Improving Fire Emergency Response in New York: Binary Optimization Strategies and Pre-Assessment of Resources

Ferrando Muñoz, G., Lanza Méndez, D.A., López Fresco, J., Lozano Gómez, S., Sánchez Gil, M.

Degree on Data Science. Universitat Politécnica de València.

Abstract— This research introduces an Integer Linear Programming model to optimize NYC's fire emergency response, considering vehicle allocation and station capacities. Extensive experiments demonstrate the model's effectiveness in reducing response times. Implementing ORTools, the study shows a marked improvement in resource utilization. The results indicate a substantial potential for enhanced public safety through strategic vehicle distribution, providing valuable insights for future urban emergency response planning.

I. INTRODUCTION

Fire emergencies pose a significant threat to both lives and property, necessitating efficient emergency response services. The Fire Department of NYC, serving its five major boroughs (Manhattan, Brooklyn, Bronx, Queens, and Staten Island), plays a vital role in addressing not only fires but also chemical spills, rescues, and medical emergencies. With a fleet of 354 specialized vehicles, distributed across 218 stations and organized into five companies (Engine, Ladder, Rescue, Squad, and Hazmat), the department is well-equipped for diverse emergencies.

This article focuses on optimizing the location of these vehicles using Integer Linear Programming to enhance response efficiency. The goal is to strategically position the vehicles considering factors like neighborhood density and incident frequency. The methodology involves developing an optimization model, implementing it using ORTools, a Python library, and evaluating its temporal scalability with different solvers. The article aims to demonstrate how this optimized distribution can improve the department's emergency response and maintain the city's welfare. Following the introduction, the article will detail the review of related research, the mathematical formulation of the model, its implementation and will conclude with a experimentation.

II. RELATED WORK

This article [1] examines disaster management, focusing on the allocation and distribution of resources in emergencies, and for this reason, it has influenced us in the creation of the scorings and the structure of the presented model. Experiments show that certain disaster management problems are efficiently solved using the simplex algorithm. However, since our model uses binary variables, the CBC solver may be more suitable. The goal is not to find the optimal location of stations, but an approach that improves the allocation of resources to the stations, and in this case, CBC is more effective.

The article [2] deals with the problem of locating and resourcing emergency stations for oil spills in the Canadian Arctic. For this purpose, an MIP is developed with many similarities to our task, although the approach is based on incident coverage rather than resource allocation, which would be our case. The maximization of the objective function, the assignment of weights, and the selection of variables have been very useful in refining the details of our model and calculating the scorings.

The study [3] proposes the use of evolutionary algorithms to solve the complex allocation of timetables in university environments, addressing the efficient distribution of resources such as teachers, subjects and classrooms. This related work can help us greatly in the mathematical formulation part of the problem, as it is conceptually similar to our problem. Regarding the optimisation part, it is not useful to us as it uses other techniques that we are not familiar with.

The article [4] discusses optimizing fire station locations in Texas, aiming for a 4-minute emergency response time. It uses a circular area model around each station to ensure emergencies within these areas are reached within this timeframe. The article's approach, differs from the case study in question. It employs an NLP technique with real-value coordinates as decision variables, which isn't applicable to the case study. Additionally, the method of assessing service quality based on response time, though interesting, is not relevant to the case study..

This study [5] was chosen as it involves a linear optimization problem related to vehicle assignment and is tested using data from NYC. The problem at hand is the assignment of shared taxis with clients. Regarding similarities with our project, the model is also an integer linear optimization, and the objective function also utilizes binary variables. It differs in that the variables are less complex, as they do not take into account shift changes or vehicle types, and it has the sole constraint of vehicle capacity. They also use scoring, but in their case, it consists of only two variables and is designed as a cost to minimize, whereas we maximize our score.

III. PROPOSED MODEL

A. Exploratory Data Analysis

After obtaining timely knowledge about the envisaged problem, an extensive exploratory and descriptive analysis of the data has been carried out in order to identify the aspects that need to be taken into account for the modelling.

