SchoolRide: A Platform for School Bus Disruption Management and Operational Resilience

Vakul Nath, Fangqi Liu, Guocheng He, David Rogers, Ajay D. Chhokra,
Jose Paolo Talusan, Meiyi Ma, Ayan Mukhopadhyay, and Abhishek Dubey
Vanderbilt University
Nashville, TN, USA

***Abstract*—As a societal-scale transportation Cyber-Physical System (CPS), school transportation integrates large-scale physical operations with cyber components for planning and control under uncertainty. Despite its scale and societal importance, the system remains vulnerable to operational disruptions such as vehicle breakdowns, road closures, traffic congestion, and driver absences. This work demonstrates how data-driven optimization can enhance operational resilience in a real-world school transit context. To advance research in this domain, we introduce SchoolRide, a platform developed in close collaboration with a school district in the southern United States. SchoolRide serves as a comprehensive testbed for studying and evaluating robust operational policies for disruption management, enabling systematic investigation of strategies under realistic data and operational constraints. We design an integrated pipeline for dynamic bus status collection and formulate the School Bus Disruption Management (SBDM) problem as a combinatorial optimization task that replans routes based on predefined schedules, real-time status, and disruption events. The framework balances student service quality (e.g., waiting time and school delays) with operational efficiency (e.g., route adjustments and driver workload). We explore heuristic and optimization-based approaches that leverage historical disruption logs from the partner district to proactively replan routes and evaluate their performance using synthetic data generated from real-world operational records to protect privacy. Our approach outperforms current operational policies, effectively preserving service quality while reducing disruptions and workload. This work contributes to improving transportation resilience and advancing data-driven decision support for school transportation systems.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The U.S. public school bus network is the nation's largest cyber-physical transportation system, operating over 480,000 buses to transport 26 million students daily [1]. As a societal-scale CPS, it couples physical operations (vehicles, drivers, and road networks) with cyber components for planning, monitoring, and control. Despite its scale and importance, the system remains vulnerable to disruptions such as vehicle breakdowns, traffic congestion, and driver shortages. Surveys indicate 91% of districts face driver shortages and 60% have shortened or canceled routes [2]. In one partner district, daily absences of up to 11 drivers from a roster of 209 require dispatchers to reassign overloaded buses in real time [3]. Similar challenges arise in major cities, where thousands of delay incidents occur monthly [4], and in coastal regions where flooding forces route detours or virtual instruction [5].

Despite such operational fragility, real-time disruption management has received limited attention in intelligent transportation research. Most districts rely on manual coordination, with minimal decision support for rerouting or compliance with policy constraints (e.g., prohibitions on roadside parking or mixing students from different schools). Managing disruptions is thus a complex optimization task balancing student service quality (waiting time, delays, and schedule adherence) with operational constraints such as route changes, driver workload, and limited spare capacity. Unlike classical Vehicle or School Bus Routing Problems (VRP/SBRP), disruption management requires continuous vehicle monitoring and real-time route updates respecting institutional regulations.

This paper presents **SchoolRide**, an open, modular, decision support system built with community partners for real-time school-bus disruption management. This system integrates live telemetry, disruption events, and scheduling data into a unified operational view of the district. When a disruption occurs, it seeks to recover service rapidly while preserving quality of life for operators, students and families. Practitioners can use the system to monitor the state of the fleet, respond to disruptions such as driver absences, vehicle breakdowns, road closures, and severe congestion. It also supports experimentation with alternative disruption-response policies and comparison of their performance across realistic synthetic or historical scenarios before adaption of policies in routine operations. **SchoolRide** continuously ingests disruption events and live telemetry, reconstructs the current operational state of the fleet (bus locations, loads, and remaining routes), solves a constrained rerouting problem to propose recovery actions, and updates planned routes accordingly. As buses execute these updated routes, new telemetry is observed and any further disruptions are processed in the same way, allowing the system to adapt as operating conditions change throughout the day.

Contributions. We review prior work in school transportation and disruption management (Section II) and extend the School Bus Routing Problem with a disruption-aware formulation that balances service quality and operational efficiency under transportation constraints (Sections IV and V). We develop **SchoolRide**, a pipeline that combines disruption modeling, travel-time prediction, and constrained optimization for timely rerouting (Section III). We also design a location-agnostic synthetic data generator and evaluate our approach on both synthetic benchmarks and real district data, showing strong performance and scalability (Section VI).

