions software; 2) tailored speed profiles for different regions in UK, based on real-world traffic data and driving behaviours of engineers data from a field service provider; 3) computing emissions data for every granularity like road segment level, task level, engineer level, day level and so on. Although fuel consumption data gives company an idea of the overall emissions associated with each vehicle and engineer, the emission data gives more detailed information on fuel consumption e.g. per task; 4) Switchable green scheduling and routing simulation engines depending on the users need: a faster solution algorithm (reduced variable neighbourhood search) and another algorithm runs longer but yields better solutions (random neighbourhood Tabu search); 6) Map visualizing (Heat map): the emissions on travels are colored and benchmarked against different levels of CO_2 emissions.

1.1 Green Vehicle Routing Problem

Green vehicle routing is one of the subjects of study. Initial works in this area have primarily focused on reducing the total distance travelled to reduce the total emissions. [1] took a distance-based emission model when solving a green capacitated vehicle routing problem. They used average fuel consumption data per 100km for a heavy-duty truck with various states of load. Fuel consumption is directly related to CO_2 emissions as the fixed carbon content in chemical compound of fuel. A fuel conversion factor (converting to CO_2 emissions) was derived to be 2.61kg CO_2/litre. A linear programming model with CO_2 emission minimisation objective function was built and solved by a Tabu search algorithm. CO_2 emissions are not only influenced by travel distance, but by many other factors, such as road characteristics, vehicle speed and load. According to [2], speed is more important than distance travelled when estimating emissions. During times of congestion, vehicles generate much more emissions than when travelling at free flow speed. It is important to measure time varying travel speed to capture congestion situations. Therefore, time-dependent vehicle routing and scheduling problem (VRSP) emerges. This problem is alike traditional vehicle routing problems (VRP), but with the travelling times between nodes depend on the time of the day, such as peak and off-peak times with different traffic flow speeds. [3] studied a CO_2 emission minimisation vehicle routing problem with fluctuating travel speed. Test cases were generated on a London road map

with real traffic data, including speed data for each road. Also, there are multiple possible road links that connect two customers sites like in real life, so path selection will be part of the decision problem. They treated travel speed as a decision variable, ranging up to the current maximum traffic flow speed on the link travelled. They showed that with path selection and speed optimisation, 2-3% emissions are saved comparing with always using fastest path and traffic flow speed. However, if the speed range emissions level is not very sensitive to speed change, setting travel speed to traffic flow speed will give a good solution as well.

1.2 CO_2 Emission Models

As emissions is affected by travel speed, vehicle types and many other factors, the shortest or quickest route may not be optimal with respect to carbon emissions or fuel consumptions. In this case, green vehicle routing depends on accurate computation of the carbon emissions to generate a route plan that is truly greener than the distance or travel time minimisation [4]. There are various emission models that differ in nature of estimating emission levels and fuel consumptions. They are either a macroscale model that aggregate total emissions and works as a rough estimate or a microscale model that can predict a relatively more accurate second-by-second emission level. A detailed review of existing emission models can be found in [5]. Regardless of emission models applied, CO_2 emissions could be 15-20% higher in real traffic conditions as quoted in [6]. One possible reason is most emission models predict emissions based on hot stabilised engine conditions and do not consider cold engine starts when emissions will be 10% higher. Two emissions models are frequently used in the literature of green vehicle routing problem. The MEET project funded by the European Union aimed to provide a basic, Europe-wide procedure for evaluating the impacts of transport on air pollution [7]. Several models were developed for different types of vehicles with various weight ranges. Those models are microscale level models, which are able to predict second-by-second emission levels. Vehicle emission levels are calculated based on travel speed, vehicle mass and vehicle type, and can be adjusted to road gradient as well as vehicle load. The MEET model is an average of various speed-emission curves of a range of driving cycles. A driving cycle means a specific pattern of driving behaviour containing a different combination of stops, starts, accelerations and decelerations. So, the effect of acceleration is implicitly included. Similar models can be found in [8], developed in the UK by TRL (transportation research lab) and named NAEI. NAEI database are based on many measurements from various programs conducted over year. The database complied as part of MEET project, also included data from TRL.

