Optimization of Delivery Scheduling Under Traffic Congestion Constraints

Edouard Seguier, Louis Lhotte

February 22, 2025

Abstract

This report presents a comprehensive study on the optimization of delivery scheduling under traffic congestion constraints. The problem is formulated as a Boolean Satisfiability (SAT) problem, where the goal is to assign delivery tasks to a set of delivery personnel (livreurs) while respecting client deadlines, traffic congestion, and operational constraints. The problem is encoded into a SAT formulation using the PySAT library, and the solvability of the problem is analyzed under varying conditions. The results are visualized to understand the relationship between the number of clients, delivery personnel, and time constraints on the solvability of the problem.

1 Introduction

The delivery scheduling problem is a critical aspect of logistics and supply chain management. In urban environments, traffic congestion significantly impacts delivery times, making it essential to optimize delivery schedules to meet client deadlines. This report addresses the problem of assigning delivery tasks to a set of delivery personnel (livreurs) while considering traffic congestion and operational constraints such as delivery deadlines and working hours.

The problem is formulated as a Boolean Satisfiability (SAT) problem, which is a well-known NP-complete problem. The SAT formulation allows us to encode the constraints of the delivery scheduling problem and use SAT solvers to find feasible solutions. The primary objective is to determine the conditions under which the problem is solvable and to analyze the impact of various parameters on the solvability of the problem.

2 Problem Formulation

The delivery scheduling problem can be formally defined as follows:

- Let $C = \{c_1, c_2, \ldots, c_n\}$ be a set of clients, each with a specific delivery deadline d_j and a location l_j.
- Let $L = \{l_1, l_2, \ldots, l_m\}$ be a set of delivery personnel (livreurs) available to perform the deliveries.
- Let $T = \{t_1, t_2, \ldots, t_k\}$ be a set of time slots available for scheduling deliveries.
- Each location l_j has an associated traffic congestion matrix M_j, where $M_j(t)$ represents the congestion level at time t.
- The goal is to assign each client c_j to a livreur l_i and a time slot t_k such that:
  - Each client is assigned to exactly one livreur.
  - Each livreur is assigned at most one client at any given time.
  - The delivery is completed before the client's deadline, considering the traffic congestion.
  - The total working hours of each livreur do not exceed a predefined limit (e.g., 8 hours).

3 Methodology

The problem is encoded as a SAT problem using the PySAT library. The encoding involves defining Boolean variables and clauses that represent the constraints of the problem. The following steps outline the methodology:

3.1 Data Generation

The data generation process involves creating random instances of the problem with varying numbers of clients, livreurs, and time slots. The deadlines and locations of the clients are also randomly generated. The traffic congestion matrix is constructed based on the locations and time slots.

$$\text{generate\_data}(min\_clients, max\_clients, min\_livreurs, max\_livreurs, min\_max\_time, max\_max\_time) \tag{1}$$

3.2 Congestion Matrix Construction

The congestion matrix is built based on the locations of the clients and the predicted traffic congestion levels. The congestion levels are used to calculate the time required for each delivery, which is then used to enforce the deadline constraints.

$$\text{build\_congestion\_matrix}(locations, max\_time, predictions) \tag{2}$$

3.3 SAT Encoding

The SAT encoding involves defining Boolean variables and clauses that represent the constraints of the problem. The variables $A(i, j)$ and $T(i, j, t)$ represent the assignment of client j to livreur i and the start time t of the delivery, respectively. The clauses enforce the following constraints:

- Each client is assigned to exactly one livreur.
- Each delivery starts at exactly one time slot.
- The delivery is completed before the client's deadline, considering the traffic congestion.
- No overlapping deliveries for any livreur.
- The total working hours of each livreur do not exceed the predefined limit.

$$\text{encode\_sat}(num\_livreurs, num\_clients, max\_time, deadlines, congestion\_matrix, alpha) \tag{3}$$

3.4 Solvability Check

The solvability of the problem is checked using a SAT solver. The solver is initialized with the clauses generated by the SAT encoding, and the solver's `solve` method is called to determine if a solution exists.