II. Related Work

SBRP research spans a broad set of subproblems, from stop selection and route generation to scheduling and bell-time adjustment, with complex constraints and problem settings [6]–[9]. Existing work emphasizes multi-school and mixed-load scenarios, advanced metaheuristics, and uncertainty in ridership and travel times, typically at the planning stage. However, most prior work focuses on offline optimization and does not address real-time disruption handling or operational decision support. This gap highlights the need for solutions that integrate algorithmic routing with adaptive operational workflows.

A wide range of optimization models has been developed for the SBRP. Exact formulations [10] and decomposition-based approaches [11] provide strong guarantees on structured instances, while column-generation methods [12] and metaheuristics such as GRASP+VND and genetic algorithms [13], [14] scale more effectively to large planning problems. In parallel, learning-based routing methods have been developed for VRP variants: constructive neural models generate routes end-to-end [15], [16], iterative-improvement models refine tours using enhanced positional encodings [17], and hybrid approaches integrate learned components into search procedures [18], [19]. Although these techniques perform strongly on VRP settings, they are trained on simplified routing distributions and do not incorporate the policy, timing, eligibility, and multi-school compatibility constraints that define the SBRP, leaving them oriented toward offline optimization rather than the real-time, regulation-driven demands of school bus operations.

Beyond classical SBRP models, disruption management has been studied in broader vehicle-routing contexts. Prior work examines recovery strategies for breakdowns or execution-stage disturbances in freight systems [20]–[23], typically focusing on restoring feasibility and minimizing deviation from the original plan under flexible service windows and fewer policy constraints. School bus operations, by contrast, are governed by tightly coupled route structures, strict bell-time schedules, maximum ride-time limits, student–stop eligibility, and compatibility rules that sharply restrict feasible adjustments. As a result, disruption-management techniques from general VRP settings cannot be directly applied without explicitly modeling these educational, safety, and equity constraints at scale.

III. System Architecture

Our proposed system, **SchoolRide**, is a CPS-based platform that senses real-time bus status, detects disruptions, and automatically computes updated routing plans. The recommended rerouting solutions are delivered to dispatchers through a user interface. We consider two types of disruption events: **external disruptions**, such as road closures or severe congestion, where the system recomputes travel times between stops using online OSRM (Open Street Routing Machine) queries, producing an updated travel-time matrix that reflects the changed network; and **intrinsic disruptions**, such as bus breakdowns or driver absences, where the affected vehicle is removed from service and both its onboard students and its unvisited pickup stops must be reassigned to other working buses. In both cases, the system triggers a **rerouting step** that replans service for all impacted buses and remaining stops, using the formulation and decision approaches described in Sections IV and V. To support this CPS-based functionality, **SchoolRide** is implemented as a collection of modular services connected through well-defined interfaces. The following introduces the three primary service categories.

1) Data Services. This component senses and aggregates real-time information on bus locations, passenger counts, operational status, and disruption events, and stores the processed streams in a PostgreSQL database. The current implementation ingests incident notifications from the county's Computer-Aided Dispatch (CAD) system, real-time vehicle telemetry from GPS providers such as Samsara, and samples social media (e.g., county's Traffic Operation Center's X) feeds to identify potential disruptions. The resulting data streams provide the CPS sensing layer that drives the planning and rerouting logic.

2) Disruption Management Services. This component forms the system computational core, integrating baseline routing, travel-time prediction, and disruption-aware replanning. The routing module computes shortest paths between stops using graph-based algorithms over OpenStreetMap (OSM) data [24]. OSRM provides updated travel times along each path when road conditions change, such as during closures or congestion, and these updated times are used to compute stop-level ETAs by combining live telemetry with route geometry. When a disruption is detected, the planning layer retrieves the current system state from the Data Services, assembles an updated travel-time matrix, and applies heuristic-based algorithms to produce feasible rerouting options. This problem description and solutions are presented in Sections IV and V.

3) User Interface Services. This component presents the CPS outputs to end users. A Plotly-based dashboard allows dispatchers to visualize live bus status, report disruption events, and review rerouting plans. A chatbot module, built on an LLM-based pipeline, supports natural language queries for system status and contextual information. A notification module delivers personalized updates to parents and guardians regarding route changes, expected arrivals, and active disruptions via push or SMS alerts. Together, these modules support real-time monitoring, communication, and human-in-the-loop decision making.

IV. School Bus Disruption Management

We formulate the School Bus Disruption Management (SBDM) problem, which is solved whenever a disruption occurs, using the planned routes, current vehicle status, and the disruption event to replan the remaining service and mitigate disruption impact.