1.3 Time-Dependent Speed Profile

Both MEET and NAEI models take speed as an input to calculate emissions. According to [4], using fixed speed to compute emissions in emission model can sometimes give less than half of those computed under realistic driving conditions

with fluctuating speed. The author demonstrated the effect of fluctuating speed on the accuracy of carbon emission computations in green vehicle routing. The time-dependent travel speed is necessary. It is modelled by discretizing the day into a number of time buckets (1 hour duration or simply morning, midday and afternoon), and then assigned a unique mean speed value to each bucket. [9] divided a day into 24 and 15 time slots respectively, and travel speeds for each time slot for each road segment are estimated using real-life travel data, assuming in each time slot the traffic condition is stable.

2 Problem Statement

The problem studied is new and has some different features compared to the traditional time-dependent vehicle routing problem. As with tradition models the travel speed is time-dependent, and time-window constraints are considered. The new features are 1) the objective of CO_2 emission minimisation; 2) heterogeneous fleet are used; 3) skill matching constraints are propagated. The vehicle scheduling and routing problem of this study involve real life task locations in UK. $V = v_0, v_1, \cdots, v_n$ is a set of nodes representing task locations and v_0 represents the depot, which is the engineers start work and end work location. $A = \{(v_i, v_j) \,|\, v_i, v_j \in V, i \neq j\}$ is the arch set, and we assume engineers will always take the shortest path between two nodes, so there is only one path linking the same pair of nodes. Each arch of A is associated with a non-negative cost C_{ij}, which represents the cost of traveling from v_i to v_j. In this study, this cost is the amount of CO_2 emissions computed using emissions models mentioned previously. The travel speed v_t is a time-dependent speed, derived from our speed profile. Engineers drive different vehicles which have different emission formulas. Each task has a specified skill requirement and each engineer has a set of skills that he/she is capable of conducting. Task can only be scheduled to an engineer with the specified skill. Each task has a time window requirement $[b_i, e_i]$, where b_i is the earliest possible time to start this task and e_i is the latest time. The service start time must fall in to this range. Each vehicle/engineer has a working period, all travels and works can only be scheduled within this working period. In a summary, the problem is a time-dependent green vehicle routing problem with heterogeneous fleet, time window and skill matching constraints. The components of the green scheduling decision support system for this problem are road distance estimation, tailored speed profiles, vehicle specification mapping to emission formula, and scheduling and routing computation engines. Engines consist of different metaheuristics and during the search only feasible solutions satisfying all constraints described (feasible) are evaluated. The engines will be discussed in detail in the next section.

2.1 Travel Distance Estimation

Given locations (longitude and latitude) of two sites, line distance between two sites can be calculated using Haversine formula, which gives the shortest distance

over the earths surface between two sites ignoring terrain. However, in real life vehicles do need to follow the road construction to travel between two sites. 2000 task pairs were randomly selected and the line distance between each pair of tasks was calculated. Then, Google Map Distance API was called to get the road distance estimate between same pairs of tasks. Assuming there is a linear relationship between line distance and road distance, a linear regression analysis was carried out. Assumed that: $RoadDistance = \beta * LineDistance + error$. Linear regression analysis was run to get the estimate value of parameter β. Taking London as an example, the linear model is significant and with a 98.2% R square value, meaning a high goodness of fit value. P-value of F statistic is $0.00 \leq 0.05$, which shows the model is better than the intercept-only model and $\beta = 1.256$.

