Simulated Annealing Approach for the Vehicle Routing Problem with Synchronized Visits

Seddik Hadjadj and Hamamache Kheddouci
Laboratoire d'Informatique en Image et Systèmes d'Information, Villeurbanne, France

Keywords: Vehicle Routing Problem, Simulated Annealing, Temporal Precedence Constraint, Synchronized Visits.

Abstract: In this paper, we consider a vehicle routing problem in which customers have to be visited by two different vehicles in a certain order. The first vehicle has to visit *n* customers to pickup or deliver empty containers, while the second has to deliver ready-mixed concrete by pouring it into the previously delivered containers. This problem is a vehicle routing problem (VRP) variant involving synchronized visits and temporal precedence constraints.

We propose an ILP formulation and a simulated annealing algorithm to minimize the total travel times of both vehicles, and we provide experiments that show very satisfying results.

1 INTRODUCTION

This work is carried out in collaboration with a company which is specialized in the sale of ready-mixed concrete.

Today, a ready-mixed concrete order requires necessarily the mobilization of one or more *mixer trucks*, even when dealing with very small quantities of concrete. However, such type of truck is cumbersome, expensive, and disproportionate in some cases, especially when delivering small quantities of concrete.

Therefore, the company proposes a new delivery method to deal more effectively with such orders. The idea is to share a single mixer truck by several customers with small quantities ($\leq$ 2000 liters), which implies organizing optimized mixer truck tours.

On the other hand, to ensure the profitability of these truck tours, waiting times at each customer's location have to be reduced. Today, a mixer truck has to wait until the concrete is poured on site to leave a customer's location, and this causes a huge waste of time. A solution to this problem is to pour the concrete from the truck into special containers instead of pouring it directly on site which is more difficult and takes more time. The truck can then leave faster, and the customer can use the concrete in the containers all day long. Waiting times are then drastically reduced.

However, since the containers which are supposed to contain the concrete are special, they also must be delivered to the customer. This implies dealing with another vehicle tour to deliver the containers and pick them up after they have been used.

In brief, this new method is a three-step process:

1. A vehicle delivers a number of empty containers to the customer;
2. Thereafter, a mixer truck delivers a certain quantity of ready-mixed concrete by pouring it into the containers delivered;
3. The next day, the vehicle returns to the customer to pick up the containers after they have been used.

To ensure the profitability of this method, the company needs a decision support system that can generate two efficient and synchronized vehicle tours: a pickup & delivery tour for the containers, and a mixer truck tour to deliver the concrete, knowing that each customer has to receive the containers before the concrete.

This problem is a vehicle routing problem variant involving combined vehicle tours and temporal precedence constraints, it is then obviously NP-Hard (Lenstra and Kan, 1981).

Despite the fact that synchronized vehicle tours have several real world applications, there are, to the best of our knowledge, only few works dealing with such problems. Among these works, (Bredström and Rönnqvist, 2008) presented the "combined vehicle routing and scheduling problem with time windows and additional temporal constraint". Some real world problems have been presented, and a mathematical programming model has been proposed. A heuristic solution has also been studied to deal with real world instances considering three different objective functions: minimizing the total travel times, maximizing the sum of preference (a customer can prefer being visited by a specific vehicle), and optimizing the workload balance.

(Dohn et al., 2011) studied the vehicle routing problem with time windows and temporal Dependencies (VRPTWTD). Some specific constraints have been considered like the minimum / maximum gap between the starting time and the ending time of visits and a branch and cut and price algorithm was proposed to minimize the total travel cost.

Overall, most of papers dealing with vehicle routing problems with temporal precedence constraints consider exact methods to solve those problems. Only few works attempted to solve such problems with heuristics.

(Afifi et al., 2016) have proposed a heuristic to deal with the VRP with time windows and synchronized visits (VRPTWSyn). They studied a simulated annealing based algorithm to deal more effectively with real-world instances. They obtained satisfying results in very short computational times.

More recently, (Hojabri et al., 2018) proposed an Adaptive Large Neighbourhood Search coupled with Constraint Programming to tackle a problem in which the arrival times of two vehicles at two different customer locations have to be synchronized. The algorithm shows satisfying results on instances derived from standard benchmarks.

