1 Introduction

The persistent over-reliance on single-occupancy vehicles exacerbates urban transportation challenges, particularly along shared routes where multiple individuals travel similar paths. This inefficient mode of transportation not only worsens traffic congestion but also significantly increases emissions and energy consumption. While autonomous vehicles (AVs) present transformative opportunities for urban mobility, current research often overlooks the potential for coordinated operation between different vehicle types.

This paper addresses this gap by introducing a novel concept of *active* and *passive* autonomous vehicles, where smaller passive vehicles (PVs) can temporarily attach to larger active vehicles (AVs) during shared highway segments. This approach leverages the aerodynamic benefits of platooning while maintaining individual mobility flexibility. The core innovation lies in dynamic grouping and ungrouping algorithms that optimize system-wide energy efficiency while accommodating individual travel requirements.

Our main contributions are:

- A simplified problem formulation with deterministic assumptions that serves as an algorithmic baseline and maps cleanly to classic optimization problems
- A comprehensive formulation addressing real-world uncertainties including traffic dynamics and temporal constraints
- Efficient algorithms for both centralized and opportunistic vehicle matching and platoon management

2 Differences with Existing Platooning System

Traditional Vehicle Platooning: Coordinated Following

Existing vehicle platooning primarily refers to a cooperative driving strategy where multiple vehicles travel closely together in a convoy, maintaining small, constant inter-vehicle distances using automated driving technology and vehicle-to-vehicle (V2V) communication. In this paradigm, a lead vehicle dictates the platoon's speed and trajectory, while follower vehicles autonomously control their acceleration and braking to maintain a safe gap. The primary benefits are derived from aerodynamic drafting, where following vehicles experience reduced air resistance, leading to fuel savings typically in the range of 10-20% for trucks. The vehicles remain physically independent, and the "formation" is virtual, maintained through sophisticated control algorithms. Joining or leaving the platoon is a fluid process involving adjusting speed to merge into or exit from the convoy without any physical connection or disconnection.

Our Differentiated Approach: Active-Passive Physical Integration

Our proposed system fundamentally reimagines platooning by introducing a heterogeneous architecture of Active Vehicles (AVs) and Passive Vehicles (PVs), coupled with physical attachment and detachment. Unlike traditional platooning, where all vehicles are peers and consume their own power, our model allows smaller PVs to be physically towed by larger, specialized AVs during shared highway segments. This is not merely coordinated driving but a temporary physical merger where a PV's propulsion is deactivated, leading to energy savings that approach 100% for the towed vehicle during the attached phase, far surpassing the marginal gains from aerodynamics. This shifts the paradigm from "collaborative driving" to an on-demand, energy-transfer-based mobility service, introducing novel challenges and optimization problems in dynamic coupling/decoupling, capacity-aware fleet matching, and the management of physical rendezvous maneuvers, which are absent in the conventional platooning literature.

3 Simplified Problem Formulation and Solution

3.1 Problem Definition and System Model

We consider a coordinated platoon formation system operating on a unidirectional highway segment. The system comprises two distinct vehicle classes: *Active Vehicles (AVs)* capable of towing multiple smaller *Passive Vehicles (PVs).* The primary objective is to maximize total energy savings through strategic platoon formation while respecting vehicle capacity constraints and route compatibility.

3.1.1 Mathematical Formulation. The platoon formation problem is formulated as an optimization problem that matches Passive Vehicles (PVs) to Active Vehicles (AVs) to maximize total energy savings while respecting operational constraints.

The highway is represented as a sequence of segments between entry and exit points. Each vehicle has a predefined route from its entry point to its exit point.

The key components of the formulation are:

- **Decision Variables:** Binary variables x_{ij} that indicate whether PV j is assigned to AV i
- **Energy Savings:** Each assignment (i, j) yields energy saving S_{ij} proportional to their shared travel distance
- **Capacity Constraints:** Each AV i can tow at most C_i PVs simultaneously
- **Assignment Constraints:** Each PV can be assigned to at most one AV
- **Feasibility Constraints:** Platooning is only considered if vehicles share sufficient travel distance

The optimization problem is formally stated as:

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \cdot S_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j=1}^{M} x_{ij} \leq C_i \quad \forall i \in A \tag{2}$$

$$\sum_{i=1}^{N} x_{ij} \leq 1 \quad \forall j \in P \tag{3}$$

$$x_{ij} \cdot d_{ij} \geq x_{ij} \cdot L_{min} \quad \forall i, j \tag{4}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \tag{5}$$

where:

- N: Number of Active Vehicles
- M: Number of Passive Vehicles
- S_{ij}: Energy saving when PV j is towed by AV i
- C_i: Towing capacity of AV i
- d_{ij}: Shared travel distance between AV i and PV j
- L_{min}: Minimum shared distance required for platooning

This formulation captures the essential trade-off between maximizing energy savings through platooning while respecting the physical constraints of the system. The problem belongs to the class of constrained assignment problems with similarities to multiple knapsack problems.