2.2 Real-Life Travel Speed Profile

Based on the travel journals of engineers from a field service provider, the real travel speed profile was generated for each working area in the UK. Taking London as an example, the profile will capture the average traffic condition throughout the day. It is expected that there will be two peak-hour periods, where the travel speed is low. The travel speed for the hour that the engineer starts to travel to a task is computed. For each task, we will have a speed entry. A day is divided into 24 one-hour-slots, 0:00-24:00. If an engineer travels in a time slot, the travel speed in that slot will be recorded. Then we took the average speed of that slot over 121 days to construct a speed profile. We used speed data entries for 6 months from London. Figure 1 (left) shows two rush hour periods: earlier morning and late afternoon. This advocates expectation of real life traffic conditions. To reduce the total variance of the data, several time slots are grouped together to smooth the data pattern as shown in Fig. 1 (right). The grouping criteria is to group time slots with similar means and hence allocate only one group for the midnight period, as there is less traffic, and therefore travel speeds should be similar. This speed profile will be updated after short period of time by an automated system. Data storage requires the mean value for each hour and the entry number. [9] have carried out similar speed analysis for 24 hours in Stuttgart, Germany, with 230 million speed data (3 years).
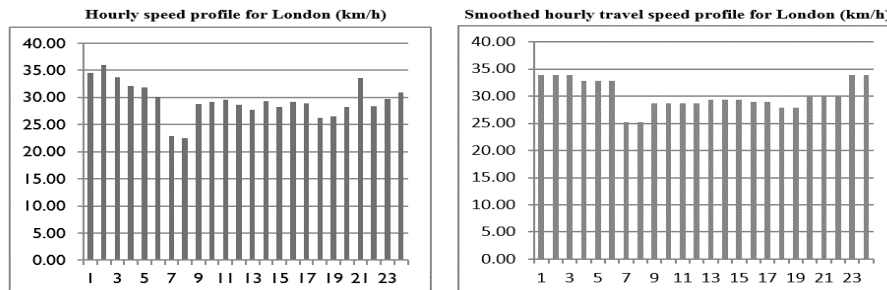

Fig. 1: Speed (km/h) profile for London raw vs. smoothed version

2.3 Vehicle Details Mapped to Emission Formulas

There are 265 CO_2 emission formulas for different vehicle specification in NAEI project and 33 different emission formulas in MEET project. Both project have formulas for other types of pollutants such as CO, HC and NOx. Given an engineer vehicles registration number, the information of engineer vehicles details can be found from TRL website or from company database. Then this vehicle is mapped to one specified formula within the NAEI/MEET formulas set. In NAEI we need vehicle basic types (such as small commercial vehicle or heavy-duty vehicle), fuel type, gross weight, European emission standard and engine capacity, however in MEET, we need basic type, fuel type and gross weight.

3 Solution Engines

3.1 Initial Solution

An initial feasible solution is constructed using an insertion heuristic. This algorithm generates a good quality initial feasible solution, which is important to the success of Tabu search algorithm. RVNS algorithm is more robust to the quality of the initial solution but also requires it to be feasible. Given a list of engineers who are working for a certain period and a list of tasks to be scheduled, a traditional insertion heuristics will randomly choose a task from the task list and insert it into the best and feasible partial route. In vehicle routing problem with time windows and skill matching constraints, this method will lead to problems as it may first insert some tasks that are less difficult to accommodate and leave the tasks which are more difficult to accommodate to be inserted later. In this paper, an extended insertion heuristic is introduced. First, all tasks are ranked in descending order of their difficulties to insert. The difficulty is measured by task time window width and task requested skills rareness. We want to schedule the most difficult task first. $Difficulty_i = a \cdot TimeWindowWidth(i) + \beta \cdot Taskrareness(i), Taskrareness(i) = No.\ of\ Eng\ has\ the\ skill/Times\ the\ skill\ are\ requested.$ The smaller the $Difficulty_i$, the harder it is to accommodate the task. In this study, α and β are tuned to be 0.01 and 300 to give good performance. The cost in step 6 is the additional amount of CO_2 emissions when inserting this task to the current partial route.