(Sarasola and Doerner, 2019) studied a generalization of synchronization constraints in which customers have to be delivered by one or more logistics service providers. Their approach is based on self-imposed time windows. The algorithm obtained satisfying results considering the minimization of idle times (ie, the non service time between the first and the last delivery at each customer location).

(Liu et al., 2019) tackled the vehicle routing problem with time windows and synchronization constraints. Some customers may require two simultaneous deliveries from two different vehicles, and time windows are fixed for each customer. They proposed a Mixed-Integer Programming Model, and a Large Neighbourhood Search to deal with large instances. The approach was tested on standard benchmark instances and three objective functions were studied including the total travel times of all vehicles and the sum of assigned negative preferences. The tests conducted show better results than (Bredström and Rönnqvist, 2008) and (Afifi et al., 2016).

This paper aims to provide an efficient heuristic that can handle real-world instances to build two synchronized vehicle tours minimizing the total travel times and satisfying temporal precedence constraints.

The paper is structured as follows. Section 2 provides an ILP formulation for the problem tackled in this work. Section 3 presents our heuristic approach to solve the problem. Section 4 is devoted to some experimental results, and Section 5 concludes the paper.

2 PROBLEM FORMULATION

Let V_1 and V_2 be two vehicles such that:

- V_1 is in charge of delivering (or picking up) empty containers;
- V_2 is a mixer truck carrying ready-mixed concrete.

Note that for both V_1 and V_2, we do not take into account vehicle capacity constraints, we assume that V_1 has a sufficient capacity to carry all the containers involved during a tour, and V_2 leaves the depot with a sufficient quantity of concrete to satisfy all customers' demands.
We consider:

- D_1 the depot of V_1 (D_1 is the arrival and departure point of V_1's tour);
- D_2 the depot of V_2 (D_2 is the arrival and departure point of V_2's tour);
- $N = \{1, ..., n\}$ a set of n customers who require a visit of V_1 and/or V_2;
- $N = N_p \cup N_d$, where:
  - N_d is the set of customers who require delivering containers + concrete (who require a visit of both V_1 and V_2);
  - N_p is the set of customers who require picking up containers (who require a visit of V_1 only);
  - $N_p \cap N_d = \emptyset$.

The problem can be defined on a complete graph $G = (V, E)$ as follows (see Figure 1):

- $V = \{D_1\} \cup \{D_2\} \cup N$ is a set of $n + 2$ nodes;
- $E = \{(i, j) : i, j \in V, i \neq j\}$ is a set of edges representing connections between nodes;
- $C = \{c_{i,j} : (i, j) \in E\}$ represents the travel time between i and j ($c_{i,j} = c_{j,i}, \forall (i, j) \in E$);
- $D = \{d_i : i \in N\}$ is a set of customers' demands ($|d_i|$ is the number of containers to deliver to / pick up from customer i : $d_i < 0 \quad \forall i \in N_d$ and $d_i > 0 \quad \forall i \in N_p$);

Assuming that:

- $x_{i,j}$ is a boolean variable such that:

 - $x_{i,j} = 1$ if j is visited immediately after i by V_1,
 - 0 otherwise.

- $y_{i,j}$ is a boolean variable such that:
  - $y_{i,j} = 1$ if j is visited immediately after i by V_2,
  - $y_{i,j} = 0$ otherwise.

  (Note that $y_{i,j} = 0 \quad \forall i \in N_p, \forall j \in N_p$).
- $t_{1,i}$ represents the departure time of V_1 from customer i location ($t_{1,0}$ represents the departure time of V_1 from the depot D_1);
- $t_{2,i}$ represents the departure time of V_2 from customer i location ($t_{2,0}$ represents de departure time of V_2 from the depot D_2).

