# The Self-Driving Pizza Car: Optimizing Control Algorithms for Autonomous Delivery Systems


Nischal Sinha,[1] Bryce Ferguson[2]
[1]Great Oak High School, 32555 Deer Hollow Way, Temecula, CA 92592
[2]Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106
Email: nish.d.sinha@gmail.com


UC SANTA BARBARA
Research Mentorship Program
Summer Sessions


## Abstract

Recent years have seen an increase in attention to autonomous vehicles, especially in the field of autonomous delivery using self-driving trucks and cars. While the creation of such autonomous vehicles has become fairly straight-forward, a far less explored question is what constitutes an ideal algorithm to determine delivery routes. One unique case of such delivery systems is that of self-driving pizza cars (SDPCs), wherein a vehicle must consider incoming orders, prices, wait times, and other factors, all while baking orders as the car travels. In this paper we pose a computer model to simulate an SDPC on a network of locations. We then develop both greedy and temporal difference (TD) control algorithms, prioritizing various properties related to customer experience and efficiency. Finally, we compare the delivery efficiencies of each algorithm. Ultimately, TD algorithms consistently outperformed their greedy counterparts, while there was no conclusive control which applied optimally over all metrics.


## Autonomous Vehicles, AMoD, and SDPCs

**Autonomous vehicles** are an increasingly popular possible solution to modernizing delivery systems & public transportation (autonomous mobility on demand, or AMoD)

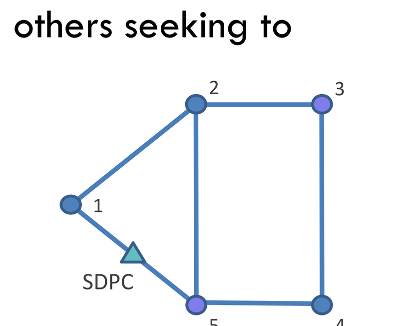
**Applications in Pizza Delivery**

- A multibillion dollar industry: $9.8 billion spent on delivery in the US in 2018 [1]
- Domino's, Pizza Hut, Zume Pizza, and others seeking to automate pizza delivery services
- Increases efficiency and thus profits

**The Self-Driving Pizza Car (SDPC)**

A hypothetical vehicle that:

- Receives stochastic orders
- Decides which order to satisfy
- Bakes pizza *as it drives to locations*
- Does not need to return to mother vertex between orders



A sample network of 5 locations and the roads between them. Here the SDPC is moving from location 1 to satisfy an order, in purple, at location 5.

**Current Control Algorithms**

Greedy algorithms vs Reinforcement learning (RL)

- Greedy algorithms are trained on specific data → hard to adapt to different or large networks [4]
- Temporal difference [TD(λ)] learns over time to optimize for any given network
- Current research into TD for AMoD is for individual cars & local decision making, not routing fleets [6]
- No existing algorithms consider time constraints of baking in addition to other factors

### Our Goals

1. Build out a program to model an SDPC operating on a graph of locations
2. Develop greedy and TD algorithms that dictate which orders the SDPC satisfies
3. Measure and analyze the efficiencies of the various algorithms operating on the model

## Framework & Stochastic Orders

- Weighted graph with $n$ nodes
- Nodes = locations
- Edges = distances
- Orders generate randomly over time and expire if left unsatisfied
- Components: bake time ($b$), price ($p$), current wait ($w$)

$$o_i = (b_i, c_i, w_i)$$

## Decision Making

***J:*** cost of executing system
***x:*** current state of the system at time $t$
***u:*** control which the system will take (moving to next node)
***f(x, u):*** estimation for the next state of the system after executing control c

$$u = argmin_u\, J(f(x, u))$$

**General goal:** minimize the accumulated sum of the costs over the course of the simulation

## Traveling

SDPC denoted by extra node on the graph
Begins traveling when decision for next node is made
**When traveling:** edges span from SDPC to previous node and from SDPC to current node, where weights represent distances
**If…**
The SDPC arrives at a node before the bake time is elapsed:

- wait at the node until the order is satisfied

The next location is more than 1 node away

- Travel along the graph to the next location, passing intermediate nodes but not delivering orders there

## Algorithms

Calculating cost $J$ for every possible state becomes too intensive for a high number of nodes and timesteps, so it must be **approximated** using algorithms

### Greedy

Approximates minimum $J$ by taking actions with the lowest **immediate** cost

- Does not consider future states
- Cost = Cost of moving from state $x$ to $x^+$