3.2 Selection of Metaheuristics

In a survey paper of vehicle routing problem, metaheuristics (71.25% of papers reviewed) are used more often than exact methods and classical heuristics, furthermore simulation and real-time solution methods are rarely applied. Vehicle routing problem is known to be NP-hard, and exact methods are computational costly for industrial sized problems [10]. The state of the art metaheuristics used

---
Algorithm 1 InsertionAlgorithm

1: **Rank** tasks in descending order of $Difficulty_i$
2: **Get** ordered taskList
3: $i \leftarrow 1$
4: **Repeat**
5:     insert $(task_i)$ to lowest cost and feasible position in current partial route
6: $i \leftarrow i + 1$
7: **Until** $i =$ taskList length
---

in green vehicle routing problems are reviewed by [11], and they include: tabu search, simulated annealing, variable neighborhood search, genetic algorithm and ant colony optimization. We will include tabu search and two variants of VNS. We exclude the use of genetic algorithm as its characteristics is not convenient for time-window constrained problems. During the mutate or crossover stage, problem feasibilities are hard to maintain. We tried simulated annealing method but this algorithm depends largely on the decision of parameters and choice of the cooling strategy. We have 5 different test cases, each may have a different set of parameters, SA is not very robust and parameters tuning require extra computational time. Experimental results of SA with different configurations are shown in Fig. 2 showing the solution performance against number of iterations. Among these four configurations, (4) gives the best solution in terms of objective value but computational time is 300s comparing to 0.06s in (1).
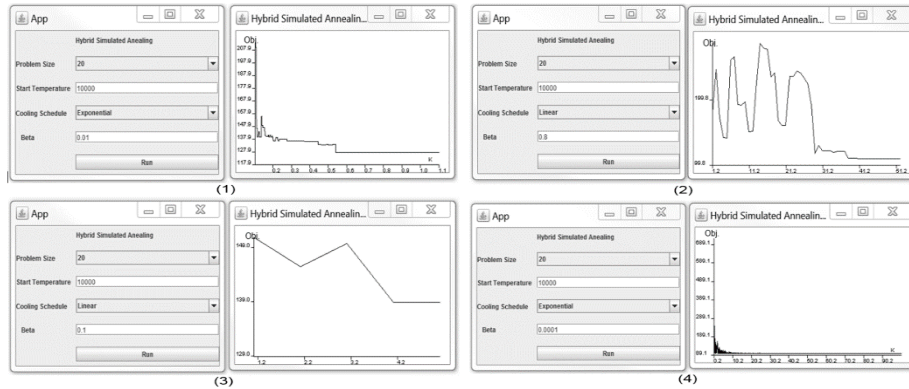

Fig. 2: Simulated annealing results on same data set with different configurations

Neighbourhood Operators In this paper, 6 different operators are used to enable a wider exploration of solution space as shown in Fig. 3. Those operators are a mix of node, link and chain changes.

1. 2-Opt: this is applied to each single route by breaking two links and reconnect. Before changes the route is A-B-C-D-E-F-G-H and after changes the route is A-B-F-E-D-C-G-H.

2. Swap: this is applied to each single route by swapping the position of two nodes. Before changes the route is A-B-C-D-E and after changes the route is A-D-C-B-E.
3. Exchange: this is applied to a pair of routes by swapping the position of two nodes from each route. Before changes the route is A-B-C and D-E-F and after changes the route is A-B-E and D-C-F.
4. Relocation: this is applied to a pair of routes by taking out one note from one route and insert it into another. Before changes the route is A-B-C-D and E-F-G-H and after changes the route is A-B-D and E-F-C-G-H.
5. 2-Opt*: this is applied to a pair of routes by breaking one link from each route and reconnect two routes. Before changes the route is A-B-C-D and E-F-G-H and after changes the route is A-B-D and E-F-C-G-H
6. Or-Opt: this is applied to each single route by moving a chain of k nodes to a new position, e.g. chain D-E will be relocated. Before changes the route is A-B-C-D-E-F and after changes the route is A-E-D-B-C-F.
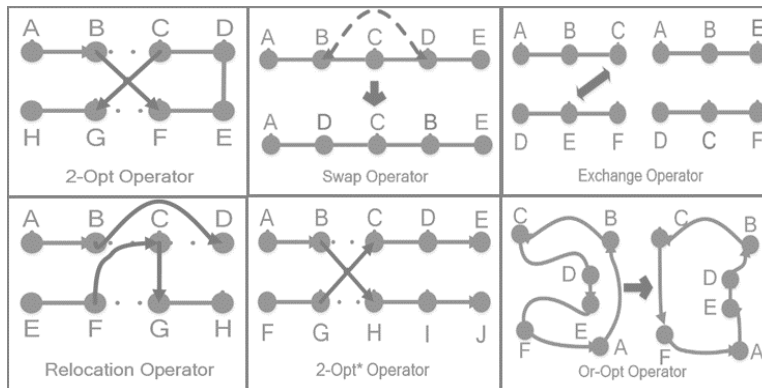