System Model. We consider a school bus system with a set of vehicles $\mathcal{V}$, pickup stops $\mathcal{P}$, and school destinations $\mathcal{D}$. Each vehicle $k \in \mathcal{V}$ is assigned a predefined route $R^{\text{pre}}(k)$ representing its planned sequence of stops. Each pickup stop $p \in \mathcal{P}$ has a planned arrival time $T_p(p)$ and student load $l(p)$. Each school $d \in \mathcal{D}$ has a bell time $T_b(d)$. Each vehicle k departs at time $T_s(k)$ and has capacity limit $C^{max}(k)$. We use a binary mapping $\mathsf{P2D}(p, d)$, where $\mathsf{P2D}(p, d) = 1$ if pickup stop p is assigned to school d, and 0 otherwise. Each pickup stop is associated with exactly one school destination through P2D. Stops appearing on different routes are treated as distinct nodes even if they are at the same physical location, allowing route-specific timing and service context to differ.

Disruption Management. When a disruption occurs, we reconstruct the rerouting problem based on the current system state. External disruptions, such as road closures, induce an updated travel-time matrix $Trav'$, where $Trav'(i, j)$ denotes the revised travel time between stops i and j. Intrinsic disruptions, such as bus breakdowns, remove affected vehicles from service; the remaining operational vehicles are denoted by $\mathcal{V}^\star \subseteq \mathcal{V}$. Let $\mathcal{P}^{\text{rem}} \subseteq \mathcal{P}$ denote the set of pickup stops that have not yet been served at the disruption time. For a failed vehicle, both its unserved pickup stops in $\mathcal{P}^{\text{rem}}$ and any onboard students not yet dropped off must be reassigned to the active fleet. We represent such onboard students as additional pending drop-off demand at the disruption location, each associated with its assigned school destination.

For each active vehicle $k \in \mathcal{V}^\star$, let $loc(k)$ denote its current location (or next stop if en route), $ETA(k)$ its current estimated arrival time, and $R^{\text{pass}}(k)$ the prefix of already visited stops. Only the unserved portion of the route is reoptimized. The updated route for vehicle k is written as $R^\star(k) = R^{\text{pass}}(k) \,\|\, R^{\text{rem}\star}(k)$, where $R^{\text{pass}}(k)$ denotes the fixed prefix of already visited stops and $R^{\text{rem}\star}(k)$ denotes the remaining portion of the route to be optimized using $Trav'$ and the updated system state. We denote by r_i^k the i-th stop in the reoptimized suffix of route $R^\star(k)$ for $i \geq 1$, while $i = 0$ denotes the vehicle state at the disruption time.

The updated routes must satisfy the following constraints:

(C1) **Stop Assignment.** Each unserved pickup stop must be assigned to exactly one active vehicle:

$$\sum_{k \in \mathcal{V}^\star} \mathcal{I}[p \in R^\star(k)] = 1, \qquad \forall p \in \mathcal{P}^{\text{rem}} \tag{1}$$

where $\mathcal{I}[\cdot]$ is the indicator function, equal to 1 if the condition holds and 0 otherwise.

(C2) **Pickup–Drop-off Precedence.** Each pickup stop must be followed later in the route by its assigned school destination. For every pickup stop $r_j^k \in R^\star(k) \cap \mathcal{P}^{\text{rem}}$:

$$\sum_{i=j+1}^{|R^\star(k)|} \mathsf{P2D}(r_j^k, r_i^k) > 0, \qquad \forall k \in \mathcal{V}^\star \tag{2}$$

(C3) **Capacity Constraint.** Let $Q_d^k(i)$ denote the number of onboard students on vehicle k whose destination is school d, immediately after serving stop r_i^k on route $R^\star(k)$. It is updated as follows:

$$Q_d^k(i) = \begin{cases} Q_d^k(i-1) + l(r_i^k), & r_i^k \in \mathcal{P},\ \mathsf{P2D}(r_i^k, d) = 1 \\ 0, & r_i^k = d \\ Q_d^k(i-1), & \text{otherwise} \end{cases} \tag{3}$$

for all $d \in \mathcal{D}$ and $i \geq 1$, with initialization $Q_d^k(0) = Q_{d,0}^k$, where $Q_{d,0}^k$ is the onboard load already assigned to destination d at the disruption time. The total onboard load after serving stop r_i^k is

$$Q^k(i) = \sum_{d \in \mathcal{D}} Q_d^k(i) \leq C^{max}(k), \qquad \forall i,\ \forall k \in \mathcal{V}^\star \tag{4}$$

(C4) **Mixed-Load Restriction.** To prevent misrouting and comply with district policies prohibiting students with different school destinations from riding together, a bus may serve only one destination group at a time. Therefore, the onboard load must be zero immediately after each school drop-off:

$$Q^k(i) = 0, \qquad \forall r_i^k \in \mathcal{D},\ \forall k \in \mathcal{V}^\star \tag{5}$$

ensuring that all passengers are unloaded before pickups associated with another destination are served.