The objective is to find two optimized vehicle tours T_{V_1} and T_{V_2} minimizing the total travel times, such that T_{V_1} is a pickup & delivery tour through n customers, and T_{V_2} is a concrete delivery tour through the n_d customers who have received containers.
Thus, we formulate the following integer linear program:

$$min \quad \sum_{i=0}^{n}\sum_{j=0}^{n} x_{i,j}c_{i,j} + \sum_{i=0}^{n}\sum_{j=0}^{n} y_{i,j}c_{i,j} \qquad (1)$$

Subject to:

$$\sum_{j\in N} x_{i,j} = 1 \quad \forall i \in \{D_1\} \cup N \qquad (2)$$

$$\sum_{i\in N} x_{i,j} = 1 \quad \forall j \in \{D_1\} \cup N \qquad (3)$$

$$\sum_{j\in N_d} y_{i,j} = 1 \quad \forall i \in \{D_2\} \cup N_d \qquad (4)$$

$$\sum_{i\in N_d} y_{i,j} = 1 \quad \forall j \in \{D_2\} \cup N_d \qquad (5)$$

$$x_{i,D_2} = 0 \quad \forall i \in \{D_1\} \cup N \qquad (6)$$

$$x_{D_2,i} = 0 \quad \forall i \in \{D_1\} \cup N \qquad (7)$$

$$y_{i,j} = 0 \quad \forall i,j \in \{D_1\} \cup N_p \qquad (8)$$

$$u_i - u_j + |N|x_{i,j} \leq |N| - 1 \quad \forall i,j \in N, i \neq j \qquad (9)$$

$$v_i - v_j + |N_d|y_{i,j} \leq |N_d| - 1 \quad \forall i,j \in N_d, i \neq j \qquad (10)$$

$$t_{1,j} - t_{1,i} + MAXx_{i,j} \leq MAX + c_{i,j} \quad \forall i,j \in \{D_1\} \cup N \qquad (11)$$

$$t_{1,j} - t_{1,i} + MINx_{i,j} \geq MIN + c_{i,j} \quad \forall i,j \in \{D_1\} \cup N \qquad (12)$$

$$t_{2,j} - t_{2,i} + MAXy_{i,j} \leq MAX + c_{i,j} \quad \forall i,j \in \{D_2\} \cup N_d \qquad (13)$$

$$t_{2,j} - t_{2,i} + MINy_{i,j} \geq MIN + c_{i,j} \quad \forall i,j \in \{D_2\} \cup N_d \qquad (14)$$

$$t_{1,0} = 0 \qquad (15)$$

$$t_{2,i} \geq t_{1,i} \quad \forall i \in N_d \qquad (16)$$

$$0 \leq u_i \leq |N| \quad \forall i \in N \qquad (17)$$

$$0 \leq v_i \leq |N_d| \quad \forall i \in N_d \qquad (18)$$

$$t_{1,i} \in \mathbb{N} \quad \forall i \in \{D_1\} \cup N \qquad (19)$$

$$t_{2,i} \in \mathbb{N} \quad \forall i \in \{D_2\} \cup N_d \qquad (20)$$

$$x_{i,j} \in \{0,1\} \quad \forall i,j \in \{D_1\} \cup \{D_2\} \cup N \qquad (21)$$

Where:

- Constraints (2) and (3) ensure that each customer is visited exactly once by vehicle V_1;
- Constraints (4) and (5) require that each "delivery customer" is visited exactly once by vehicle V_2;
- Constraints (6) and (7) relate to the fact that V_1 cannot visit the depot of V_2, while (8) ensures that V_2 cannot visit neither the depot of V_1 nor the "pickup customers";
- Constraint (9) references a dummy variable u_i to ensure that all customers are connected by a single tour of V_1, and not by two or more disjointed tours;
- Constraint (10) references a dummy variable v_i to ensure that all delivery customers are connected by a single tour of V_2, and not by two or more disjoint tours;
- Constraints (11) and (12) concern the computing of departure times of V_1 from each customer's location. Thus, we distinguish two cases:
  - If customer j is visited by V_1 immediately after customer i (if $x_{i,j} = 1$), then, from (11) and (12), we have:

    $$t_{1,j} - t_{1,i} + MAX \leq MAX + c_{i,j}$$

    $$t_{1,j} - t_{1,i} + MIN \geq MIN + c_{i,j}$$

    which implies:

    $$t_{1,j} - t_{1,i} \leq c_{i,j}$$

    $$t_{1,j} - t_{1,i} \geq c_{i,j}$$

    and then:

    $$t_{1,j} = t_{1,i} + c_{i,j}$$

    That means that the departure time from customer j is equal to the departure time from customer i + the travel time between i and j. Note that waiting times at each customer's location are ignored.
  - If customer j is not visited by V_1 immediately after customer i (if $x_{i,j} = 0$), then, from (11) and (12), we have:

    $$t_{1,j} - t_{1,i} \leq MAX + c_{i,j}$$

    $$t_{1,j} - t_{1,i} \geq MIN + c_{i,j}$$

    Knowing that MAX and MIN are two sufficiently large(resp. small) numbers, constraints (11) and (12) are then always satisfied when $x_{i,j} = 0$;
- In the same way as for constraints (11) and (12), (13) and (14) relate to the computing of departure times of V_2 from each "delivery" customer;
- (15) requires that V_1 leaves its depot at $t = 0$;
- Constraint (16) concern the temporal precedence between V_1 and V_2. The vehicle V_2 cannot arrives at a customer's location before V_1.

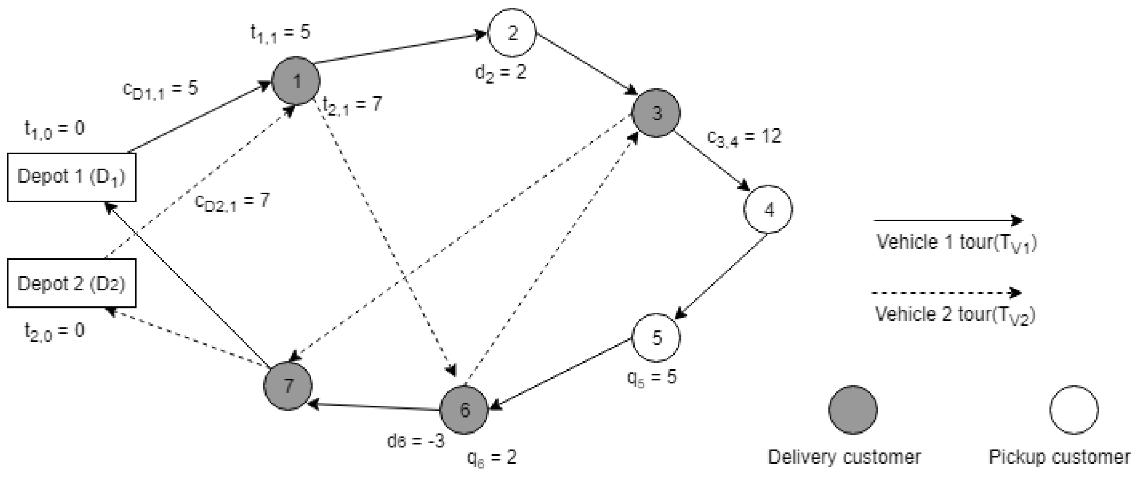

Figure 1: Synchronized vehicle tours.

3 SIMULATED ANNEALING ALGORITHM

In order to tackle the problem described above, we propose a simulated annealing-based approach (Kirkpatrick et al., 1983) (see Algorithm 1).

Simulated annealing is a well-known metaheuristic which has been widely studied and gave satisfying results for several types of optimization problems, including VRP's (Chiang and Russell, 1996) (Czech and Czarnas, 2002) (Van Breedam, 1995).

Algorithm 1: Simulated annealing framework.

```
1: $iterationsWithoutImprovement \leftarrow 0$ ;
2: $T \leftarrow T_0$ ;
3: $S \leftarrow initialSolution()$ ;
4: $bestSolution \leftarrow S$ ;
5: $N \leftarrow \emptyset$ ;
6: while ($iterationsWithoutImprovement < maxIterations$) do
7:     $iterationsWithoutImprovement{+}{+}$ ;
8:     $N \leftarrow Neighborhood(S)$ ;
9:     $S\prime \leftarrow BestNeighbour(N)$ ;
10:    $\Delta \leftarrow f(S\prime) - f(S)$ ;
11:    $p \leftarrow Random(0,1)$ ;
12:    if ($p \leq e^{-\Delta/T}$) then
13:        $S \leftarrow S'$ ;
14:        if ($f(S) \leq f(bestSolution)$) then
15:            $bestSolution \leftarrow S$ ;
16:            $iterationsWithoutImprovement \leftarrow 0$ ;
17:        end if
18:    end if
19:    $T \leftarrow \alpha T$ ;
20: end while
```

The main idea of simulated annealing is to try to move, at each iteration, from a current solution S, to a neighbour one $S\prime$ according to a certain probability $e^{-\Delta/T}$. This probability is updated at each iteration and depends on two parameters:

- $\Delta = f(S\prime) - f(S)$. If $S\prime$ improves S (if $\Delta \leq 0$), then, the probability to move to $S\prime$ is equal to 1. Otherwise, the smaller Δ, the higher the probability to move to $S\prime$;
- The temperature T. This parameter starts with an initial value T_0 and decreases progressively at each iteration. The greater T, the higher the probability to move to $S\prime$. In other words, the algorithm is more likely to accept a candidate solution $S\prime$ during the first iterations.