The first step was to find the differences between the two shifts (Day and Night) to see if a different allocation was necessary for each shift. This would result in the acquisition of an important factor, which would be added in the decision variables and, consequently, in the rest of the elements. Fig. 1 shows two graphs of incident duration times for day and night.

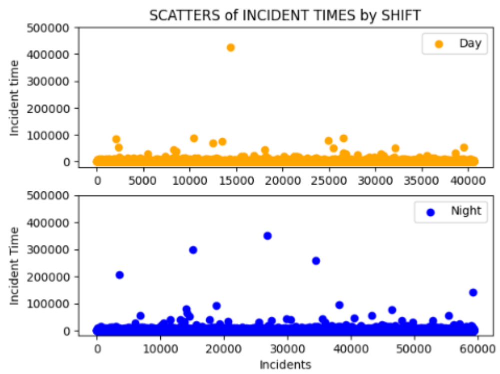

Fig. 1 - Incident duration times for day and night.

There are differences both in the number of incidents, with more incidents occurring at night, and in the duration, with incidents taking slightly longer to control at night. As can be seen in the following graph, which corresponds to the number of vehicles used on each shift, there are significant differences, although it would be expected that if there were more incidents, the number of vehicles would also be greater. However, the relationship is not directly proportional, as fires are more dangerous at night and require more help. Calculating the ratio of vehicles to incidents, we see that at night it is 1.9 and during the day it is 1.84, so we do find differences.

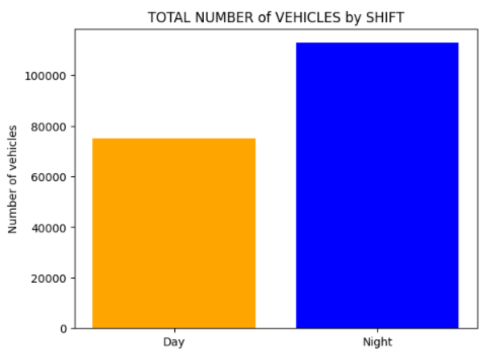

Fig. 2 - Total number of vehicles by shift.

Using these three measures, we now look at these differences broken down by district. Looking at the Fig. 3 and Fig. 4 we see two very different behaviours, Manhattan is more prone to have incidents during the day while Brooklyn and Queens at night, being in the respective shifts the boroughs with the highest number of incidents. However, the borough where the ratio is higher in both shifts is the Bronx, which may indicate that it is a borough whose accidents are somewhat more dangerous or difficult to extinguish.

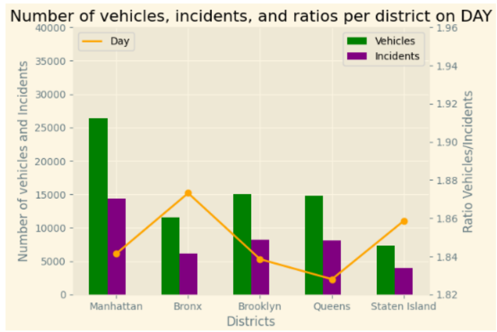

Fig. 3 - Vehicles, incidents and ratios per district on day.

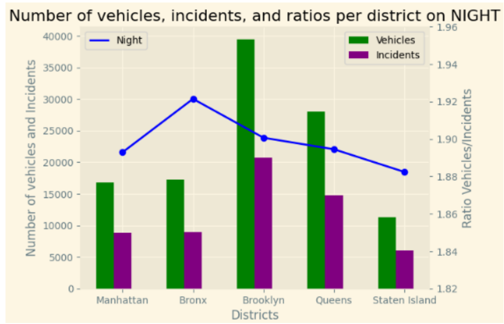

Fig. 4 - Vehicles, incidents and ratios per district on night.