(C5) **Arrival-Time Update and Early-Arrival Constraint.** Let $T_a(r_i^k)$ denote the arrival time at stop r_i^k, with initialization $T_a(r_0^k) = ETA(k)$. Arrival times are updated recursively as

$$T_a(r_{i+1}^k) = \max\big(T_a(r_i^k), T_p(r_i^k)\big) + T_s(r_i^k) + Trav'(r_i^k, r_{i+1}^k), \tag{6}$$

for all $i < |R^\star(k)|$ and $k \in \mathcal{V}^\star$. We then require

$$T_a(r_i^k) \geq T_p(r_i^k) - \theta, \tag{7}$$

for all $r_i^k \in R^\star(k)$ and $k \in \mathcal{V}^\star$ to prevent excessive early arrivals, where θ is the maximum allowable early-arrival buffer and $T_s(r_i^k)$ denotes the service time at stop r_i^k.

Objective. We evaluate a rerouting plan using four metrics: pickup waiting time, school drop-off delay, route changes from the original plan, and riding-time violations.

$$T_{wait}(R^\star) = \sum_{k \in \mathcal{V}^\star} \sum_{r_i^k \in R^\star(k) \cap \mathcal{P}^{\text{rem}}} \max(0, T_a(r_i^k) - T_p(r_i^k))\, l(r_i^k) \tag{8}$$

$$T_{sch}(R^\star) = \sum_{k \in \mathcal{V}^\star} \sum_{r_i^k \in R^\star(k) \cap \mathcal{D}} \max(0, T_a(r_i^k) - T_b(r_i^k))\, Q^k(i-1) \tag{9}$$

$$C_{route}(R^\star, R^{\text{pre}}) = \sum_{k \in \mathcal{V}^\star} \Big(|E_k^\star \setminus E_k^{\text{pre}}| + |E_k^{\text{pre}} \setminus E_k^\star| \Big) \tag{10}$$

$$C_{\text{ride}}(R^\star) = \sum_{k \in \mathcal{V}^\star} \sum_{r_i^k \in \mathcal{P}_k^\star} l(r_i^k) \max\big(0, T_a(\phi(r_i^k)) - T_a(r_i^k) - \Gamma_{\text{ride}}\big) \tag{11}$$

where $\mathcal{P}_k^\star = R^\star(k) \cap \mathcal{P}^{\text{rem}}$ is the set of pickup stops served by vehicle k, $E_k^\star$ and E_k^{pre} are the sets of consecutive-stop segments in the updated and planned routes, respectively, $\phi(r_i^k)$ is the first downstream stop in $R^\star(k)$ satisfying

Algorithm 1: AdvIns: Insertion Heuristic

Input: routes, U_0, $\mathcal{E}_{\rm disr}$, T_p, $Trav$. **Output:** updated routes. $U \leftarrow U_0$.
foreach $(u, v) \in \mathcal{E}_{\rm disr}$ **do**
  Find k with (u, v) consecutive in $R^\star(k)$.
  seg', $spill \leftarrow$ ReorderSegment($R^\star(k), u$); replace that segment in $R^\star(k)$ by seg'; $U \leftarrow U \cup$ spill.
foreach $p \in U$ **do**
  Let d be its school with $\mathsf{P2D}(p, d) = 1$.
  $S_{\rm ins} \leftarrow \{p\}$ if some $R^\star(k)$ contains d; else $\{p, d\}$.
  For each $k \in \mathcal{V}^\star$ compute pos_k, $cost_k \leftarrow$ ComputeInsertionCost($R^\star(k), S_{\rm ins}$).
  Choose $k^\star = \arg\min_k cost_k$ and insert $S_{\rm ins}$ at $pos_{k^\star}$.
return $\{R^\star(k)\}_{k \in \mathcal{V}^\star}$.