$$\text{check\_solvability}(num\_clients, num\_livreurs, max\_time, deadlines, locations, alpha, predictions) \tag{4}$$

4 Implementation

The implementation involves the following steps:

4.1 Data Generation

To analyze the solvability of different problem instances, we generated datasets by varying key parameters:

- extbfNumber of clients (n): Ranges from 2 to 100 in increments of 15.
- extbfNumber of livreurs (m): Ranges from 2 to 30 in increments of 5.
- extbfMaximum time available (T): Ranges from 12 to 96 in increments of 10.
- extbfInstances per parameter combination: 10 instances were generated for each combination.

For each generated instance:

1. Deadlines were randomly sampled between 1 and the maximum available time.
2. Client locations were randomly assigned among predefined options.
3. The SAT solver was used to check whether a feasible solution existed.

The solvability rate for each parameter combination was recorded and stored in a CSV file for further analysis. This dataset enables the study of how different constraints impact the feasibility of delivery scheduling problems.

4.2 Congestion Matrix Construction

The congestion matrix is constructed based on the locations of the clients and the predicted traffic congestion levels. The congestion levels are used to calculate the time required for each delivery, which is then used to enforce the deadline constraints.

4.3 SAT Encoding

The SAT encoding process involves translating the delivery scheduling problem into a Boolean Satisfiability (SAT) problem. This is achieved by defining Boolean variables and clauses that represent the constraints of the problem. The encoding ensures that all constraints are satisfied if and only if the SAT solver finds a solution. Below, we detail the implementation of each constraint.

4.3.1 Variables

Two types of Boolean variables are defined:

Assignment Variables $A(i, j)$

- Represents whether livreur i is assigned to deliver to client j.
- Each variable $A(i, j)$ is assigned a unique identifier:

$$A(i, j) = i \cdot \text{num\_clients} + j + 1 \tag{5}$$

Time Variables $T(i, j, t)$

- Represents whether livreur i starts delivering to client j at time t.
- Each variable $T(i, j, t)$ is assigned a unique identifier:

$$T(i, j, t) = \text{num\_livreurs} \cdot \text{num\_clients} + (i \cdot \text{num\_clients} \cdot \text{max\_time}) + (j \cdot \text{max\_time}) + t + 1 \tag{6}$$

4.3.2 Constraints

The constraints are implemented as clauses in the SAT problem. Each clause is a disjunction (logical OR) of literals (variables or their negations). The following constraints are encoded:

Each Client is Assigned to Exactly One Livreur

- For each client j, at least one livreur must be assigned:

$$\bigvee_{i=1}^{\text{num\_livreurs}} A(i,j) \tag{7}$$

- For each client j, no two livreurs can be assigned:

$$\neg A(i_1,j) \vee \neg A(i_2,j), \quad \forall i_1, i_2 \text{ where } i_1 < i_2 \tag{8}$$

Each Delivery Starts at Exactly One Time

- For each livreur i and client j, at least one start time t must be chosen:

$$\bigvee_{t=1}^{\text{max\_time}} T(i,j,t) \tag{9}$$

- No two start times t_1 and t_2 can be chosen for the same delivery:

$$\neg T(i,j,t_1) \vee \neg T(i,j,t_2), \quad \forall t_1, t_2 \text{ where } t_1 < t_2 \tag{10}$$

Respect Deadlines For each livreur i, client j, and time t, if the delivery cannot be completed before the deadline d_j, the corresponding $T(i,j,t)$ is forbidden:

$$\neg T(i,j,t), \quad \text{if } t + \text{congestion\_matrix}[j][t] \cdot \alpha > d_j \tag{11}$$

No Overlapping Deliveries for a Livreur For each livreur i, if a delivery to client j starts at time t, no other delivery can start during the busy period $[t+1, t+\text{busy\_time})$:

$$\neg T(i,j,t) \vee \neg T(i,j_2,t_{\text{busy}}), \quad \forall j_2 \neq j, t_{\text{busy}} \in [t+1, t+\text{busy\_time}) \tag{12}$$