3.2 Proposed Algorithms

3.2.1 Greedy Maximum-Weight Matching Algorithm. Our greedy approach leverages the observation that near-optimal solutions can be obtained by prioritizing assignments that offer the highest immediate energy savings. The algorithm operates in three distinct phases: candidate generation, priority ordering, and sequential assignment.

Phase 1: Candidate Generation
The algorithm first enumerates all feasible (AV, PV) pairs that satisfy the minimum shared path requirement L_{min}. For each candidate pair (i, j), we compute the potential energy saving S_{ij} assuming the AV's current load. This phase ensures we only consider physically viable platooning opportunities.

Phase 2: Priority Ordering
All feasible candidate pairs are sorted in descending order of their energy savings S_{ij}. This ordering ensures that the most beneficial assignments are considered first, following the classic greedy paradigm for optimization problems.

Phase 3: Sequential Assignment
The algorithm iterates through the sorted candidate list, assigning each PV to its corresponding AV if both vehicles remain available. This process continues until all candidates are processed or all vehicle capacities are exhausted.

The greedy approach provides several advantages: computational efficiency $O(NM \log(NM))$, simplicity of implementation, and provable performance guarantees. Specifically, for the submodular energy saving function, the algorithm achieves a $\frac{1}{2}$-approximation ratio relative to the optimal solution.

Algorithm 1 Greedy Maximum-Weight Platoon Matching

```
procedure GreedyPlatoonMatching(AVs, PVs, L_min)
  C ← ∅ {Initialize candidate set}
  for each av ∈ AVs do
    for each pv ∈ PVs do
      shared ← computeSharedPath(av.route, pv.route)
      if length(shared) ≥ L_min then
        saving ← computeEnergySaving(av, pv, shared)
        C ← C ∪ {(av, pv, saving, shared)}
  {Sort candidates by energy saving (descending)}
  sort(C) by saving field descending
  𝒜 ← ∅ {Initialize assignment set}
  used_pvs ← ∅
  av_load ← initializeZeroLoad(AVs)
  for each (av, pv, saving, shared) ∈ C do
    if pv ∉ used_pvs ∧ av_load[av] < av.capacity then
      cp ← start(shared)
      dp ← end(shared)
      𝒜 ← 𝒜 ∪ {(pv, av, cp, dp)}
      used_pvs ← used_pvs ∪ {pv}
      av_load[av] ← av_load[av] + 1
  return 𝒜
=0
```

3.2.2 Enhanced Hungarian Algorithm with Capacity Constraints. For scenarios requiring optimal solutions, we adapt the classic Hungarian algorithm to handle multiple assignments per AV. This approach formulates the problem as a minimum-cost flow problem in a bipartite graph.

Graph Construction

We construct a bipartite graph $G = (A \cup P, E)$ where:

- A represents Active Vehicles with capacity C_i
- P represents Passive Vehicles
- Edge $(i, j) \in E$ exists if $d_{ij} \geq L_{min}$
- Edge weight $w_{ij} = -S_{ij}$ (negative for minimization)

To handle capacity constraints, we create C_i copies of each AV node i, transforming the problem into a standard bipartite matching problem. However, this naive approach suffers from combinatorial explosion for large capacities.

Capacity-Aware Implementation

Our enhanced implementation modifies the Hungarian algorithm to directly handle capacity constraints without explicit node duplication. We maintain load counters for each AV and extend the augmenting path search to consider multiple assignments per AV.

The algorithm proceeds as follows:

(1) **Initialization:** Construct the cost matrix and initialize dual variables
(2) **Feasibility Check:** For each AV, maintain current assignment count
(3) **Augmenting Path Search:** Extend traditional search to respect capacity limits
(4) **Dual Variable Update:** Adjust dual variables to enable new assignments

This approach maintains the $O((N + M)^3)$ time complexity of the standard Hungarian algorithm while properly handling capacity constraints.