Algorithm 2: Pruning Heuristic

Input: disrupted segments, working buses $\mathcal{V}^\star$, $Trav$, P2D, l, T_p, timing. **Output:** perVehicle and globalCandidates. requests $\leftarrow$ MakeRequestNodes(segments,P2D,l,T_p,timing);
busStates $\leftarrow$ BuildBusStates($\mathcal{V}^\star$,$Trav$,T_p);
perVehicle[k] $\leftarrow \emptyset$ for each $k \in \mathcal{V}^\star$; rvEdges $\leftarrow \emptyset$.
foreach *request r in requests* **do**
  candBuses $\leftarrow$ ClosestCandidatesForRequest(r, $\mathcal{V}^\star$, $Trav$, P2D, l, timing, maxCandidates);
  foreach *(bus,label) in candBuses* **do**
    state $\leftarrow$ busStates[bus.id]; eval $\leftarrow$ EvaluateRequestOnBus(r,state, $Trav$,timing,label,T_p);
    if *eval is defined* **then**
      perVehicle[bus.id].append((request_id, bus.id, eval.cost, eval.priority_rank, eval.metrics));
      rvEdges $\leftarrow$ rvEdges $\cup\{$(request_id, bus.id)$\}$.
Sort perVehicle[k] and keep top-N by (priority_rank, cost).
globalCandidates $\leftarrow$ top-M of all perVehicle[k].
return perVehicle, globalCandidates, rvEdges.

$\mathsf{P2D}(r_i^k, \phi(r_i^k)) = 1$, and $\Gamma_{\rm ride}$ is the maximum allowable student riding time.

The objective function of SBDM is

$$\mathcal{J}(R^\star, R^{\rm pre}) = \omega_{\rm t}\, T_{wait}(R^\star) + \omega_{\rm d}\, T_{sch}(R^\star) + \omega_{\rm c}\, C_{route}(R^\star, R^{\rm pre}) + \omega_{\rm r}\, C_{\rm ride}(R^\star) \quad (12)$$

The weights $\omega_{\rm t}, \omega_{\rm d}, \omega_{\rm c}$, and $\omega_{\rm r}$ are nonnegative coefficients reflecting relative importance of service quality and operational stability caliberared to partnered district's dispatcher preferences.

V. Approach

Dynamic Travel-Time and ETA. SchoolRide maintains a dynamic $Trav$ over the stop graph. Baseline entries are derived from OSRM as point-to-point travel times between stops, and disruptions such as road closures are represented by modifying the OSRM graph and setting the speed on selected edges to zero, thereby blocking those links and updating the corresponding entries in $Trav$. Given a remaining route and departure time, an ETA routine then propagates the bus along its stops using T and stop service times, enforcing capacity and timing constraints and returning predicted arrivals for the rerouting algorithms.

Disruption Response. When a disruption occurs, SchoolRide reconstructs the live routing state and forms a set of disrupted pickups U_0. These include (i) all unvisited pickups and undelivered students on any bus that has become unavailable, and (ii) pickups on routes whose remaining path would traverse road segments affected by external disruptions such as closures or severe congestion. **AdvIns** (Algorithm 1) operates on U_0 and the current routes of the working buses $\mathcal{V}^\star$. If a road closure is present, each impacted arc (u, v) is first handled with ReorderSegment, which greedily reorders the segment starting at u to avoid the blocked edge and adds any newly unserviceable pickups to U_0. AdvIns then reinserts each pickup in U_0 by evaluating feasible insertion positions across working buses under the constraints of Section IV, selecting the lowest-cost feasible placement until all disrupted pickups are served.

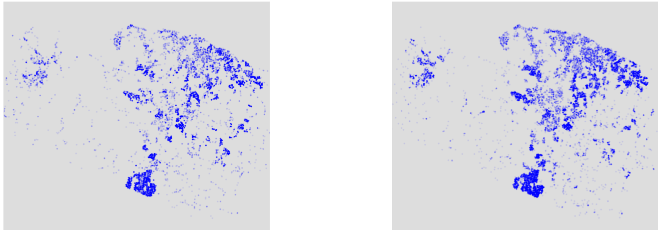
Input Pruning. Because disruption management is time-critical, **SchoolRide** further improves response time by introducing a pruning stage inspired by [25] that narrows the search space before calling **AdvIns**. Algorithm 2 groups disrupted pickups that share a school into a single request and identifies only a small set of promising request-bus pairs. For each request r with pickups P_r and school s_r, ClosestCandidatesForRequest selects nearby feasible buses using proximity, school overlap, and earliest finish time. EvaluateRequestOnBus then estimates the delay impact of inserting P_r before s_r, assigns a priority label, and ranks candidates. Each bus retains its top $N_{\rm veh}$ candidates, and the system keeps the global top $N_{\rm glob}$ across all buses. These pruned request-bus pairs are passed to **AdvIns**, reducing computation while preserving solution quality.

Metaheuristic Improvement. When time permits, **SchoolRide** refines **AdvIns** routes using **Hexaly** [26], a commercial optimizer combining mixed-integer programming with large-neighborhood search. The SBDM model (Section IV) is warm-started from the **AdvIns** solution and pruned request–bus candidates; **Hexaly** then explores neighboring solutions to further reduce the weighted objective J while enforcing all operational constraints. This optional stage improves solution quality when additional computation time is available without affecting real-time responsiveness.