Maximum Working Hours for a Livreur For each livreur i, if a delivery to client j starts at time t, no other delivery can start within the next 8 to 12 hours:

$$\neg T(i,j,t) \vee \neg T(i,j,t+k), \quad \forall k \in [8,12] \tag{13}$$

4.3.3 Summary of Clauses

The final set of clauses includes:

1. **Assignment Constraints**: Ensure each client is assigned to exactly one livreur.
2. **Time Constraints**: Ensure each delivery starts at exactly one time.
3. **Deadline Constraints**: Ensure deliveries are completed before deadlines.
4. **Overlap Constraints**: Prevent overlapping deliveries for each livreur.
5. **Working Hours Constraints**: Limit the working hours of each livreur.

These clauses are passed to a SAT solver, which determines whether a valid assignment of variables exists that satisfies all constraints. If a solution exists, it corresponds to a feasible delivery schedule.

4.4 Solvability Check

To evaluate the feasibility of different scheduling scenarios, we performed a solvability check on multiple problem instances with varying parameters. For each combination of the number of clients, number of livreurs, and maximum available time, we generated ten problem instances and assessed how many were solvable.

The solvability check involved:

- Randomly generating deadlines and delivery locations for each client.
- Constructing the SAT problem based on the generated constraints.
- Using a SAT solver to determine if a valid schedule exists.
- Recording the proportion of solvable instances for each parameter combination.

This process allowed us to analyze how problem complexity impacts solvability and to identify parameter ranges where feasible schedules are more likely to exist.

5 Results

The results of the solvability check are analyzed to understand the relationship between the number of clients, livreurs, and time constraints on the solvability of the problem. The results are visualized using line plots and heatmaps to show the solvability rate under different conditions.

5.1 Line Plots

The line plots depict the solvability rate as a function of a single parameter while averaging over the other two parameters. This approach isolates the impact of each variable on the solvability rate.

5.1.1 Effect of Number of Clients

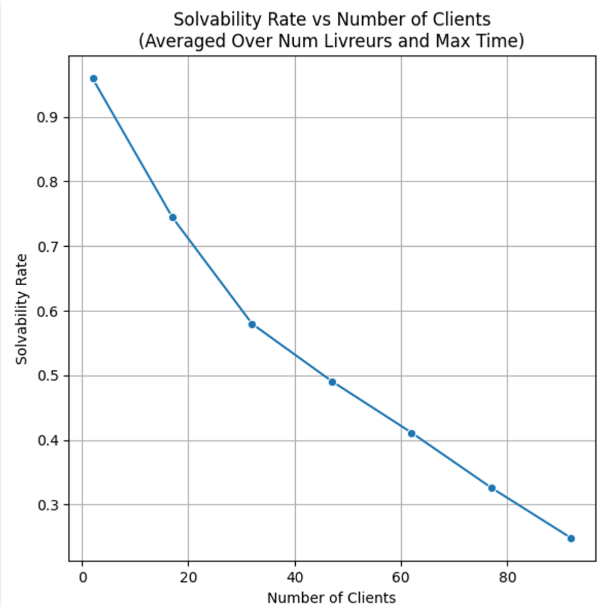

Figure 1: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

The first plot shows that the solvability rate decreases as the number of clients increases. This trend is expected, as more clients introduce additional constraints, increasing the complexity of the problem. With more clients, livreurs must cover more locations within limited time constraints, making it harder to find feasible schedules.