Fig. 3: Neighborhoods' operators

Variable Neighbourhood Search Variable neighbourhood search is a meta-heuristic algorithm based on systematic changes of neighbourhoods in the search process. The ability of escaping from local optimum is enhanced by starting the search in each neighbourhood from a random neighbour of the incumbent solution. Since it was introduced, this method has developed rapidly and found success in solving combinatorial optimisation problem. The idea of VNS is based on the observation that a local minimum found in one neighbourhood structure is not necessary a local minimum for other neighbourhood structures; A global minimum is a local minimum in the combination of all possible neighbourhood structures; Local minima found by different neighbourhoods are relatively close to each other for many problems. There are several variants of VNS, and in this paper, variable neighbourhood descent and reduced variable neighbourhood search will be studied. [12].

Variable Neighbourhood Descent (VND) VND does not involve randomness in the search. Using the current neighbourhood operator k, local search

is performed with the best move strategy until no improvement can be found with the current neighbourhood operator. The search then continues using the next neighbourhood operator $k + 1$, or go back to neighbourhood operator 1 if $k = k_{max}$. This searching process will terminate until no improvement is found for all k.

Algorithm 2 VND(x)

1: **repeat**
2:    $k \leftarrow 1$
3:    **repeat**
4:       $x^{'} \leftarrow BestNeighbour(x, k)$
5:       **if then** $x^{'}$ is better than x, $x \leftarrow x^{'}$
6:       **else** $k = k + 1$
7:       **end if**
8:    **until** $k = k_{max}$
9: **until** no improvement could be found

Reduced Variable Neighbourhood Search (RVNS) The problem with descent algorithms is that it can get trapped to the local optima and cannot escape from it even though we use variable neighbourhood operators. RVNS overcomes this by generating all trial solutions through shake, i.e., randomly generating a solution $x^{'}$ from the kth neighbourhood of x. The stopping criterion in this paper is chosen to be a maximum run time.

Algorithm 3 RVNS(x)

1: $Cputime \leftarrow 0$
2: **repeat**
3:    $k \leftarrow 1$
4:    **repeat**
5:       $x^{'} \leftarrow Shake(x, k)$
6:       **if then** $x^{'}$ is better than x, $x \leftarrow x^{'}$
7:       **else** $k = k + 1$
8:       **end if**
9:    **until** $k = k_{max}$
10: **until** $Cputime > time_{max}$

Tabu Search Tabu search is a metaheuristic search method employing local search methods and have been widely used in vehicle routing problems with time window constraints. Comparing to descent methods, the objective is allowed to deteriorate in order to avoid local minima. To prevent cycling, solutions that are recently visited are prohibited and stored in a tabu list. The stopping criteria is either it reaches the maximum total iterations numbered *iter_max* or the best solution has not been updated for *iter_cons_max* iterations. Step 6 to 8 perform local search methods within a specified neighbourhood. This neighbourhood is generated at each iteration by a neighbourhood operator which is randomly

selected from the previous mentioned six operators. Then the solution that has been previously visited (memorised in tabu list) or violated constraints will be eliminated. The reason for only keeping feasible solutions is once the infeasible solution is accepted, it is hard to regain feasibility using a tabu search algorithm. Then in step 6, local search takes a best improvement strategy, which could be switched to first improvement strategy. The structure of memory is a short-term memory with a fix size of 6, which means the 6 most recent solution will be in the tabu list. $f(x)$ is the fitness score of the solution x, and in this paper it is the total CO_2 emissions amount of the solution.