VI. Experiments

A. Synthetic Data and Disruption

This subsection describes the synthetic data generation pipeline to create a high–fidelity surrogate of a mid–sized U.S.



(a) Real stops (b) Synthetic stops

Fig. 1: Real vs. synthetic bus-stop locations.

school transportation system (Figure 1). The process produces a linked set of artifacts: (i) sampled rider locations, (ii) bus stops, (iii) candidate routing scenarios, and (iv) validated final routes and summary statistics. The pipeline consumes public attendance zone polygons, residential building footprints, per-school enrollment counts as inputs. The workflow begins by constructing per-school residential inventories by intersecting attendance-zone polygons with building footprints and estimating each building's student potential. Dense housing clusters are then detected using DBSCAN, with buildings in these clusters up-weighted to reflect realistic residential concentrations. Each school's synthetic ridership is sampled from this weighted pool, after which OSRM is used to compute network-based distance matrices to guide stop generation. Bus stops are created via constrained agglomerative clustering, selecting medoids as stop locations and assigning riders based on walkable network distances. Using these stops, multiple routing scenarios are produced by sweeping heuristic parameters and then filtered under operational VRP constraints such as bus capacity, ride-time limits, schedule buffers, and campus-sharing rules. The synthetic dataset is validated against real metrics to ensure spatial and aggregate fidelity, and all resulting artifacts. The resulting synthetic dataset contains 26,434 riders, 6,985 stops, 439 routes, and 256 buses, compared with 23,066 riders, 6,200 stops, 357 routes, and 153 buses in the real district.

Disruption sampling. We simulate stochastic disruptions using probabilistic models driven by bus schedules and historical logs. For each instance, expected counts of bus breakdowns, driver absences, and road disruptions are computed as $\texttt{pro\_breakdowns} \cdot |\mathcal{V}|$, $\texttt{pro\_absence} \cdot |\mathcal{V}|$, and $\texttt{pro\_disruption} \cdot |\mathcal{P}|$, where in our experiments $\texttt{pro\_breakdowns} = 0.02$, $\texttt{pro\_absence} = 0.05$, and $\texttt{pro\_disruption} = 0.01$. The actual numbers of events are then sampled from a small integer window of width $\pm\texttt{var}$ (with $\texttt{var} = 1$) around these means. Breakdowns and absences are assigned to randomly chosen buses, with absences modeled as breakdowns at route start and other breakdowns placed uniformly within each bus's operating window. Traffic disruptions (road closures and incidents) are sampled from historical days by selecting impacted route segments and assigning each a random time within the corresponding bus's service window. Policies are evaluated over many random seeds and 1 and 3 idle buses, yielding stochastically generated scenarios that reflect realistic disruption patterns.

B. Results

We evaluate the system across a range of synthetic instances with varying stop counts and time budgets introduced originally by [6]. The RSRB and CSCB families comprise the benchmark instances used to evaluate our system. In the **RSRB** (*Random Schools, Random Bus-stops*) and **CSCB** (*Clustered Schools, Clustered Bus-stops*) instances, both m schools and n stops are scattered uniformly over a $20\,\text{mi} \times 20\,\text{mi}$ region, with $m \in \{6, 12, 25, 50, 100\}$, $\quad n \in \{250, 500, 1000, 2000\}$. Student time windows are drawn uniformly in $[07{:}00, 11{:}00]$ with lengths of 10–30 min, and the maximum riding time (MRT) is set to 5 400 s (75 min). All buses are homogeneous (capacity 66) and depart from a central depot.

Baselines. Since SBDM lacks standardized school-specific baselines, we compare **AdvIns** against two dispatcher-inspired human-intuition policies that reflect state-of-practice operations.

HI-DI: Human Intuition - Dispatch Idle A myopic rule that, after a breakdown or disruption, shifts the remaining workload to another bus. It first dispatches idle vehicles, if none are available, it selects the earliest-finishing bus and appends disrupted stops whenever capacity and timing constraints allow.

HI-CM: Human Intuition - Closest Merge - A locally adaptive rule that, after a disruption, reassigns affected stops to the closest compatible existing route. It prefers working buses already serving the same school, then idle or earliest-finishing buses, and greedily merges pickups into these routes subject to constraints, without system-wide re-optimization.