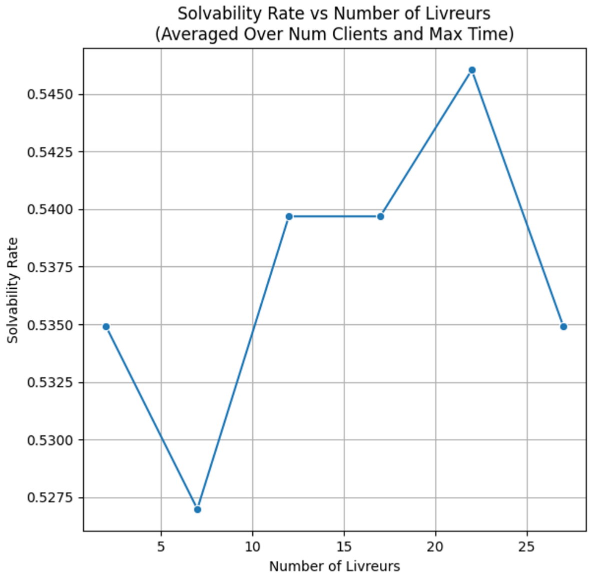

Figure 2: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

5.1.2 Effect of Number of Livreurs

The second plot illustrates the relationship between the number of livreurs and the solvability rate. Unlike the number of clients, this relationship does not exhibit a strictly increasing trend. While adding more livreurs generally improves solvability, diminishing returns appear beyond a certain point, likely due to constraints imposed by deadlines and limited time slots.

5.1.3 Effect of Available Time Slots

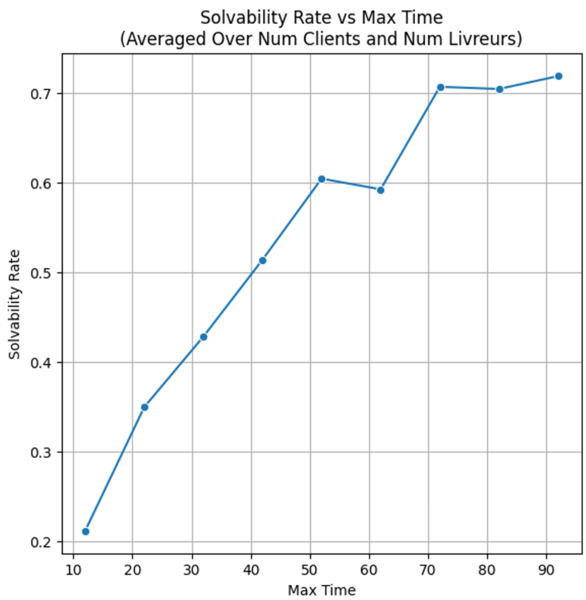

Figure 3: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

The third plot demonstrates that increasing the available time slots significantly improves the solvability rate. A longer scheduling horizon allows more flexibility in assigning deliveries, reducing conflicts and enabling more feasible solutions.

5.2 Heatmaps

The heatmaps provide a more detailed view of how pairs of parameters jointly influence the solvability rate, averaged over the third parameter.

5.2.1 Interaction Between Number of Clients and Livreurs

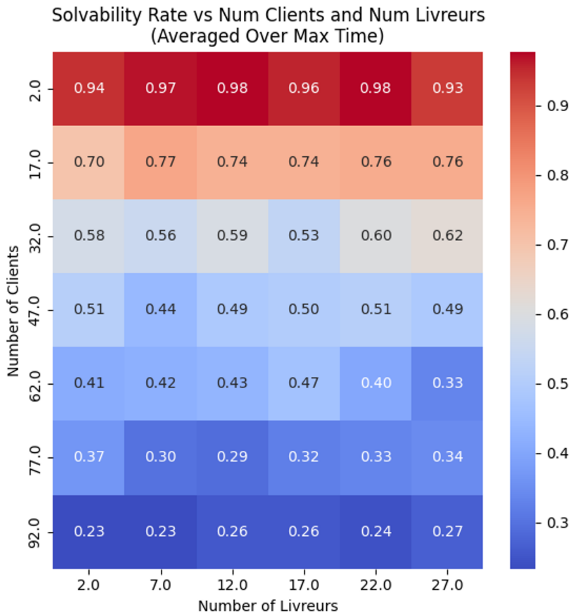

Figure 4: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

The first heatmap shows that solvability is highest when the number of clients is low and the number of livreurs is high. However, as the number of clients increases, adding more livreurs mitigates but does not entirely prevent the drop in solvability. This suggests that while increasing the workforce helps, it cannot fully counteract the complexity introduced by a growing client base.

