Smart Passenger Center: Real-Time Optimization of Urban Public Transport

Alfonso Corrado, Simona Barba, Isabel Carozzo and Simone Nardi
MerMec Engineering srl
Via Livornese 1019, 56122, Pisa, Italy
{alfonso.corrado, simona.barba, isabel.carozzo, simone.nardi}@mermec-engineering.com

Abstract

This paper introduces the Smart Passenger Center (SPaCe), MerMec's solution for real-time optimization of urban public transport. The proposed system, based on cutting-edge artificial intelligence technologies, provides real-time vehicle status information and passenger activity monitoring. All of the information acquired from the vehicles is used to optimize the performance of the vehicle fleet for real-time traffic management. The purpose of the paper is to explain the reasons and benefits of such Intelligent Transport System (ITS) with a deep understanding of its architecture and advanced functionalities. Additionally, a practical application of the data collected by SPaCe is proposed for the task of bus timetable optimization. The goal is to schedule bus trips in a way that maximizes multiple conflicting goals such as service quality and operating costs, based on real time collected data. The reported results show how the developed optimization system can support the decision-making process to balance the interests between passengers and public transport agencies.

Introduction

In recent years, urbanization and population growth have led to an increase in demand for public transport, placing significant pressure on existing systems. It is estimated that an average of 40% of the world population spends at least one hour on the road every day (Zhang et al. 2011). Issues such as overcrowding, delays, and unreliable services have become commonplace, leading to dissatisfaction among passengers and reducing the attractiveness of public transport as a viable option for commuters. Increasing the number of infrastructures will not be sufficient to solve this issues, instead a new strategic plan will need to be implemented (Needs 2012). Public transport will have to become a more integrated system in which information, management, and control work in synergy. Traffic flows will need to be rearranged, in order to encourage a balance between the various transport modes. (Mangiaracina et al. 2017).

The European Union defined Intelligent Transport Systems (ITS) as an advanced set of applications that provide innovative services for traffic management. ITS combines transport engineering with telecommunications, electronics, and information technology. The primary goals of ITS are to improve capacity, efficiency, and safety, as well as reduce energy consumption and environmental impact (European Union 2010).

Copyright © 2023 by the authors. All rights reserved.

Figure 1: Privacy by design: the video streams are processed on edge and no image is stored permanently in all operations involving sensitive personal data such as seat counts or passenger analytics

To the authors best knowledge, there is currently no off-the-shelf single solution that can fulfill all the intended goals of an ITS, by gathering and analyzing data in real-time while enhancing various aspects of public transport. Today's systems offer management recommendations based on transport load projections derived from historical data, public surveys, or the expertise of the network administrators. They typically use urban mobility simulators, such as SUMO (Lopez et al. 2018), to assess the impact on vehicle fleet performance of changing passenger volumes and traffic conditions. However, a system that uses only data simulations of the past is unprepared to manage unforeseen circumstances, including route issues or road closures, making it only suitable for long-term planning. On the other hand, on-vehicle surveillance systems are already capable of collecting raw data from sensors that could be used to optimize transport, adjust supply to demand, and improve maintenance, provided that a dedicated operator analyzes the videos received by the cameras 24 hours a day.

The major transport industrial players are beginning to test out strategies to improve urban transport on the existing infrastructure. Thanks to the modern Artificial Intelligence (AI) technologies and the continuous evolution of communication systems, several trials for real-time people flow analysis are being conducted. There are two most important examples in this sector: the MASTRIA system (developed by Alstom), and the NAIA system (developed by Thalesgroup). Both systems are not ready for the market, they are in the experimental phase, not tested in real envi-

ronment.

Within this context, MerMec introduces the Smart Passenger Center (SPaCe). The proposed system is able to analyze vehicle status and passenger activities in real time and provide suggestions for fleet management.

A better understanding of passenger habits is the key to designing a travel experience that meets their needs and improves their experience. In this way, public transport becomes attractive for users and profitable for agencies. Indeed, predicting how people will travel and behave within the infrastructure helps streamline operations, timely match capacity to demand, optimize stop selection, and identify new revenue opportunities. Efficiency, organization, and passenger satisfaction are the main goals of any modern system, and the SPaCe system helps operators achieve them.