A breakdown by vehicle also shows differences. In this case we show in Fig. 5 the percentage of each type of vehicle with respect to day and night. It turns out that Ladder, being the type with the highest number of vehicles, has the expected distribution coinciding with the proportion of incidents between each shift. The same is not true for Squad and Rescue, which are mostly used at night, and Hazmat is used more during the day, although taking into account the proportion of incidents. We have therefore decided to make two different allocations depending on the shift, as the differences found in this analysis are significant and need to be taken into account, assigning a different score depending on whether the vehicle is used during the day or at night.

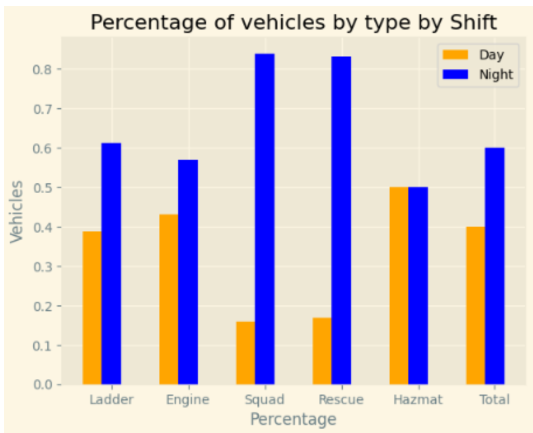

Fig. 5 - Percentage of vehicles by type and shift.

B. Scoring Calculation

The scoring S_{ijk} is a value that multiplies each variable X_{ijk} to increase or decrease the probability of being included in the model. The scoring S_{ijk} is composed of "sub-scorings" extracted from the JSON files provided in the Jupyter Notebook of the project. The extraction process is as follows:

For a vehicle of type i assigned to station j in shift k, we obtain the centroid of the neighborhood closest to the station, which we assume is the station's neighborhood. Subsequently, we calculate the population density of this neighborhood, so that the higher the density, the final score will increase. Once this is done, we calculate how many nearby stations (those within a certain number of seconds) there are from station j, so if there are fewer nearby stations, the final score will increase. We think this is reasonable as we consider an isolated station will have to be on service more frequently than one that is near other stations. We also check the capacity of station j, with lower capacity resulting in a higher final score, as we aim to fill stations with lower capacity first. To determine which neighborhood the vehicle i will go to when there is a shift change, we look at the number of incidents for day (Fig. 6) and night (Fig. 7) in the nearby neighborhoods (again, those within a certain number of seconds) to station j, and the higher the number, the higher the final score, increasing the probability of the vehicle being assigned. Finally, for vehicle types, the fewer there are, the higher the final score.

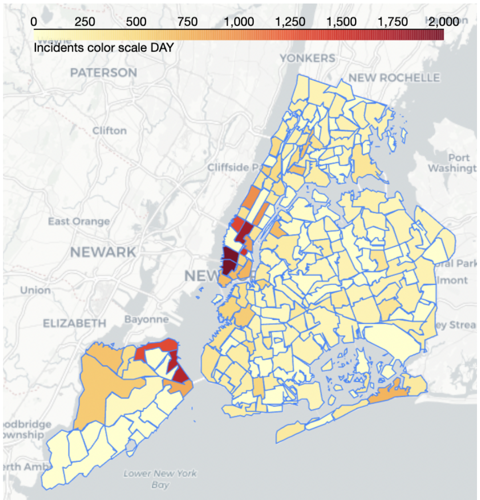

Fig. 6 - Neighborhoods Incidents on day.

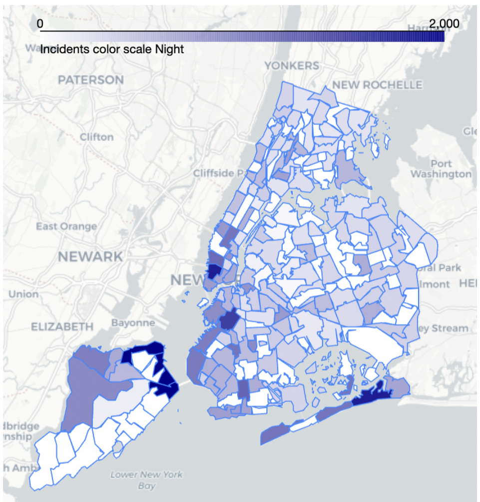

Fig. 7 - Neighborhoods Incidents on night.