As it appears in Algo.1, the simulated annealing needs an initial solution, and a heuristic to generate the neighbourhood of the current solution at each iteration.

3.1 Initial Solution

As an initial solution, we consider the solution obtained by the "nearest neighbour algorithm". Thus, for the vehicle V_1, we start from the depot D_1 at $t = 0$ and we move at each iteration to the nearest customer from the current location of V_1 (in terms of travel time) until all customers have been visited. The same procedure is applied for V_2: we start from D_2 and move at each iteration to the nearest "delivery customer", until all delivery customers have been visited by V_2. Note that the departure time of V_2 from the depot D_2 is set to $t_{1,i} - c_{D_2,i}$, where i is the first customer to be visited by V_2, $t_{1,i}$ is the departure time of V_1 from customer's i location, and $c_{D_2,i}$ the travel time between D_2 and i. In other words, we choose the moment where V_2 leaves the depot to visit customer i in such a way that both V_1 and V_2 arrive at the same time at customer's i location.

3.2 Neighbourhood Structure

We define a neighbour of a solution S as a solution in which we swap the positions of two random customers in one or both tours. Thus, to generate a neighbour solution:

- We start by choosing 2 random customers i and j among all the customers to be visited (including "pickup" customers and "delivery" customers);
- If at least one of these customers is a "pickup" customer (ie: he requires a visit of V_1 only), then we swap the positions of i and j in V_1's tour;
- If the two chosen customers are "delivery" customers (ie: they require a visit of both vehicles), then we swap positions of i and j either in V_1's tour, or in V_2's tour, or in both tours (we choose randomly one of these three strategies).

So, as it is described in Figure 2, the neighbour solution (on the right side) is obtained by swapping the positions of customer 1 and customer 3. In the initial solution, customer 1 is the first customer to be visited by both V_1 and V_2, while customer 3 is at the third position in V_1's tour, and at the second position in V_2's tour. A neighbour solution is then generated by moving customer 1 from positions $[1,1]$ to $[3,2]$ and customer 3 from positions $[3,2]$ to $[1,1]$.

4 COMPUTATIONAL RESULTS

The heuristic and the ILP model presented above were implemented in Java and CPLEX 12.9. Both were executed on AMD A10-7700K Radeon R7, 3.40 GHz With 8 GB RAM.

To the best of our knowledge, there are no benchmark instances for simultaneous vehicle routing problems with pickup & delivery. Therefore, we tested our algorithm on the Euclidian PDTSP instances generated by (Gendreau et al., 1999), which consider a single depot for each instance. The benchmark contains 180 instances where the number of customers varies between 25 and 200. We adapted the instances to fit our constraints by considering the depot and the first customer of each instance as the depots of the two vehicles considered in our problem.

However, to the best of our knowledge, there are no papers dealing with the same problem as ours, so, we have compared our results to the optimal solutions found by our ILP model.

Concerning the simulated annealing parameters, several values have been tested for the initial temperature T_0 and the parameter α. In the following, we consider $T_0 = 4800$, $\alpha = 0.95$, and maxIterations $= 200$.

Table 1 shows the average results obtained by our simulated annealing for the 6 classes of instances (25, 50, 75, 100, 150, and 200 nodes).
The first column represents the class of instances (the number of nodes), the second is the average total travel times obtained by our simulated annealing on the 30 instances tested for each class. The third column gives the average V_1 travel times, while the fourth column represents the average V_2 travel times (note that total travel times = V_1 travel times + V_2 travel times). The fifth column shows the average tardiness obtained by the algorithm. For each instance, the sum of tardiness is the sum of the gaps between the arrival time of V_1 and the arrival time of V_2 at each customer's location. In other words, given a solution, $tardiness = \sum_{i \in N_d} t_{2,i} - t_{1,i}$. Note that tardiness concerns only "delivery" customers and its value is always positive because we have the constraint $t_{2,i} \geq t_{1,i}$. The last column gives the CPU time in ms.