Algorithm 4 TabuSearch

1: $iter \leftarrow 1, iter\_cons \leftarrow 1; x_{best} \leftarrow x_{initial}, x_{current} \leftarrow x_{initial}$
2: **repeat**
3:     $k = random(1\cdots6)$
4:     $k \leftarrow 1, Cputime \leftarrow 0$
5:     **Generate** $neighbour_k(x_{current})$/$*$ Neighbourhood solutions by kth operator $*$/
6:     Remove tabu and infeasible solution in $neighbour_k(x_{current})$
7:     $x_{current} \leftarrow$ **Best**$(neighbour_k(x_{current}))$
8:     Update TabuList: add $x_{current}$, remove the least recent x
9:     **if then** $f(x_{current}) < f(x_{best}), iter\_cons \leftarrow 1$
10:    **else** $iter\_cons \leftarrow iter\_cons + 1; iter \leftarrow iter + 1$
11:    **end if**
12: **until** $iter = iter\_max$ or $iter\_cons = iter\_cons\_max$

3.3   Real-Life Test Cases and Comparison Between Solution Engines

Five real life data samples were taken from a field service provider with similar problem size (around 100 tasks per sample). The performance measures of the solution are the number of vehicles or engineers scheduled to fulfil all the tasks, total amount of CO_2 emission in grams generated by the final routes planning and number of trips that are bundled together (tasks at the same location). The performance of initial feasible solution by the insertion algorithm are shown in Table 1. Our different test cases have different features. For example, in Fig. 4, comparing test case 1 and test case 3, case 3 has a higher number of wide spread tasks on the map, that is why its initial solution has a higher CO_2 emissions.

Table 1: Initial Solution

Text case	No. of Task	No.Engineer	CO_2 kg	Bundle Trip
Case 1	103	19	187.613	46
Case 2	110	18	150.044	51
Case 3	108	20	236.522	46
Case 4	100	17	201.032	42
Case 5	106	24	233.129	41

Fig. 4: Test case 1 (left) vs. test case 3 (right)

3.4 Comparison of Metaheuristics on Real-Life Test Cases

VND vs. RVNS For each test case, we first let VND solve the problem and get the solutions and Cpu times, then we let RVNS run for 100 seconds which are similar to the Cpu times of VND and compare the results. As RVNS algorithm has a stochastic nature, this algorithm is run for 30 times to conduct statistical analysis. A one-sample t-test is carried out to see if the mean solution of RVNS is significantly different from the VND, and the significance level is set to be 0.1. The results are shown in Table 2 with the best solution highlighted in bold. Other than case 2, RVNS yields significantly better solution than VND and it produces the best solution for all 5 cases. Comparing the emissions results, on average RVNS gives 3.7% less emissions with the same computational time. Results suggest that RVNS outperforms VND.

Table 2: Comparison between VND and RVNS

	Emissions kg	Case 1	Case 2	Case 3	Case 4	Case 5
RVNS	Best	148.95	143.32	158.90	130.61	161.95
	Average	160.56	146.74	181.00	165.97	183.73
	S.D	6.28	2.11	14.17	15.14	10.15
VND		171.87	144.34	187.34	170.81	198.52
Cpu time (seconds)		100	100	150	300	600
p-value		0.000	0.000	0.020	0.091	0.000
Different?		Y	Y	Y	Y	Y

RVNS vs. Tabu Tabu search algorithm randomly changes neighbourhood, so this algorithm is also run for 30 times for each test cases. The stopping criteria is set to be 15000 maximum evaluations and 300 non-improving evaluations, then the computational time is fluctuating, unlike RVNS which is set to be terminated at a certain maximum run time. Table 3 shows the comparisons of the two engines by the means of average CO_2 emissions and average computational time. The

best solution value of CO_2 emissions is highlighted in bold. Among the five test cases, these two engines generated similar solutions for case 3 and case 5 by means of CO_2 emissions, but the computational time of Tabu algorithm is on average 5.37 times longer than RVNS. For the other test cases, Tabu algorithm gave significantly better results than RVNS, the CO_2 emissions is on average 5.35% less, but at an expense of 11.72 times of computational time. Although the best solution was found by the Tabu search algorithm in most cases, results suggest RVNS is a good trade-off between solution quality and computational time for industrial application.