Once the scores are extracted, we normalize each one using Min-Max. Then, we multiply these normalized scorings by fixed weights and sum them to form the final scoring, as shown in the Input parameters formula. The weights considered to multiply each score are shown in table 1.

TABLE 1
SCORING IMPORTANCES

Scoring	Importance
Nearby Neighborhood j	0.25
Density j	0.05
Capacity j	0.15
Nearby Stations j	0.2
Shift Incidents k	0.25
Vehicle i,k	0.1

Where the sum of these weights is equal to 1.

Afterwards, the scores are again normalized by min-max and ready to use with the model. In light of this, and in order to visualise how the definitive scores are distributed, we can take a look at the Fig. 8.

This almost-normal distribution implies that the design of the scores is well-constructed and effectively distributed to help improve the resource allocation.

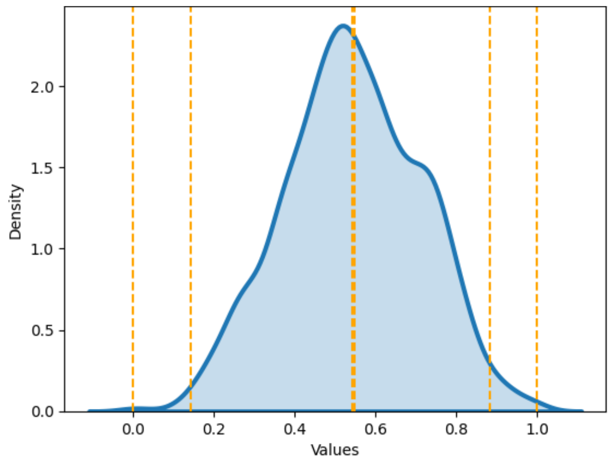

Fig. 8 - Final scores distribution.

C. Model Especification

Conceptually, the objective is to build a model to determine at which stations the different types of existing emergency vehicles are allocated. The proposed model attempts to place vehicles at stations in such a way as to maximize service to neighbourhoods.

Maximise: Serving neighbourhoods.

Subject to:

- The total number of vehicles assigned at a station must be a minimum of 1 and a maximum of the station's capacity.
- The number of assigned vehicles of each type must be exactly the number of vehicles of each type available.
- No more than one vehicle of any one type may move from a station.
- A maximum of one vehicle of each type may arrive at the station.

- For each vehicle type and station: Shift2 = Shift1 -exits + entrances.
- There cannot be more than two vehicles of type 3 in the same district.
- There may be a maximum of 3 type 4 cars in each district and a maximum of 1.
- Movements may not be further than 780 seconds apart.

Mathematically, an IP formulation of the optimization model for the allocation of vehicles at the stations is shown below.

1) Decision Variables: With regard to the decision variables, two types of binary variables are available, one to control the allocation of the different types of vehicles to the different stations in the given shift, and the other to manage the changes of vehicles between stations from one shift to another.

$X_{i,j,k}\{1: if\ vehicle\ type\ i\ is\ at\ station\ j\ on\ shift\ k.\ 0\ otherwise.$

Binary. $X_{i,j,k} = \{0,1\}$, $X_{i,j,k} \geq 0$, i = 1…5, j = 1…219, k = 1, 2 (1)

$Y_{i,j,k}\{1: if\ vehicle\ type\ i\ is\ relocated\ from\ station\ j\ to\ k.\ 0\ otherwise.$

Binary. $Y_{i,j,k} = \{0,1\}$, $Y_{i,j,k} \geq 0$, i = 1…5, j = 1…219, k = 1…219, j != k (2)

2) Constants:

TABLE 2
CONSTANTS AND EXPLANATION

Constant	Explanation
J	Set of stations, |J| = 219
K	Set of shifts, |K| = 2
I	Set of vehicles types, |I| = 5
N	Set of Neighbourhoods, |N| = 195
D_i	District i, each has a group of associated neighbourhoods, i = 1…5 $\cap_{i=1}^{5} D_i = \phi$, $\sum_{i=1}^{5} \lvert D_i \rvert = 195$
$S_{i,j,k}$	Scoring applied to vehicle type i at station j and shift k, $\forall\ i, j, k: i\epsilon I,\ j\epsilon J,\ k\epsilon K$
C_j	Capacity of station j, $\forall\ j: j\epsilon J$
T_i	Maximum number of vehicles of type i, $\forall\ i: i\epsilon I$
$R_{j,k}$	Distance in seconds from station j to station k, $\forall\ j, k: j\epsilon J,\ k\epsilon K$

3) Objective function and constraints: An objective function formulation consisting of a single equation (3) maximizes the weighted allocation of vehicles at stations.

$$Max\ Z: \sum_{j=1}^{|J|} \sum_{k=1}^{|K|} \sum_{i=1}^{|I|} \left(S_{i,j,k} *\ X_{i,j,k}\right) \ (3)$$

Subject to:

$$1 \leq \sum_{i=1}^{|I|} X_{i,j,k} \leq C_j\ , \forall\ j, k:\ j\epsilon J,\ k\epsilon K\ [bound\ vhs\ station\ j] (4)$$

$$\sum_{j=1}^{|J|} X_{i,j,k} =\ T_i\ , \forall\ i, k: i\epsilon I,\ k\epsilon K\ [all\ vehicles\ assigned]\ (5)$$

$$\sum_{k=1}^{|J|} Y_{i,j,k} \leq X_{i,j,1}\ , \forall\ i, j: i\epsilon I,\ j\epsilon J\ [movement\ of\ vehicles]\ (6)$$

$$\sum_{j=1}^{|J|} Y_{i,j,k} \leq 1\ -\ X_{i,j,1}\ , \forall\ i, k: i\epsilon I,\ k\epsilon J\ [arrivals\ of\ cars]\ (7)$$

$$X_{i,j,2} = X_{i,j,1} - \sum_{k=1}^{|J|} Y_{i,j,k} + \sum_{k=1}^{|J|} Y_{i,k,j}\ , \forall\ i, j, k: i\epsilon I, j, k\epsilon J\ [equal\ shifts\ vh\ i]\ (8$$

$$\sum_{j}^{D_i} X_{3,j,k} \leq 2 \quad \forall\ i, k: i\ =\ 1...5,\ k\epsilon K\ [district\ i\ vh\ 3]\ (9)$$

$$1 \leq \sum_{j=1}^{D_i} X_{4,j,k} \leq 3\ \ , \forall\ i, k: i\ =\ 1...5,\ k\epsilon K\ [district\ i\ bound\ vh\ 4]\ (10)$$

$$Y_{i,j,k} * R_{j,k} \leq 780\ \ , \forall\ i, j, k: i\epsilon I,\ j, k\epsilon J\ [max\ vh\ movement]\ (11)$$

With constraint (4) we ensure that all stations have at least one vehicle and at most the value corresponding to their capacity. With constraint (5) we ensure that in each shift all vehicles are allocated. With constraints (6) and (7) we ensure that only a maximum of one vehicle of each type will depart and arrive at a station. With constraint (8) we ensure that at shift change no extra vehicles are added or removed. With restriction (9) and (10) we ensure that vehicle types 3 and 4, of which there are few units, are equally distributed. With constraint (11) we ensure that vehicle shifts will not involve excessively long shifts according to the 780 seconds criterion.

IV. IMPLEMENTATION DETAILS

Regarding the model implementation, it has been done in an .ipynb file, the model creation code is divided into four main parts:

The first one is the creation of the scores for each vehicle, station and shift.

The second part is the creation of variables and the objective function. Two types of variables, X_{ijk} and Y_{ijk} are created. The first type of variables indicates whether a vehicle of type i is assigned in station j in shift k, in total there are 2190 variables of this type.