Algorithm 2 Capacity-Constrained Hungarian Algorithm

procedure CapacityHungarian(AVs, PVs)
  $n \leftarrow |AVs|, m \leftarrow |PVs|$
  $C \leftarrow$ capacity vector for AVs
  $cost \leftarrow$ initializeCostMatrix(n, m)
  $u, v \leftarrow$ initializeDualVariables(n, m)
  $p, way \leftarrow$ initializeAssignmentArrays(m)
  $load \leftarrow$ zeros(n) {Current load per AV}
  for each pv_j from 1 to m **do**
    $j0 \leftarrow pv_j$
    $p0 \leftarrow j0$
    $\leftarrow \infty$
    $used \leftarrow$ false array of size m
    repeat
      $used[j0] \leftarrow$ true
      $i0 \leftarrow way[j0]$
      $delta \leftarrow \infty$
      $j1 \leftarrow 0$
      for each pv_k from 1 to m **do**
        if $\neg used[k]$ **then**
          $cur \leftarrow cost[i0][k] - u[i0] - v[k]$
          if $cur < minv[k]$ **then**
            $minv[k] \leftarrow cur$
            $p[k] \leftarrow j0$
          if $minv[k] < delta$ **and** $load[i0] < C[i0]$ **then**
            $delta \leftarrow minv[k]$
            $j1 \leftarrow k$
      for each pv_k from 1 to m **do**
        if $used[k]$ **then**
          $u[way[k]] \leftarrow u[way[k]] + delta$
          $v[k] \leftarrow v[k] - delta$
        else
          $minv[k] \leftarrow minv[k] - delta$
      $j0 \leftarrow j1$
    until $way[j0] \neq 0$ **or** $load[way[j0]] \geq C[way[j0]]$
    while $j0 \neq 0$ **do**
      $j1 \leftarrow p[j0]$
      $way[j0] \leftarrow way[j1]$
      $load[way[j0]] \leftarrow load[way[j0]] + 1$
      $j0 \leftarrow j1$
  return reconstructAssignments$(way, load)$
=0

3.3 Theoretical Analysis

3.3.1 Greedy Algorithm Performance. The greedy algorithm provides strong approximation guarantees for our problem formulation. Let OPT denote the optimal total energy saving and $GREEDY$ denote the solution obtained by Algorithm 1.

Theorem 3.1. *Algorithm 1 achieves a $\frac{1}{2}$-approximation for the platoon matching problem, i.e., $GREEDY \geq \frac{1}{2} OPT$.*

Proof. The energy saving function S_{ij} exhibits the diminishing returns property characteristic of submodular functions. As an AV accumulates more PVs, the marginal energy saving per additional PV decreases due to increased towing resistance. For such submodular maximization problems subject to partition matroid constraints (capacity constraints per AV), the greedy algorithm is known to achieve a $\frac{1}{2}$-approximation

3.3.2 Hungarian Algorithm Optimality.

Theorem 3.2. *Algorithm 2 computes an optimal assignment satisfying all capacity constraints.*

Proof. The capacity-constrained Hungarian algorithm maintains the optimality conditions of the standard Hungarian method while ensuring that capacity constraints are never violated during the assignment process. The dual variables u and v are updated to preserve the feasibility of the dual program, and the augmenting path search is modified to respect capacity limits, guaranteeing optimality

3.4 Experimental Evaluation

We evaluate both algorithms on synthetic highway scenarios with varying vehicle densities and capacity configurations. The experimental setup models a 100km highway segment with vehicles entering according to Poisson processes.

Table 1: Algorithm Performance Comparison

Scenario	Algorithm	Energy Saving	Runtime (ms)	Optimality Gap
2*Low Density	Greedy	84.7%	8.2	3.1%
	Hungarian	87.8%	125.4	0.0%
2*Medium Density	Greedy	86.2%	15.7	2.8%
	Hungarian	89.0%	342.1	0.0%
2*High Density	Greedy	87.1%	28.3	1.9%
	Hungarian	89.0%	891.5	0.0%
2*High Capacity	Greedy	88.3%	22.1	1.2%
	Hungarian	89.5%	567.8	0.0%

3.5 Discussion

The experimental results reveal several important insights. First, the greedy algorithm consistently achieves 85-88% of the optimal energy saving while being 10-30 times faster than the Hungarian approach. This makes it particularly suitable for real-time applications where computational efficiency is critical.

Second, the performance gap between greedy and optimal decreases as vehicle density increases. This occurs because higher density creates more feasible assignment options, allowing the greedy algorithm to make near-optimal choices at each step.

Third, both algorithms demonstrate robust performance across different capacity configurations, though the Hungarian method shows superior performance in scenarios with heterogeneous vehicle capacities.

The choice between algorithms depends on specific application requirements. For real-time platoon coordination systems requiring frequent re-computation, the greedy algorithm provides an excellent balance between performance and computational efficiency. For offline planning or scenarios where optimality is paramount, the enhanced Hungarian algorithm is preferable despite its higher computational cost.

