**Evaluating graph algorithms for optimal ride sharing solutions**

Derek Stearns

Abstract

*Since the start of the 21st century an exponential increase of interest has sprouted in the study of ride sharing methods for a multitude of reasons. It is the aim of this paper to understand the discoveries in the field as well as possible claims of finding optimized algorithms for a sample of objectives and to simulate one.*

**Keywords**: Ride sharing, route, minimum cost, algorithm, simulation

**Introduction**  
As interest in large projects such as hyper loops increase and increases in Electronic Vehicle (EV) manufacturing occur, additional focus on optimizing human travel continues to be at the forefront of study by public and private organizations. As shown by Martins et al in their paper *Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms* they show the exponential increase of terms such as “ride-sharing”, “car-sharing” and “carpooling” occurring in cited research from 2008 to 2019 with 0 to over 4000 citations, respectively. Generally speaking, works cited are using undirected graph models to designate nodes as any number of conditions or constraints such as the source, destination, driver, passenger, departure time, available seats of a car; and the edges are represented where each of the conditions satisfies the relationship in the objective or constraint. Examples of these edges include a road or path where weights can be applied for length or time to travel or general match ability of two vertices such as a driver and rider(s) (Dutta & Sholley, 2018). Understanding the general state of this body of research and the questions that have been asked, answered or optimized and unsolved is necessary to understand what topics might be useful to further the research in this wide field. This article attempts to understand the possible topics in the study of graph algorithms to solve optimization problems for objectives in ridesharing such as **minimum cost to the driver or riders** and **minimum cars utilized**, explore possible solutions proposed, and simulate the results of the compared algorithms.

**Review of Literature**

Earliest mention of the ridesharing problem comes from the 1940’s which required the problem to generalize into a static, pickup and delivery problem (Faruk et al., 2020). Ridesharing problem solutions resurge again in the 1970’s during a world oil shortage (A. Di Febbraro et al., 2013). This type of problem has been solved using a matching strategy and is still being studied today. A prime example is *Online Matching in a Ride-Sharing Platform* (Dutta & Sholley, 2018)*.* Dynamic matching research can also be found as recent as 2016 in Schrieck et al’s work, *A matching Algorithm for Dynamic Ridesharing* (Schreieck et al., 2016). As time has passed the evolution and expansion of the car sharing problem objectives and constraints are evident. By first reviewing Martins’ paper, a researcher in car-sharing optimization one can understand the breadth of possible topics and possible objectives worthy to be studied. The organization of the paper breaks down the possible objectives into several categories, covering 86 different research papers from a collection of 1,355 gathered. They first define terms. They offer generalizing the terms car-pooling and ride-sharing with the term “car-sharing” and cite specific objectives for car-pooling, such as route optimization from source to destination, as opposed to ride-sharing, which has a higher tolerance for detours for additional passengers. They also discuss and categorize the algorithmic methodology of each study into three main categories: exact, heuristic or metaheuristic, and other. Their proposal suggests utilizing three objectives and proposes using mixed continuous-integer linear programming in favor of finding more dynamic or metaheuristic methodologies. (Martins et al., 2021) Using a modified heuristic as introduced in pShare, by Huang et al, suggests further evidence why the *exact* methodology, or sometimes called simple matching, must be generally abandoned in favor of more metaheuristic solutions (Huang et al., 2022).

**Objectives and Methodology**

**Dynamic Systems**

The rapid expansion of the field of this study can be validated further by examining Di Febbraro et al’s research in one of the earliest modern suggestions of a dynamic ridesharing system (contrasted to “static” or “exact”). However, the term dynamic has been used at least since 1977 (Kornhauser et al., 1977) to describe solutions to ride sharing. Many cited works claim that modern research must include a dynamic component. While a bulletin board used in 1940 as described by Faruk et al is dynamic, the use of computers or smartphones in a dynamic way is relatively new; especially with the continued development of smart phone capabilities (Faruk et al., 2020).

**Simulation**

Additionally, an important factor to Martins et al, as the title highlights, is the identification of algorithmic methods that employ metaheuristic elements in addition to providing *simulation*. This is well illustrated with figure 8 in their paper where the authors attempt to display the tradeoffs of the collection of algorithms examined into rankings across categories Dynamism, Optimality, Speed, Flexibility, Scalability, and Uncertainty. While maybe not the pioneer in adding simulation to their research, the complexity of the systems has increased significantly over time, as evident in Martins’ research. As smartphones allow people to compute faster and in more dynamic ways, this creates additional potential considerations and types of objectives or variants. (Martins et al., 2021).

**Mixed Travel and Privacy**

Gu et al suggests that static problems are closely related to dynamic problems and can be used to solve dynamic problems by using solutions for static problems in an iterative manner (Gu et al., 2018). Additional novel variants include one from Martins et al that include accounting for the mixed type of travel possible, such as mixing ride shares and public transportation; and the concept of internet privacy users may demand when using a dynamic system such as on their smart phone introduced by Huang.

**Variants and Constraints**

Variants of the problem, beyond simple shortest path, include detouring constraints, rider preferences of company and drivers, the car type and capacity of the car, time bounds.

A table will be inserted to summarize the common constraints below:

**Solutions**

Now that the history of the problem has been reviewed, its evolution and variants, can be summarized. Perhaps the most generally similar and famous of ridesharing problems is the travelling salesman problem (TSP). Optimization of certain TSP problems are known to be NP-hard (Gu et al., 2018). Car sharing can be paralleled to the travelling salesman problem, which is known to require heuristics to reach desired maximal results [REF]. It is also recommended that simulation is the only reasonable way to provide evidence of superior heuristics as suggested by Martins et al. Gu et al explores this boundary including an adjustment to the classic car sharing problem by allowing any driver to also be a passenger. They supply requirements and proofs of simple minimization problems are still NP-hard, however, they also offer a polynomial solution for minimizing the number of drivers under certain conditions. Those conditions include a unique destination, detour tolerance rigidity, and a unique preferred path (Gu et al., 2018). pShare offers a minimum solution by minimizing the detouring route.