First, we note that the computational times are very short, even for the largest instances. Furthermore, we see that pickup & delivery tours (vehicle V_1) are more costly than concrete delivery tours (V_2) because they involve more customers. However, knowing that the number of customers visited by V_1 is about the double of the number of customers visited by V_2 in the considered instances, the impact of V_2's tours may appear relatively important (it represents more than 40% of the total travel times for all classes of instances). This can be explained by the fact that V_1's tour are more flexible than V_2's ones, since V_2 is subject to a constraint that enforces it to do not visit a customer before V_1. Thus, good solutions may be rejected because they do not satisfy this constraint.

Table 2 shows a comparison between simulated annealing results and the optimal solutions found by our ILP model. Note that we could not obtain optimal solutions in reasonable time for large instances (≥ 75). Thus, in the following table, we do not compare all the instances as in table 1, only 14 instances have been compared, the first 10 instances of 25 nodes, and the first 4 instances of 50 nodes.

First, we observe that our simulated annealing obtained very satisfying results. The solutions found are globally close to optimal solutions, and the computational times are very much shorter (up to x 27000). Furthermore, for large instances, optimal solutions could not be obtained in reasonable time, contrary to

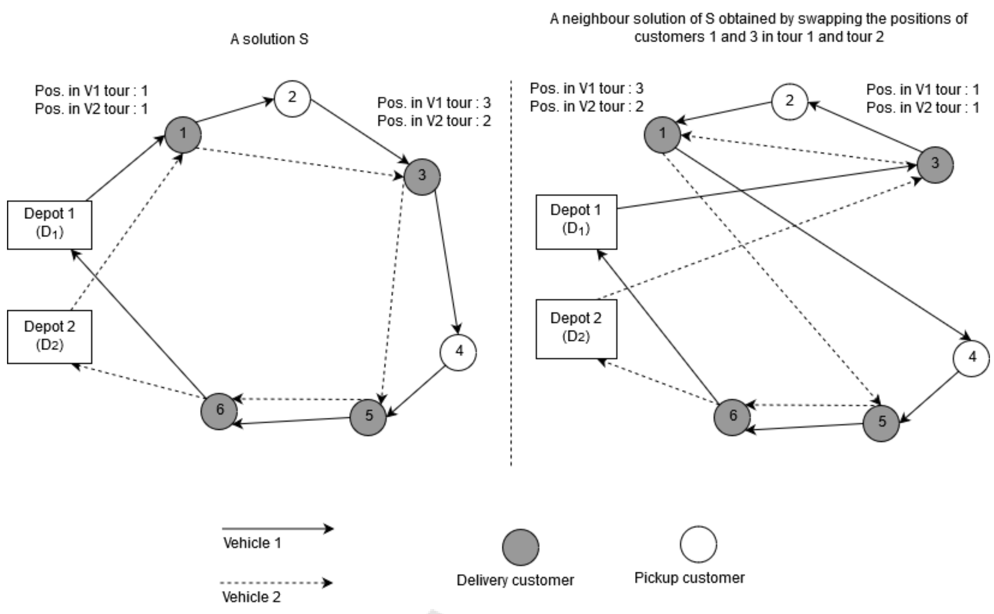

Figure 2: Neighbourhood generation.

Table 1: Simulated annealing results for the instances of (Gendreau et al., 1999).

Number of nodes	Total travel times	Pickup & delivery tour (V1)	Concrete delivery tour (V2)	Tardiness	CPU time (ms)
25	908	530 (58%)	377(42%)	2461	95
50	1266	685 (54%)	580 (46%)	9276	457
75	1467	837(57%)	629 (43%)	21148	1046
100	1692	967 (57%)	724 (43%)	32761	2029
150	2047	1216 (57%)	830 (43%)	59562	6719
200	2451	1462 (57%)	988 (43%)	93546	18198

our heuristic which is effective for large instances (see Table 1).