Smart Passenger Center

SPaCe is a video analytics system that helps the transport industry improve transit operations and passenger experience, prevent security incidents, and improve post-event investigations. Moreover, SPaCe is an integrated system of on-board analytics, communication technologies, services that automate the control and optimize the decision processes. The system is designed to support supervisors, field operators, and passengers, and it is suited to control public transport and optimize vehicle fleet distribution. It is a decision support tool that assists operators in managing passenger flow within stations and vehicles.

The main features of SPaCe are:

- optimize public transport resources by predicting and adjusting for fluctuations in passengers volume;
- deploy of new automatic surveillance technologies for the safety of infrastructures, vehicles and passengers;
- promote the use of public transport to reduce the environmental impact.

To address these tasks, SPaCe incorporates various AI techniques handling large amounts of data generated by vehicles and passengers.

Architecture Overview

The SPaCe system is composed of three main components: Peripheral Component, Server Component and User Component. Each component performs different but interconnected operations, in particular:

1. **Peripheral Component** deals with the structure of the peripheral SPaCe system, both on board the vehicle and in the stations. It gets data from sensors, does preliminary processing and anonymization, and then distributes the information to the servers and users.
2. **Server Component** has the responsibility of gathering data from peripheral nodes and providing synthetic information on the status of people, vehicles, and stations. It also provides a standardized API for the User Component, offering a level of abstraction from the data structures of the system domain. Finally, it monitors the status of the SPaCe system, continuously checking its correct operation.

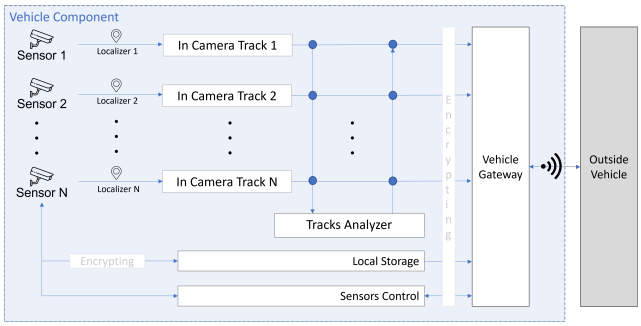

Figure 2: Peripheral Component software architecture

3. **User Component** shows relevant information and analysis results based on user type (i.e. passengers, operators or supervisors) and purpose (i.e. transport status check, optimization, safety, maintenance).

The three components are logically interconnected and must continuously exchange information. The distribution of information between the SPaCe modules takes place through a Data Distribution Service (DDS). SPaCe uses the middleware provided by ROS2 to integrate the DDS between the various subsystems (Macenski et al. 2022). For the sake of brevity, this paper focuses on a description of Peripheral and Server Components only.

Peripheral Component

The Peripheral Component is interfaced to multiple sensors to monitor the environment, including 2D (e.g. RGB cameras) and 3D (e.g. stereo cameras or Lidars) sensors. Each sensor has a dedicated processing pipeline that communicates with the calculus modules. The Peripheral Component hosts the analysis modules for the extraction of the information used by the SPaCe system and necessary for the anonymization of the entire data flow. The architecture of the Peripheral Component is shown in Figure 2. The two main modules, *In Camera Track* and *Track Analyzer*, are described below.

In Camera Track The module detects the elements (people and objects) within the scene and reconstructs the space-time consistency between the identified elements. It builds a *track* by following the movement of each element, each time the element is in the field of view of the camera. Then, it delivers the computed tracks to the *Track Analyzer* module. The main steps of this module are:

- **Object Detector** classifies and locates all elements in the image, which is useful to identify people and objects in the scene. This task is performed through convolutional neural networks (CNNs) with supervised learning models, e.g., YOLOv7 (Wang, Bochkovskiy, and Liao 2022).
- **Object Tracker** performs single-camera object tracking. A unique identification number (ID) is assigned to all elements in a given frame. The same element is recognized in subsequent frames, carrying forward the assigned IDs until the element leaves the camera view. The algorithm

used is based on a combination of Kalman Filters and Hungarian Matching algorithms (Bewley et al. 2016).

Track Analyzer The module is composed of multiple specialized microservices that provide details for each detected track and reconstruct the information necessary for the subsequent analyses. Each track is related to a single element which is visible in the scene for a certain time interval. The track keeps the set of data extracted by each microservice for a specific element in the given time interval.