A variant of TSP, the vehicle routing problem (VRP), as mentioned by Google proposes methods to find optimal or sub optimal routes for multiple vehicles visiting a set of locations. Google provides sub-optimal solutions for general TSP problems with their OR tools package (*Vehicle Routing  |  OR-Tools  |  Google Developers*, n.d.). The specifics of which they expand upon into car-capacity constraints as well as time window constraints.

Martins et al supplies a list of metaheuristics that includes genetic algorithms, tabu search, and greedy randomized adaptive search procedures. Tabu search appeared often in the research for this article. One example is the research by Ben Cheikh-Graiet et al in “*A Tabu Search based metaheuristic for dynamic carpooling optimization”* (Ben Cheikh-Graiet et al., 2020)*.*

Concorde also supplies solutions that claim optimality for a variety of TSP problems with their TSPLIB library (*TSPLIB*, n.d.).

Below is a summary table of solutions implemented:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Problem Class** | **Objective/Variant** | **Algorithm** | **Source** | **Python Code Available** |
| TSP | Min Total Dist Travelled (TSP classic) | Tabu Search | AlgorithmExamples.com | Yes |
| Matching | Min Total Dist Travelled | Min Matching | Github | Yes |
| VRP | Min Total Dist Travelled |  | Google OR | Yes |
| VRP | Min Number of Drivers |  | Gu et al | No |
| VRP | Min Detour route | pShare | Huang et al | No |
| TSP | Multiple Solutions | TSPLIB | TSPLIB | No |
| DCP | Max Rider with Min Drivers | DyCos | Sondes et al | No |

**Simulation**

As it can be seen from the table above, there are three examples of minimizing where available python code can be used to simulate the problem. This paper will simulate these methods and report the results to provide additional support for the situation in which a particular method might be useful.

**Analysis**

**Conclusion**

A. Di Febbraro, E. Gattorna, & N. Sacco. (2013). *Optimization of Dynamic Ridesharing Systems*. <https://doi.org/10.3141/2359-06> This article was useful to get a picture of what possible types of systems were studied in the past.

ben Cheikh-Graiet, S., Dotoli, M., & Hammadi, S. (2020). A Tabu Search based metaheuristic for dynamic carpooling optimization. *Computers and Industrial Engineering*, *140*. <https://doi.org/10.1016/j.cie.2019.106217> This article was used to understand the functionality of Tabu Search.

Dutta, C., & Sholley, C. (2018). *Online Matching in a Ride-Sharing Platform*. This article shows an example of non-bipartite matching.

Faruk, O., Id, A., Gokasar Id, I., & Id, K. (2020). *Matching algorithm for improving ride-sharing by incorporating route splits and social factors*. <https://doi.org/10.1371/journal.pone.0229674>. This article gives a matching example as well as empirical evidence of better performance.

Gu, Q. P., Liang, J. L., & Zhang, G. (2018). Algorithmic analysis for ridesharing of personal vehicles. *Theoretical Computer Science*, *749*, 36–46. <https://doi.org/10.1016/J.TCS.2017.08.019> This article gives a great sample of algorithms and proofs to show np-hardness as well as an example of a solvable ride matching problem.

Huang, J., Luo, Y., Xu, M., Hu, B., & Long, J. (2022). pShare: Privacy-Preserving Ride-Sharing System with Minimum-Detouring Route. *Applied Sciences (Switzerland)*, *12*(2). <https://doi.org/10.3390/app12020842> This article gives an example of how research is implementing adjusted algorithms in integrate user privacy

Kornhauser, A. L., Mottola, P., & Stephenson, B. (1977). TRANSPORTATION EFFICIENCY AND THE FEASIBILITY OF DYNAMIC RIDE SHARING. *Transportation Research Record*, *650*. This article establishes early practice of dynamic considerations.

Martins, L. do C., de la Torre, R., Corlu, C. G., Juan, A. A., & Masmoudi, M. A. (2021). Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms. *Computers and Industrial Engineering*, *153*. <https://doi.org/10.1016/j.cie.2020.107080> This article is used as a basis for understanding the general state of research in the field of ride-sharing study.

Schreieck, M., Safetli, H., Siddiqui, S. A., Pflügler, C., Wiesche, M., & Krcmar, H. (2016). A Matching Algorithm for Dynamic Ridesharing. *Transportation Research Procedia*, *19*, 272–285. <https://doi.org/10.1016/J.TRPRO.2016.12.087> This article gives an example of simulated results of a matching algorithm that includes a dynamic component.

*TSPLIB*. (n.d.). Retrieved November 4, 2022, from <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/> This is a software package published by a university.

*Vehicle Routing  |  OR-Tools  |  Google Developers*. (n.d.). Retrieved November 4, 2022, from <https://developers.google.com/optimization/routing>. This is a software package published by Google

Possible additional research articles:

[**https://towardsdatascience.com/assigning-fastest-pick-ups-to-uber-drivers-with-linear-programming-8f8bd3c44c9a**](https://towardsdatascience.com/assigning-fastest-pick-ups-to-uber-drivers-with-linear-programming-8f8bd3c44c9a)

**https://medium.com/swlh/tabu-search-in-python-3199c44d44f1**