5.2.2 Interaction Between Number of Livreurs and Available Time Slots

The second heatmap reveals that when the number of livreurs is low, increasing the available time slots significantly boosts solvability. However, when the number of livreurs is already high, additional time slots provide less benefit, indicating that livreur availability becomes the limiting factor in such cases.

5.2.3 Interaction Between Number of Clients and Available Time Slots

The third heatmap shows that solvability decreases as the number of clients increases, but the impact is softened by extending the available time slots. This highlights the importance of scheduling flexibility in handling a high number of delivery requests.

5.3 Summary of Findings

From the visualizations, we conclude:

- Increasing the number of clients decreases solvability due to increased scheduling complexity.
- Increasing the number of livreurs improves solvability, but only up to a certain threshold beyond which constraints like deadlines become dominant.
- Increasing available time slots significantly enhances solvability, particularly when livreur availability is limited.
- The interplay between these factors suggests that optimizing delivery feasibility requires balancing workforce allocation and time constraints.

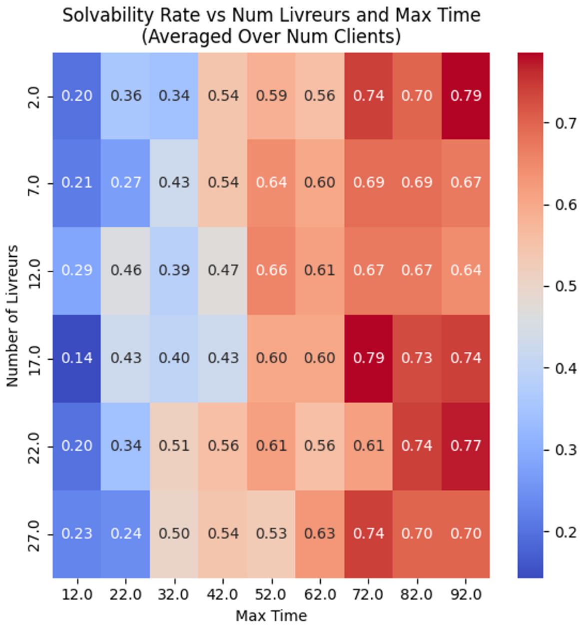

Figure 5: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

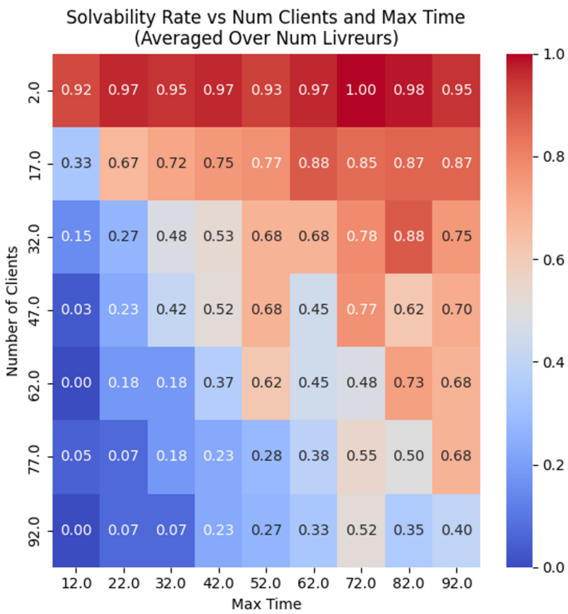

Figure 6: Solvability rate of the allocation problem as a function of (n deliverers, n clients, max time)

6 Conclusion

This report presents a comprehensive study on the optimization of delivery scheduling under traffic congestion constraints. The problem is formulated as a SAT problem, and the solvability of the problem is analyzed under varying conditions. The results provide insights into the relationship between the number of clients, livreurs, and time constraints on the solvability of the problem. The visualizations help in understanding the conditions under which the problem is solvable and can be used to guide decision-making in logistics and supply chain management.