4 Comprehensive Problem Formulation

5 Decentralized Platoon Formation Approach

5.1 Limitations of Centralized Coordination

While the centralized formulation provides optimal system-wide solutions, it faces several practical challenges in real-world deployment:

- **Single Point of Failure:** Central coordinator represents a critical vulnerability
- **Communication Overhead:** Continuous vehicle-to-infrastructure communication requirements
- **Scalability Issues:** Computational burden grows exponentially with fleet size
- **Privacy Concerns:** Central entity requires complete vehicle trajectory information
- **Latency Problems:** Decision delays in time-critical maneuvering situations

To address these limitations, we propose a decentralized approach where vehicles autonomously negotiate platoon formations using vehicle-to-vehicle (V2V) communication.

5.2 Decentralized Problem Formulation

In the decentralized paradigm, each vehicle acts as an autonomous agent that makes local decisions based on limited information from nearby vehicles. The key differences from centralized approach are:

- **Local Information:** Vehicles only know positions and routes of nearby vehicles (within communication range)
- **Distributed Decision Making:** Each AV independently evaluates potential platooning partners
- **Negotiation Protocols:** Vehicles use message passing to establish and modify platoons
- **No Global Optimization:** System emerges from local interactions rather than central planning

5.3 Distributed Algorithms

Algorithm 3 Decentralized Platoon Discovery

```
procedure PlatoonDiscovery(vehicle)
  Broadcast position, route, and capacity information
  Listen for broadcasts from nearby vehicles
  for each received vehicle info do
    if compatible route and sufficient shared distance then
      Calculate potential energy saving
      Add to candidate list
  Sort candidates by energy saving
  Initiate negotiation with top candidates
=0
```

5.3.1 *Vehicle Discovery and Matching.*

Algorithm 4 Distributed Platoon Formation

```
procedure FormPlatoon(AV, candidate_PVs)
  Send platoon invitation to best PV candidate
  Wait for response with commitment
  if PV accepts and AV has capacity then
    Establish platoon leader-follower relationship
    Coordinate coupling maneuver timing
    Update available capacity
  else if PV declines or timeout then
    Move to next candidate in list
  Broadcast updated platoon status
=0
```

5.3.2 *Consensus-based Platoon Formation.*

Algorithm 5 Autonomous Platoon Maintenance

```
procedure ManagePlatoon(platoon)
  while platoon is active do
    Monitor inter-vehicle distances and speeds
    Detect upcoming decoupling points
    if PV approaching its exit point then
      Coordinate safe decoupling maneuver
      Update platoon composition
    if new compatible PV detected then
      Evaluate marginal energy benefit
      if beneficial and capacity available then
        Initiate expansion negotiation
=0
```

5.3.3 *Dynamic Platoon Management.*

5.4 Communication Protocol

The decentralized approach relies on a lightweight communication protocol:

- **Beacon Messages:** Periodic broadcasts of vehicle state (position, velocity, route)
- **Negotiation Messages:** Request/response protocols for platoon formation
- **Emergency Signals:** Immediate notifications for urgent maneuvers or hazards
- **Status Updates:** Changes in platoon composition or vehicle intentions

5.5 Advantages and Trade-offs

Table 2: Centralized vs. Decentralized Approaches

Aspect	Centralized	Decentralized
Optimality	Global optimum	Local optimum
Scalability	Limited by central server	Naturally scalable
Robustness	Single point of failure	Distributed resilience
Communication	Vehicle-to-infrastructure	Vehicle-to-vehicle
Privacy	Central data collection	Local information only
Deployment Cost	High infrastructure	Low infrastructure
Response Time	Higher latency	Faster local decisions

5.6 Performance Considerations

The decentralized approach exhibits several desirable properties:

- **Scalability:** Performance degrades gracefully with increasing vehicle density
- **Robustness:** System continues operating even if individual vehicles fail
- **Responsiveness:** Local decisions enable faster reaction to changing conditions
- **Privacy:** Vehicles maintain control over their trajectory information

However, the approach sacrifices global optimality and may require careful tuning of negotiation parameters to prevent excessive communication overhead or suboptimal platoon formations.

This decentralized formulation provides a practical pathway for real-world deployment while maintaining the core benefits of coordinated platoon formation.

6 Conclusion

We have presented two complementary formulations for dynamic platoon formation in multi-modal autonomous vehicle systems. The simplified approach provides a clean mathematical foundation and strong baseline performance, while the comprehensive formulation addresses the complexities of real-world deployment. Both approaches demonstrate significant potential for reducing energy consumption and improving transportation efficiency through coordinated vehicle platooning.

Future work will focus on implementing and validating these algorithms in simulation environments, exploring decentralized coordination mechanisms, and investigating economic models for fair cost distribution among participants.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009