Table 3: Comparison between Tabu and RVNS

	Emissions	Case 1	Case 2	Case 3	Case 4	Case 5
RVNS	Best	148.95	143.32	158.90	**130.61**	**161.95**
	Average	160.56	146.74	181.00	165.97	183.73
	S.D	6.28	2.11	14.17	15.14	10.15
	Cpu Time	100	100	150	300	600
Tabu	Best	**142.52**	**131.57**	**156.31**	140.79	163.24
	Average	156.44	137.83	178.82	153.66	188.73
	S.D	9.00	2.74	7.01	6.15	10.19
	Average Time(s)	1642.14	1616.32	1492.69	1668.29	1665.90
	S.D.	521.45	503.35	288.39	740.51	457.93
p-value		0.044	0.000	0.454	0.000	0.105
Different?		Y	Y	N	Y	N

4 Heat Map View

The impact of the proposed green logistic engineer scheduling and routing system on the business, can be illustrated graphically. A sample of a scheduling profile of 179 engineers between 01st May and 15th May 2017 is analyzed. This sample resolves 4,173 unique tasks in Colchester and Ipswich domain. It can be noticed from Fig. 5 (left), higher emission from 7 to 9 hrs. Compared to the picture reported in Fig. 5 (right), which reports the recorded emission from 11 to 13 hrs. Where, roads in shadow represent the transited route with a certain CO_2 emission. Darker colors indicate higher CO_2 emission. The threshold of coloring is adjusted for comparison purposes according to the desired snapshot. Therefore, it is categorized in according to certain ranges, such as: $t_{low} = x[min, \mu - \sigma^2]$, $t_{medium} = x[\mu - \sigma^2, \mu + \sigma^2]$ and $t_{high} = x[\mu + \sigma^2, max]$, where x corresponds to the CO_2 value to be represented, min is the minimum CO_2 value recorded, μ is the mean of the CO_2 emissions after computation, σ^2 is the standard deviation of this CO_2 computation in the specific time window being evaluated and max is the higher recorded value for the CO_2 in the sample in that slot. Therefore, the resulting thresholds are no fixed and correspond to

the particular desired granularity. Observing the heat map, we can see that the average level of emissions of 7-9 am is higher comparing to 11-13 hrs and this is due to the traffic congestions which is as expected.
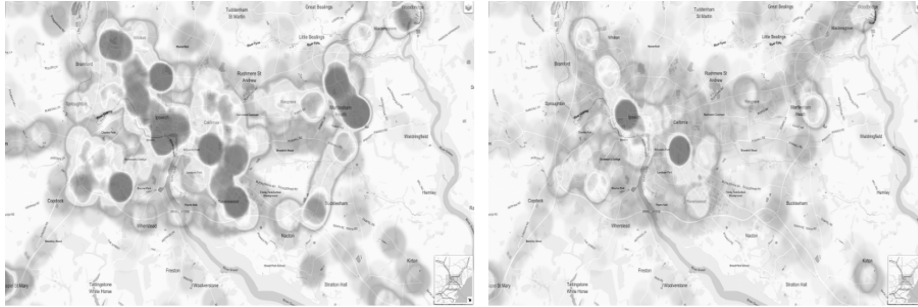

Fig. 5: CO_2 Heat map visualization. (Left) from 7 hrs. to 9 hrs. (Inclusive). (Right) from 11 hrs. to 13 hrs. (Inclusive).