On the other hand, the variables of the family Y_{ijk} indicate that a vehicle of type i moves from station j to station k in the shift change from night to day, at the end of the day it returns to the station of origin. In total 238710 variables of this type have been created.

Next in the creation of the objective function, the first step is to create it to maximise, and we add each variable X_{ijk} with its coefficient being the corresponding score.

The third part of the code is the creation of constraints: The constraints in the family (4) account for a total of 438 constraints.

The next constraints that have been implemented (5) is that the assigned vehicles have to be exactly the ones available. In total there are 10 such constraints.

The next step was to create the departure (6) and arrival (7) constraints. A vehicle of one type can only leave a station if in turn 1 it was assigned and only 1 can leave, just as only one vehicle can arrive at a station if there was none of that type, and the maximum that can arrive is one. In total there are 1095 departure restrictions and 1095 arrival restrictions.

Next, we create restrictions so that for each vehicle type and station, the number of vehicles in turn 2 must be equal to the number of vehicles in turn 1 minus departures plus arrivals (8). In total there are also 1095 such constraints.

Constraints (9) and (10) serve to ensure that the distribution of vehicles type 3 and 4 is equal. One constraint has been created for each district per shift and vehicle type. Knowing that there are 2 shifts, the type of vehicles for these constraints are 3 and 4, and that the number of districts is 5, the total number of restrictions created is 20.

The last family of constraints (11) is to ensure that location changes do not take more than 780 seconds. The total number of constraints of this type is 238.710.

In total, the model we are going to work on is made up of 240.900 variables and 242.463 constraints.

Finally, the fourth part of the code serves to solve the model and display the optimal result.

The result shown are the assignments and movements to be carried out in the shift change between night and day.

V. EXPERIMENTS

Once the model has been mathematically defined, as well as the decision variables, the objective function and the restrictions have been implemented in Python, all that remains is to solve the problem, that is, to find a solution, not just any solution, but the optimal solution that maximises the objective function.

To find the solution to an optimisation problem, ORTools provides a variety of solvers that can be based on different algorithms, such as the Simplex algorithm. However, solvers cannot solve all types of scenarios. In our case, the problem has been modelled using an Integer Linear Programming technique, so a solver capable of solving this type of scenario has to be used. For this purpose, ORTools provides solvers such as SCIP, CBC and BOP.

Knowing which solvers we can use, now the question is which of them is able to find an optimal solution in the shortest possible time? This is an interesting question. Since our research has as a case study a serious, real-world case, obtaining a solution as quickly as possible is fundamental to be able to start putting this result into practice, that is, to start with the implementation of this result, because in this way the necessary changes would be made to start providing an optimised service capable of responding more effectively to the city's emergencies.

The only way to know which solver is able to find an optimal solution in the shortest possible time is simply by testing each of the solvers and comparing them. In this case, an experiment has been carried out to study the time scalability of the model with the solvers SCIP, CBC and BOP, which have already been mentioned before.

The experimentation carried out basically consists of solving the problem 50 times with each solver, saving the solution times obtained. In other words, the problem was solved a total of 150 times: 50 times with the SCIP solver, 50 times with the CBC solver and 50 times with the BOP solver. Subsequently, the average of the 50 solution times obtained with each solver was obtained, resulting in an average of 33.62s with the SCIP solver, an average of 10.26s with the CBC solver and an average of 25.87s with the BOP solver (Fig. 9). We can clearly see that the CBC solver takes by far the shortest time to find an optimal solution to the problem.

Despite the large and more than clear difference between the three solvers with respect to the time taken to find a solution, a statistical significance test was also carried out in order to confirm with full confidence that there is a difference between the average solving times of each solver. Prior to performing the test, the Shapiro-Wilk test was used to check that the samples, i.e. the 50 solving times for each solver, did not follow a normal distribution, except for the sample corresponding to the SCIP solver. Using Levene's test, it was found that there was also no equality of variances between samples, except between the CBC and BOP samples. Having checked this, we decided to use Welch's t-test to compare the average resolution times as this test is less sensitive to samples not following a normal distribution and also to differences between variances. That said, the result obtained by applying this test was that there are differences between the average resolution times of SCIP and CBC, between the average resolution times of SCIP and BLOP and, finally, between the average resolution times of CBC and BOP.