On the other hand, we note that the solutions found by our heuristic present globally less tardiness than the optimal ones. In other words, waiting times between the visits of V_1 and V_2 at each customer's location are globally shorter with our simulated annealing solutions, which is interesting from the customer's point of view. Finally, we observe that the concrete delivery tours (V_2) obtained by the simulated annealing are much closer to optimal tours than the pickup & delivery tours (V_1). This is due to the fact that V_2 has a smaller set of customers to visit, which makes the search space smaller for the heuristic. Moreover, the fact that V_2's tours are more optimized than V_1's tours can explain why the tardiness is globally shorter with the solutions obtained by the simulated annealing. Indeed, if V_2 has a more optimized route than V_1, it would be more difficult to have a large gap between V_1 and V_2, knowing that V_1 has more customers to visit.

5 CONCLUSIONS

We presented a simulated annealing and an ILP model to tackle the vehicle routing problem with synchronized visits. The problem involves two vehicles: the first has to visit n customers to pickup or deliver empty containers, while the second has to deliver ready-mixed concrete by pouring it into the previously delivered containers. The objective function considered is the minimization of the total travel times.

We tested our algorithm and ILP model on the Euclidian PDTSP instances proposed in (Gendreau et al., 1999). We adapted the instances to fit our constraints and collected the results, which were satisfying both in terms of computational times and in terms of solution quality.

Future works will be devoted to the development of multi-objective approaches including the minimization of the total travel times and the minimization of the tardiness.

Table 2: Simulated annealing solutions VS optimal solutions (ILP).

	ILP solutions					Simulated annealing solutions				
Instance	CPU time (ms)	Total travel time	Pickup & delivery tour (V1)	Concrete delivery tour (V2)	Tardiness	CPU time (ms)	Total travel time	Pickup & delivery tour (V1)	Concrete delivery tour (V2)	Tardiness
25_0	13343	688	388	300	2037	**218**	816	510	306	**966**
25_1	15140	791	460	331	**2152**	**109**	991	641	350	2665
25_2	421	828	464	364	4838	**109**	990	524	466	**4731**
25_3	2375	789	464	325	3249	**78**	978	603	375	**3113**
25_4	12531	754	426	328	2035	**62**	801	467	334	**1280**
25_5	43703	806	453	353	3663	**62**	1018	600	418	**2781**
25_6	9656	771	440	331	**1394**	**78**	829	488	341	3960
25_7	569734	820	448	372	4298	**109**	1037	578	459	**1111**
25_8	19171	730	432	298	2722	**78**	820	468	352	**760**
25_9	104328	755	420	335	**341**	**93**	855	495	360	1685
50_0	526156	1035	581	454	9608	**468**	1268	742	526	**9222**
50_1	224953	1030	599	431	**6300**	**312**	1268	709	559	9964
50_2	13504468	968	553	415	11561	**500**	1154	658	496	**7368**
50_3	3533703	1071	595	476	12570	**562**	1231	651	580	**8599**
50_4	1179000	1007	574	433	10113	**484**	1206	722	484	**10005**

Other types of metaheuristics will be also studied, such as genetic algorithms, and more constraints will be involved including vehicle capacity constraints and time windows.

REFERENCES

Afifi, S., Dang, D.-C., and Moukrim, A. (2016). Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optimization Letters*, 10(3):511–525.

Bredström, D. and Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European journal of operational research*, 191(1):19–31.

Chiang, W.-C. and Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1):3–27.

Czech, Z. J. and Czarnas, P. (2002). Parallel simulated annealing for the vehicle routing problem with time windows. In *Proceedings 10th Euromicro workshop on parallel, distributed and network-based processing*, pages 376–383. IEEE.

Dohn, A., Rasmussen, M. S., and Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, 58(4):273–289.

Gendreau, M., Laporte, G., and Vigo, D. (1999). Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research*, 26(7):699–714.

Hojabri, H., Gendreau, M., Potvin, J.-Y., and Rousseau, L.-M. (2018). Large neighborhood search with constraint programming for a vehicle routing problem with synchronization constraints. *Computers & Operations Research*, 92:87–97.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Lenstra, J. K. and Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227.

Liu, R., Tao, Y., and Xie, X. (2019). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. *Computers & Operations Research*, 101:250–262.

Sarasola, B. and Doerner, K. F. (2019). Adaptive large neighborhood search for the vehicle routing problem with synchronization constraints at the delivery location. *Networks*.

Van Breedam, A. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, 86(3):480–490.