5   Conclusion

A green logistic constraint satisfaction and optimization based engineer scheduling and routing decision support system is designed in this paper, and tested on test cases from a real-life field service provider. Comparing to other green logistics models, this system is more accurate than traditional solutions, as it generates and uses a time-varying travel speed profile instead of a constant speed profile. This speed is then incorporated into the constraint propagation model for the problem together with availability, time window and competences constraints propagation on each engineer and the task potential couple. This proposed system adapts instantaneous emission models which are widely used in green vehicle logistic studies to the real-world case of heterogeneous vehicle fleet. The factors considered by these models are vehicle types, vehicle load and traffic conditions. As vehicle types play an important role in the amount of emissions and as no commercial software maps the two, engineers vehicles number plates are mapped to a specified emission formula. These additional factors aim to make the emissions computation more accurate. The vehicle routing problem is solved using three metaheuristics: RVNS, VND and Tabu. Experiments are carried out to compare three different metaheuristics on 5 real-life typical cases. On average RVNS gives 3.7% less emissions with the same computational time compared to VND, and therefore VND will likely not be included in the final solution engines. For most test cases, Tabu algorithm gave CO_2 related results significantly better than RVNS: the CO_2 emissions is on average 5.35% less, but at an expense of 11.72 times of computational time. Nevertheless, the significantly lower running time cost of RVNS places this algorithm at the top position of industrial operational acceptable algorithm underlying a real-time decision making support solution. The industrial solution relies on fast computations returning a set of

time-contextual recommendations about CO_2 impact and feasible time slots regarding a task-engineer assignment decision. Finally we have included both Tabu algorithm and RVNS algorithm in the solution engines, to cope with the variation of the users needs. Results suggest that RVNS is a good trade-off between solution quality and computational time for industry application, but the users could select Tabu algorithm for better solution if computational time is not a constraint. Finally, to the best of our knowledge this paper is the first to develop a Heat Map view of emissions of schedule plans at different hours of a day: a tool to help managers visualize the emission levels of scheduling and routing plans and graphically see the impact of the proposed green logistic scheduling decision support system.

References

1. Úbeda, S., Faulin, J., Serrano, A., Arcelus, F.: Solving the green capacitated vehicle routing problem using a tabu search algorithm. Lecture Notes in Management Science **6** (2014) 141–149
2. Eglese, R., Black, D.: Optimizing the routing of vehicles. 2010) Green Logistics: Improving the environmental sustainability of logistics, KoganPage (2010) 215–228
3. Qian, J., Eglese, R.: Fuel emissions optimization in vehicle routing problems with time-varying speeds. European Journal of Operational Research **248**(3) (2016) 840–848
4. Turkensteen, M.: The accuracy of carbon emission and fuel consumption computations in green vehicle routing. European Journal of Operational Research **262**(2) (2017) 647–659
5. Demir, E., Bektaş, T., Laporte, G.: A comparative analysis of several vehicle emission models for road freight transportation. Transportation Research Part D: Transport and Environment **16**(5) (2011) 347–357
6. Palmer, A.: The development of an integrated routing and carbon dioxide emissions model for goods vehicles. (2007)
7. Hickman, J., Hassel, D., Joumard, R., Samaras, Z., Sorenson, S.: Meet-methodology for calculating transport emissions and energy consumption. european commission, dg vii. Technical report, ISBN 92-828-6785-4, Luxembourg, 362 p. www. inrets. fr/infos/cost319 (1999)
8. Boulter, P., Barlow, T., McCrae, I.: Emission factors 2009: Report 3-exhaust emission factors for road vehicles in the united kingdom. TRL Published Project Report (2009)
9. Ehmke, J., Campbell, A., Thomas, B.: Vehicle routing to minimize time-dependent emissions in urban areas. European Journal of Operational Research **251**(2) (2016) 478–494
10. Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering **99** (2016) 300–313
11. Lin, C., Choy, K., Ho, G., Chung, S., Lam, H.: Survey of green vehicle routing problem: past and future trends. Expert Systems with Applications **41**(4) (2014) 1118–1138
12. Hansen, P., Mladenović, N., Pérez, J.: Variable neighbourhood search: methods and applications. Annals of Operations Research **175**(1) (2010) 367–407