In the following a subset of microservices available for person analysis are reported:

- **Face Features Extractor** service extracts the face's biometric features. It searches for a face in the given track, extracts the crop containing the face, and encodes the crop in a numerical representation. The encoded features are then used to search within a database to check if the subject has already been seen in the past. Face encoding and recognition frameworks are used, e.g., Dlib C++ Library (King 2009).
- **Person Features Extractor** service extracts the body's appearance features associated with visible clothes and accessories (e.g., the main colors of the clothes, if it has a trolley, suitcases, or backpacks, etc.). Due to the fact that faces are not easily encoded, the entire person figure is examined and unique characteristics are extrapolated. A robust appearance model of the entire figure is built. Person features extractor frameworks are used, e.g., FastReID (He et al. 2020) customized on the Bag of Tricks method (Luo et al. 2019), which is a strong baseline in the Re-ID task.
- **Human Pose Estimation** service extracts the skeleton key points of the human body. The skeleton is subsequently evaluated to detect human actions and behaviors. Human pose estimation frameworks are used, e.g., MoveNet available on TF Hub (Google 2021).

Server Component

The Server Component processes, organizes, and stores all the information coming from vehicles and stations. These activities are performed by the *System Domain*, which is divided into three main modules: *Person Domain* deals with people data, *Status Domain* deals with the environment around people and vehicles, and *Transit Domain* deals with vehicle data. Moreover, the Server Component performs *Advanced Analytics*, which process raw data through artificial intelligence and numerical optimization methods in order to obtain composite results. The architecture of the Server Component is shown in Figure 3 and described below.

Person Domain The module monitors people's behavior and tracks them along their journeys on public transport. The most crucial and challenging task here is to track people when they pass across multiple non-overlapping cameras; this task is known as person re-identification (Re-ID) (Ye et al. 2021) (see Figure 1). In order to determine which tracks belong to the same person, a clustering algorithm on both face and body features is used. The clustering algorithm is based on the DenStream (Cao et al. 2006), a generalization

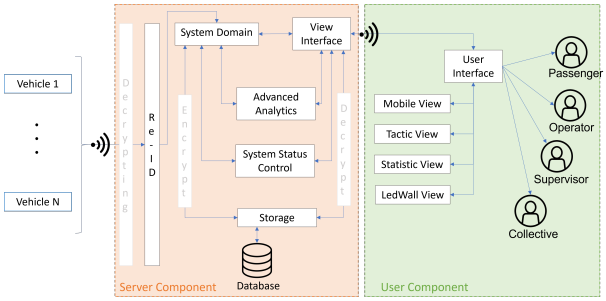

Figure 3: Server and User Components software architecture

of the well-known DBSCAN (Ester et al. 1996) designed to manage stream data. Re-identification between various tracks is driven by a face feature matching algorithm and refined by a body feature matching algorithm when faces are not available.

Status Domain The module monitors vehicle status in order to check for damage or vandalism, report abandoned objects, and promptly detect fire and smoke (see Figure 4). The module is strictly related to the Person Domain module in order to create virtual links between the environment status changes and the people involved.

Transit Domain The module tracks vehicles and controls the vehicle fleet. It handles input-output compatibility with respect to the General Transit Feed Specification (GTFS) (Google 2022), a standard format for public transport schedules and associated geographic information. The Transit Domain uses dedicated interfaces based on a SQL database to interact with GTFS. The Transit Domain, based on vehicle positions, determines when a vehicle has arrived at a stop. At each stop, the system computes the number of boarding and alighting passengers. The collected data is then used to assess the performance of the service and suggest adjustments to optimize it.

Advanced Analytics A collection of microservices which perform advanced analysis on collected data, such as real-time identification of human behavior, and real-time evaluation and improvement of the performance of the public transport. Some examples of available microservices are described below.

- **Action Recognition** service identifies the type of action that a person is performing, by analyzing the sequence of human poses obtained from the track of each person.

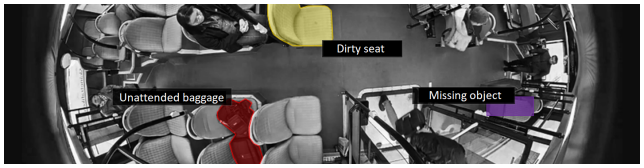

Figure 4: Examples of auxiliary functionalities: supporting cleaning activities, reporting abandoned objects, identifying damage and vandalism, detecting fire or smoke.