Across both synthetic benchmarks (Table I) and a large real-district derived from the synthetic data generation pipeline illustrated in Figure 1, our **AdvIns** insertion heuristic consistently outperforms the two human-intuition policies on student-centered metrics. In the synthetic benchmarks, **AdvIns** achieves substantially lower average stop and school delays than **HI-DI** and **HI-CM** for all stop scales and both disruption frequencies, while **Hexaly** provides only marginal additional improvement. For the large instance in Table II, demand and schedule patterns are derived from a partner district, while stops and routes are generated by our synthetic pipeline. On this calibrated real-district–scale scenario, **AdvIns** reduces arrival delay from 29.38 to 12.29 minutes with one idle bus and from 21.70 to 11.86 minutes with three idle buses; school drop-off delay falls from 36.68 to 16.95 minutes and from 27.70 to 16.75 minutes respectively, outperforming the best human-intuition policy **HI-CM** with comparable route changes. Using Algorithm 2 to target the top 20 buses among 256 buses and further improving with **Hexaly**, the combined pruning-plus-**AdvIns**-plus-**Hexaly** pipeline produces high-quality rerouting plans in under one minute. Our model returns a feasible solution if one exists by enforcing route-structure as hard constraints, while treating delays and riding-time violations as soft penalties when unavoidable, we outperform intuitive dispatcher-style rules and yield substantially lower delays with similar cost.

Table 1: Metrics using the Synthetic Data, grouped by Stop Count (lower is better). Best values in bold, second best underlined.

		250 Stops			500 Stops			1000 Stops			2000 Stops		
Policy	Idle	ArrD	DropD	RChg	ArrD	DropD	RChg	ArrD	DropD	RChg	ArrD	DropD	RChg
HI-DI	1	8.4±10.9	8.8±10.9	2.4±0.5	5.2±3.5	5.4±3.5	3.0±0.6	6.1±3.9	6.2±3.9	4.4±1.2	5.9±2.1	5.9±2.1	6.5±1.3
HI-DI	3	0.7±0.4	1.0±0.4	2.7±0.7	0.5±0.9	0.6±0.9	2.9±0.8	1.6±1.4	1.7±1.4	4.5±1.1	2.3±1.5	2.4±1.5	6.3±1.4
HI-CM	1	6.5±9.2	7.0±9.2	**2.2±0.6**	4.5±2.9	4.6±2.9	**2.5±0.7**	5.7±3.9	5.8±3.9	**3.8±1.3**	5.1±1.9	5.1±1.9	**5.5±1.4**
Hi-CM	3	0.6±0.4	0.8±0.5	**2.3±0.7**	0.3±0.2	0.4±0.2	**2.6±0.8**	1.4±1.4	1.4±1.4	**3.9±1.3**	1.8±1.5	1.8±1.5	**5.4±1.5**
AdvIns	1	2.0±1.4	2.2±1.6	4.8±1.9	0.8±0.6	0.9±0.7	6.3±2.9	1.2±0.7	1.2±0.7	11.4±4.0	0.9±0.4	0.9±0.4	17.7±4.4
AdvIns	3	1.2±0.9	1.2±0.9	5.7±1.7	0.7±0.4	0.7±0.5	7.8±2.9	0.9±0.6	0.9±0.6	11.6±3.4	0.8±0.4	0.8±0.4	17.8±4.3
AdvIns+Hexaly	1	**2.0±1.4**	**2.2±1.4**	5.1±1.7	**0.7±0.5**	**0.9±0.7**	6.4±3.0	**1.1±0.7**	**1.2±0.7**	11.4±4.0	**0.8±0.4**	**0.8±0.4**	17.7±4.4
AdvIns+Hexaly	3	**0.6±0.3**	**0.8±0.3**	2.5±0.7	**0.3±0.2**	**0.4±0.3**	2.7±0.8	**0.9±0.6**	**0.9±0.6**	11.6±3.4	**0.7±0.4**	**0.7±0.4**	17.8±4.3

Table 2: Actual Data Metrics, grouped by Stop Count. Lower is better, best values in bold, second best underlined.

Policy	Idle	ArrD	DropD	RChg
HI-DI	1	48.42±8.70	58.98±10.06	9.20±0.45
HI-DI	3	40.35±8.26	49.99±9.59	9.20±0.45
HI-CM	1	29.38±10.52	36.68±12.61	**6.00±1.58**
HI-CM	3	21.70±9.86	27.70±12.15	**5.60±1.67**
AdvIns	1	12.29±6.34	16.95±8.37	6.60±2.97
AdvIns	3	11.86±3.45	16.75±4.23	7.20±2.05
AdvIns+Hexaly	1	**12.37±6.29**	**16.78±8.20**	6.40±2.88
AdvIns+Hexaly	3	**10.81±2.74**	**15.36±3.41**	6.80±1.92