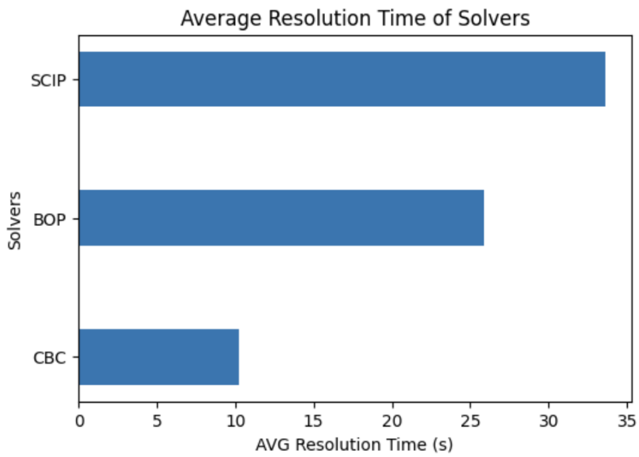

Fig. 9 - Average resolution time of solvers.

Therefore, we can safely confirm that the CBC solver takes the shortest time to find a solution to the problem, so it is the solver that should be used to obtain the fastest optimal solution and start implementing changes to the service.

VI. CONCLUSIONS

This study introduced a novel integer linear programming model for optimizing the allocation of specialized vehicles at fire stations in New York City, aiming to enhance efficiency in fire emergency response.

Our analysis indicates that strategic resource distribution significantly impacts the city's emergency response capabilities.

- Effective Optimization: The model effectively optimizes resource allocation, considering population density, incident frequency, and station capacities.
- Improved Emergency Response: Implementing this model suggests notable improvements in emergency response times, critical for saving lives and property.
- Advanced Technology Application: Utilizing ORTools and integer linear programming techniques provides an innovative and efficient approach to emergency resource management.
- Scalability and Adaptability: The model is scalable and adaptable, indicating its applicability in various emergency situations and contexts.
- Performance Analysis: Experiments confirm the model's efficacy, especially with the CBC solver,

which yielded the best results in terms of time and optimal performance.

In conclusion, this study significantly contributes to understanding and enhancing urban emergency management. The insights offer valuable perspectives for decision-makers in the NYC Fire Department and potentially, global counterparts. Implementing this model promises to improve operational efficiency, public safety, and overall emergency management in NYC.

VII. REFERENCES

[1] B. Bodaghi y N. Sukhorukova, "Solving multiresource allocation and location problems in disaster management through linear programming", 2018.

[2] T. Das, F. Goerlandt, y R. Pelot, "A mixed integer programming approach to improve oil spill response resource allocation in the Canadian arctic", Multimodal Transportation, vol. 3, núm. 1, p. 100110, 2024.

[3] J. M. M. Caballero y C. P. Arboleda, "Asignación de horarios de clases universitarias mediante algoritmos evolutivos", Rev. Educ. Ing., vol. 5, núm. 9, pp. 140–149, 2010.

[4] J.-Y. Oh, A. Hessami, y H. J. Yang, "Minimizing response time with optimal fire station allocation", Stud. Eng. Technol., vol. 6, núm. 1, p. 47, 2019.

[5] A. Simonetto, J. Monteil, y C. Gambella, "Real-time city-scale ridesharing via linear assignment problems", Transp. Res. Part C Emerg. Technol., vol. 101, pp. 208–232, 2019.

Endnote about Related Works:

- Article 1: Jorge López Fresco.
- Article 2: Samuel Lozano Gómez.
- Article 3: Marc Sánchez Gil.
- Article 4: Dylan Lanza Méndez.
- Article 5: Guillermo Ferrando Muñoz.