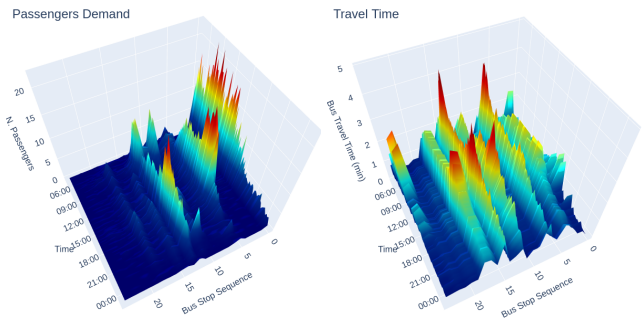

Figure 5: Example of passenger flow and travel times as a function of daily time and for each stop of the MBTA Bus Route 1.

The proposed solution is a skeleton-based approach for action recognition via CNN. The developed CNN relies on a heatmap as the base representation of human skeletons, such the one proposed in (Duan et al. 2021). The model is trained on a custom dataset of actions, derived from the well-known NTU RGB Dataset (Shahroudy et al. 2016), which contains 60 different action classes including daily, mutual, and health-related actions.

- **Fight Detection** service recognizes violence between individuals and crowds, by analyzing the sequence of human poses taken from the track of all people in the view of a single camera. The proposed approach is based on the Action Recognition service which can handle multiple-person scenarios without any additional costs. The model is trained on the AIRTLab dataset (Bianculli et al. 2020), containing video clips for both violent and non-violent scenarios.
- **Urban Public Transport Optimization** service determines public transport optimization strategies based on the data the SPaCe system has acquired. The information used is passenger numbers, travel times, and passenger profiles. The analytics modules provide online and real-time performance metrics and recommendations for public service improvement. In the rest of the paper, the proposed solutions for public transport optimization are reported in depth.

Urban Public Transport Optimization

Public transport is a very complex system, and making routes and scheduling strategies more efficient certainly has a positive impact on its operation. The larger objective is to strike a balance between several competing interests. From the passenger point of view, everybody wants to get to their destination as soon as possible. Moreover, there is an increasing demand for a tailored, capillary service that maximizes comfort and simplicity of use. On the other hand, urban transport agencies aim to guarantee that as many people as possible have access to services, while attempting to reduce operating costs and increase profit. Then, the public transport optimization problem is a Multi-objective Optimization Problem (MOP) (Marler and Arora 2004). The MOP approach seeks to optimize two or more competing goals simultaneously. Since the objectives are incompatible, there is no single solution to the problem: no single objective can be fully optimized without negatively affecting others. This leads to a set of solutions known as the Pareto front, where each solution can lean towards one goal or the other. The service management team will finally decide which of the optimized solutions is most suitable to implement in light of the objectives to be achieved.

With the use of the data collected by SPaCe many different optimization problems can be solved, for example: fair resource allocation, efficient use of resources, and meeting passenger needs. Using SPaCe for the public bus transport system, an urban transport agency can solve the above problems with the following services:

- **Bus Timetable Optimization** service proposes a strategy to make the trip scheduling of routes more effective acting on the current state of bus transport demand (Tang et al. 2020);
- **Route Network Optimization** service proposes a multi-level and multi-mode optimization model for urban bus transit route network design (Wang, Ye, and Wang 2020);
- **Customized Bus Route Optimization** service proposes a solution for on-demand reprogramming of urban vehicle routes in accordance with passenger demand, vehicle status, and traffic distribution (Guo et al. 2022).

For the sake of brevity, only the bus timetable optimization is detailed in this paper.

Bus Timetable Optimization

A timetable provides the departure times of each trip from the terminal and the expected arrival times at each stop on a route. The inefficient frequency of trip departure times results in fully loaded vehicles during peak hours and nearly empty vehicles during off-peak hours. The goal is to satisfy passenger demand while also taking business operational costs into account. Usually, public transport agencies reduce the gap between the departures of two trips on the same route during peak hours (i.e., when there is a high volume of passengers) in order to shorten waiting times. As a result, the number of buses operating simultaneously will increase, and this will raise operational costs. Furthermore, a surplus of buses during off-peak hours would result in low usage rates because of overcapacity.