VII. Conclusion

SchoolRide is a practical and extensible decision-support tool that improves both the resilience and the reliability of school transportation systems. We introduced the **AdvIns** insertion heuristic together with a pruning approach that enables efficient rerouting decisions in real time. We developed a synthetic data generation pipeline that produces realistic student rider distributions, stop locations, and routes calibrated to a real district; unlike standard SBRP benchmarks, which are based on small Euclidean instances with simplified distance metrics, our datasets explicitly respect the underlying road network and operating scale. Experiments on both these synthetic benchmarks and a real-district-scale instance show that **AdvIns** consistently outperforms human-intuition policies, achieving substantially lower student delays while maintaining a comparable level of operational disruption. The pruning layer enables these gains under tight time limits by restricting attention to a small set of promising bus–request pairs, and the **Hexaly** refinement layer leverages the same model to provide additional improvements when more computation is available.

Acknowledgment

We thank Williamson School District, Franklin Special School District, The Transportation Network Growth Opportunity (TNGO) initiative, under Tennessee Department of Economic & Community Development and National Science Foundation (Grants 2238815, 1952011 and 2443803).

References

[1] New York School Bus Contractors Association, "School bus fast facts," https://www.nysbca.com/fastfacts, 2023.

[2] HopSkipDrive, "State of school transportation report," https://www.hopskipdrive.com/blog/key-findings-2024-state-of-school-transportation-report, 2024.

[3] Partner District Transportation Department, "Internal transportation operations report," 2025, confidential document referenced with permission.

[4] New York City Council Data Team, "School bus delays," https://council.nyc.gov/data/school-bus-delays/, 2020.

[5] K. Mishkin, "At a crossroads: Flooding around hampton roads is making it harder to drive kids to and from school," 2021.

[6] J. Park and B.-I. Kim, "The school bus routing problem: A review," *European Journal of Operational Research*, 2010.

[7] W. A. Ellegood, S. Solomon, J. North, and J. F. Campbell, "School bus routing problem: Contemporary trends and research directions," 2020.

[8] K. Wang, Y. Yang, M. Liao, and A. Chen, "The school bus routing problem: A comprehensive review," *European Transport Studies*, 2025.

[9] J. Díaz-Ramírez, A. M. Rosas, B. García-Colis, and P. S. Valencia-Torres, "The school bus routing problem: A systematic literature review," 2024.

[10] P. Schittekat, M. Sevaux, and K. Sorensen, "A mathematical formulation for a school bus routing problem." IEEE, 2006.

[11] D. Bertsimas, A. Delarue, and S. Martin, "Optimizing schools' start time and bus routes," *Proceedings of the NAS*, 2019.

[12] J. Kinable, F. C. R. Spieksma, and G. Vanden Berghe, "School bus routing—a column generation approach," *ITOR*, 2014.

[13] P. Schittekat, J. Kinable, K. Sörensen, M. Sevaux, F. Spieksma, and J. Springael, "A metaheuristic for the school bus routing problem with bus stop selection," *European Journal of Operational Research*, 2013.

[14] M. Kang, S.-K. Kim, J. T. Felan, H. R. Choi, and M. Cho, "Development of a genetic algorithm for the school bus routing problem," *IJSEIA*, 2015.

[15] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv:1611.09940*, 2017.

[16] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *arXiv preprint arXiv:1803.08475*, 2019.

[17] Y. Ma, J. Li, Z. Cao, W. Song, L. Zhang, Z. Chen, and J. Tang, "Learning to iteratively solve routing problems with dual-aspect collaborative transformer."

[18] A. Hottung and K. Tierney, "Neural large neighborhood search for the capacitated vehicle routing problem," 2020.

[19] J. Fitzpatrick, D. Ajwani, and P. Carroll, "A scalable learning approach for the capacitated vehicle routing problem," *Computers & Operations Research*, 2024.

[20] X. Wang, X. Wu, and X. Hu, "A study of urgency vehicle routing disruption management problem," 2010.

[21] X. Zhang and L. Tang, "Disruption management for the vehicle routing problem with time windows," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, 2007.

[22] Q. Mu, Z. Fu, J. Lysgaard, and R. Eglese, "Disruption management of the vehicle routing problem with vehicle breakdown," *Journal of the Operational Research Society*, 2011.

[23] R. Eglese and S. Zambirinis, "Disruption management in vehicle routing and scheduling for road freight transport: a review," *TOP*, 2018.

[24] OpenStreetMap contributors, "Openstreetmap," https://www.openstreetmap.org.

[25] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," 2017.

[26] Hexaly, "Hexaly solver," https://www.hexaly.com.