The schedule of each bus route is tightly related to multiple changing variables, including passenger volume, average travel time between stops, and traffic conditions. Due to its complexity, scheduling is a task that only highly skilled professionals can perform at present. However, this process can be automated to provide highly efficient scheduling much more easily using real-time data provided by the SPaCe on–board analytics system.

Solution Design

The proposed solution is a data-driven optimization of a single bus route based on a multi-objective genetic algorithm (Tang et al. 2020). The main features of the algorithm are problem modeling, decision variables, objective functions, and constraints. They are detailed below:

1. The key inputs to the model are the actual travel times between nearby stops and the number of people getting on and off at each stop on a route. These data are continuously collected by SPaCe as they are strictly time-dependent. The data are automatically analyzed to understand how travel times and passenger volume change throughout the day, as shown in Figure 5.
2. The decision variables are the daily departure times of each trip for a single route.
3. Two objective functions are used: the number of trips for a given route ($F1$), which models the transport agency's interests, and the passenger waiting time ($F2$), which models the user's interests. Waiting time is defined as the time interval between the passenger's arrival at the bus stop and boarding the bus.
4. The constraints are the operating period and vehicle carrying capacity. For a given route, the first and last trips have fixed departure times, and the minimum and maximum number of trips per day are fixed by the parameters I_{min} and I_{max}. Further it is assumed that all buses have the same carrying capacity set to C.

The genetic algorithm NSGA-II (Deb et al. 2002) is used to solve the optimization problem because it is highly suitable for large and complex problems. Since the chosen decision variable is the timetable, a special coding scheme is used to map timetables onto the chromosomes of the algorithm population:

- The chromosome is a vector having length $I_{max}-2$, since the first and last trips are fixed.
- Each gene (the vector's element) is an integer that represents how many minutes elapse between the departure time of one trip and the next.
- The value of a gene is included in the interval $[0, H_{max}]$, where H_{max} is the maximum time interval between two consecutive trips.
- Genes with a zero value are discarded, in this way it is possible to have solutions with a number of trips lower than I_{max}.

The initial genetic population is calculated using the Integer Random Sampling (Blank and Deb 2020). The number of the population's chromosomes is set to S_p and each gene's value is randomly extracted in the range $[0, H_{max}]$. Furthermore, the genetic evolution uses the operators: Integer Simulated Binary Crossover and Integer Mutation (Deb, Sindhya, and Okabe 2007).

The objective functions computing process simulates the execution of trips along the route; from the input data, it estimates the number of passengers getting on and off at each stop based on the vehicle arrival time at the stop, which depends on the trip departure time and travel times. The proposed algorithm computes near-optimal Pareto solutions in a short time, i.e., the set of optimized timetables.

Experiments

The open data of the Massachusetts Bay Transportation Authority (MBTA 2020) is used to evaluate the bus timetable optimization system. The chosen dataset provides extensive information about public transport in the Greater Boston area, as if the SPaCe system were installed in each vehicle. The MBTA data includes three types of information: *GTFS* provides trip schedules and geographic information of each route; *Bus Rideship* provides the mean number of boarding and alighting passengers at each stop for each trip; and *Bus Arrival Departure Times* data. The MBTA bus dataset is the only one that includes this relevant data, other datasets only report static information. The lack of accurate and complete data on passengers and vehicles makes it impossible to analyze transport system status and take actions to improve it. This fact underlines the necessity of an ITS, such as SPaCe, deployed on–board of each public transport vehicle.

MBTA Scheduling					
Route	N. stops	N. trips	AM rush hour	Day	PM rush hour
65	30	36	8'	70'	12'

Table 1: Example of data used in an optimization process: the id of the route, the number of stops, the number of trips, the trip departure frequencies in minutes during the day and respect to the rush hour AM versus PM.

Several bus routes from the MBTA transit system, with different passenger flow and bus frequency characteristics, are selected, in order to evaluate the behavior of the algorithm in different scenarios. For the sake of brevity, the analysis of only one route is reported: *Route 65 Kenmore Station - Brighton Center*. According to MBTA timetables, Route 65 has a low frequency of trips during off-peak hours, but it increases significantly during peak hours, as shown in Table 1.

Several parameters have to be selected to complete the model definition. The ones related to the features of the route are: the minimum and maximum number of trips are set to $I_{min} = 27$ and $I_{max} = 45$, which are $\pm 25\%$ of the number of trips scheduled by MBTA; the maximum time interval between two consecutive trips is set to $H_{max} = 80$ based on the features of the route, as shown in Table 1; and the bus carrying capacity is set to $C = 40$ people.

The parameters pertaining to the optimization algorithm NSGA-II are: the number of generations GEN, the population size POP, the crossover rate P_c and the mutation rate P_m. The first one is set to $GEN = 400$, following (Tang et al. 2020), while a grid search is performed to determine the other values. The reduction in passenger waiting time of the optimized timetable compared to the MBTA one is the grid search metric. The optimized timetable is the solution of the Pareto Front which has the number of trips equal to that scheduled by MBTA. Based on the results of the grid search, they are set to $POP = 200$, $P_c = 0.7$, and $P_m = 0.05$.

The algorithm generates a set of possible solutions with different weights between the two objective functions. Thanks to an interactive dashboard, shown in Figure 6, the results of the optimization can be explored. The dashboard

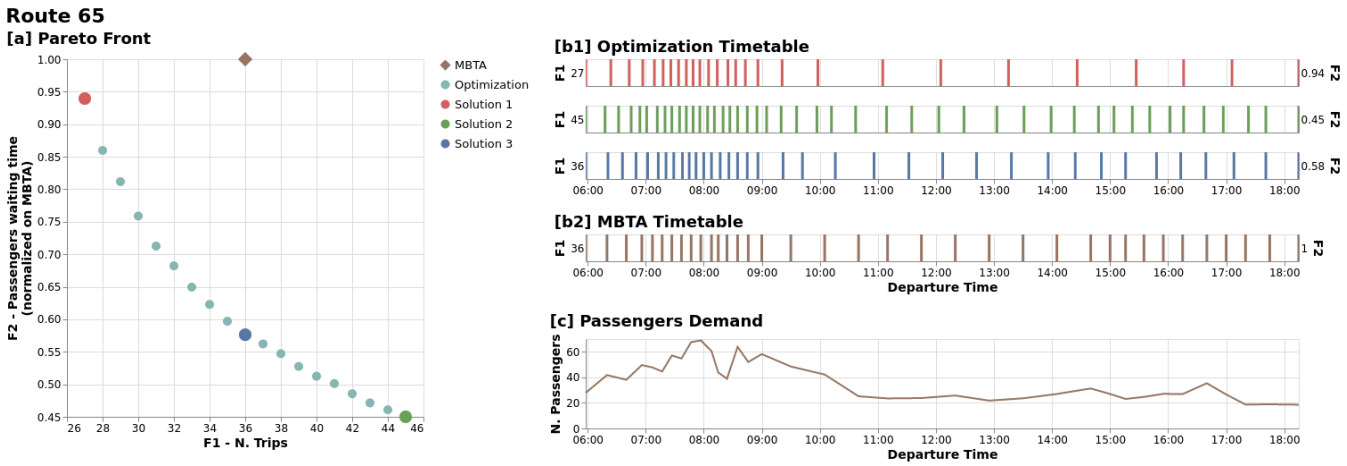

Figure 6: **Route 65**
[a] *Pareto Front*: the set of problem solutions in the plane of objective functions. Up to three solutions can be selected interactively in different colors. The diamond-shaped dot represents the current MBTA timetable.
[b1] *Optimization Timetable*: the related timetables are displayed for each of the 3 selected solutions. The vertical line represents trip departure times. The values of $F1$ and $F2$ for each timetable are displayed on the y axis, on the left and right, respectively.
[b2] *MBTA Timetable*: trip departure times of timetable currently used by MBTA.
[c] *Passenger Demand*: the total number of passengers boarding along the route. It is possible to zoom interactively on a time window in order to better visualize the timetables *[b]*, where the time axis adjusts accordingly.

aggregates data about the route and the optimization process in a single view and enables switching between the Pareto solutions. In the case of *Route 65*, the optimization produced timetables that are much more efficient than the MBTA schedule for the same route. As reported in Figure 6.a, for the same number of trips (i.e. 36 trips), the optimized solution results in a 42% reduction in overall passenger waiting time.

The public transport agency could achieve this service improvement simply by redistributing the departure times of the trips in real–time, without any additional cost; in fact, the number of buses is the same. Alternatively, if the existing waiting time is deemed appropriate, the trips could also be reduced by ten units and reprogrammed according to the optimized solution; in this way, the service costs would decrease but the waiting time would remain the same.

These solutions, which are a few of the many that the public transport agency has available, illustrate the significant benefits that come with using a real–time data–driven platform to recommend how to schedule the trips.

The proposed algorithm has produced timetables for all the examined routes that are more efficient than the current MBTA ones: less passenger waiting time for equal numbers of trips. Finally, because the Pareto solutions cover multiple scheduling options and illustrate what occurs when one goal is prioritized over another, they can offer valuable objective support to the decision-making process in order to balance the interests of the transport agencies and passengers.

Conclusions

This paper introduces the MerMec solution for real–time public transport optimization. The SPaCe system allows to monitor people, objects, and vehicles in order to optimize the management of public transport vehicles. Furthermore, it has been shown how an automatic optimization of timetable scheduling can be extremely appealing to public transport agencies: it provides an easy-to-integrate optimization service by acting on the current state of transport without requiring any human intervention. Finally, state–of–the–art analytical systems for public transport are currently under development, such as route network optimization and customized bus route optimization.

References

Alstom. Mastria: Transforming city-wide multimodal transport. https://www.alstom.com/our-solutions/digital-mobility/mastria-transforming-city-wide-multimodal-transport.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, 3464–3468.

Bianculli, M.; Falcionelli, N.; Sernani, P.; Tomassini, S.; Contardo, P.; Lombardi, M.; and Dragoni, A. F. 2020. A dataset for automatic violence detection in videos. *Data in Brief* 33:106587.

Blank, J., and Deb, K. 2020. pymoo: Multi-objective optimization in python. *IEEE Access* 8:89497–89509.

Cao, F.; Estert, M.; Qian, W.; and Zhou, A. 2006. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, 328–339. SIAM.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A

fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6(2):182–197.

Deb, K.; Sindhya, K.; and Okabe, T. 2007. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, 1187–1194.

Duan, H.; Zhao, Y.; Chen, K.; Shao, D.; Lin, D.; and Dai, B. 2021. Revisiting skeleton-based action recognition.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 226–231.

European Union. 2010. Directive 2010/40/eu of the european parliament and of the council of 7 july 2010 on the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport. *Off. J. Eur. Union* 53.

Google. 2021. Movenet: Ultra fast and accurate pose detection model. https://tfhub.dev/google/movenet/singlepose/lightning/4.

Google. 2022. Gtfs static overview. https://developers.google.com/transit/gtfs/reference.

Guo, R.; Guan, W.; Bhatnagar, S.; and Vallati, M. 2022. A two-phase optimization model for autonomous electric customized bus service design. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 383–388. IEEE.

He, L.; Liao, X.; Liu, W.; Liu, X.; Cheng, P.; and Mei, T. 2020. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*.

King, D. E. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10:1755–1758.

Lopez, P. A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.-P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; and Wießner, E. 2018. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, 2575–2582. IEEE.

Luo, H.; Gu, Y.; Liao, X.; Lai, S.; and Jiang, W. 2019. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 0–0.

Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; and Woodall, W. 2022. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics* 7(66):eabm6074.

Mangiaracina, R.; Perego, A.; Salvadori, G.; and Tumino, A. 2017. A comprehensive view of intelligent transport systems for urban smart mobility. *International Journal of Logistics Research and Applications* 20(1):39–52.

Marler, R. T., and Arora, J. S. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26:369–395.

MBTA. 2020. Mbta open data portal. https://mbta-massdot.opendata.arcgis.com/.

Needs, I. 2012. Strategic transport infrastructure needs to 2030.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, J.; Yang, Y.; Hao, W.; Liu, F.; and Wang, Y. 2020. A data-driven timetable optimization of urban bus line based on multi-objective genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems* 22(4):2417–2429.

Thalesgroup. Naia: passenger journey analytics. https://www.thalesgroup.com/en/markets/transport/naia-passenger-journey-analytics.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2022. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Wang, C.; Ye, Z.; and Wang, W. 2020. A multi-objective optimization and hybrid heuristic approach for urban bus route network design. *IEEE Access* 8:12154–12167.

Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; and Hoi, S. C. 2021. Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence* 44(6):2872–2893.

Zhang, J.; Wang, F.-Y.; Wang, K.; Lin, W.-H.; Xu, X.; and Chen, C. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 12(4):1624–1639.