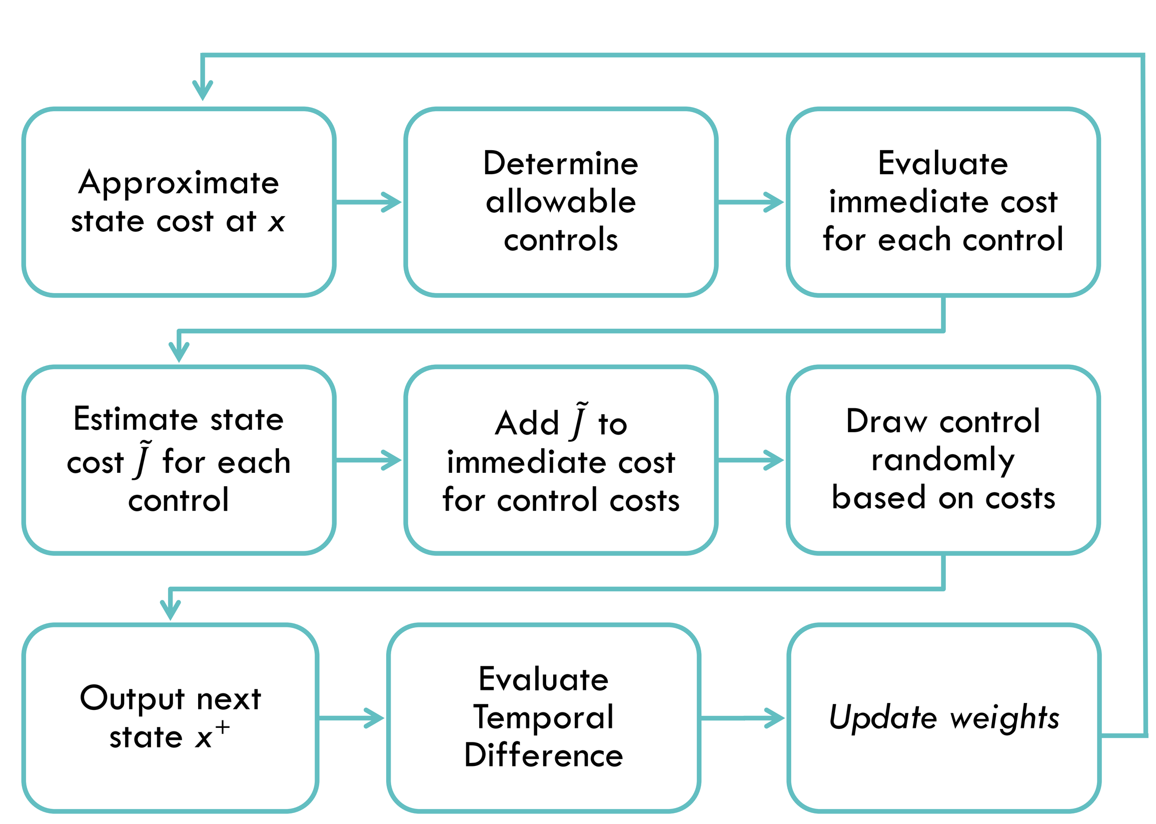
### Temporal Difference: TD(λ)

Minimizes total cost of system over course of entire simulation by considering **future states**

Calculating cost $J$ for every possible state becomes too intensive for a high number of nodes and timesteps, so instead, we **approximate** $J$ as:

$$\tilde{J}(x, r) = \sum_{i=1}^{i} r_i \cdot \phi_i(x)$$

**ϕ(x):** vector encoding designated features of the system at $x$
***r:*** weights assigned to each respective feature



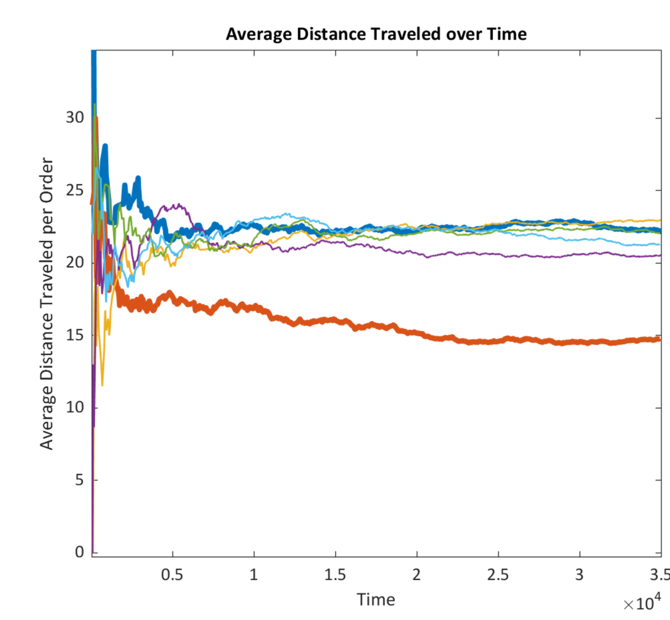

## Results

### Target Properties

- Minimum Distance
- Maximum Price
- Maximum Wait

**Algorithms Tested**

- Greedy, Min Distance
- TD, Min Distance
- Greedy, Max Price
- TD, Max Price
- Greedy, Max Wait
- TD, Max Wait

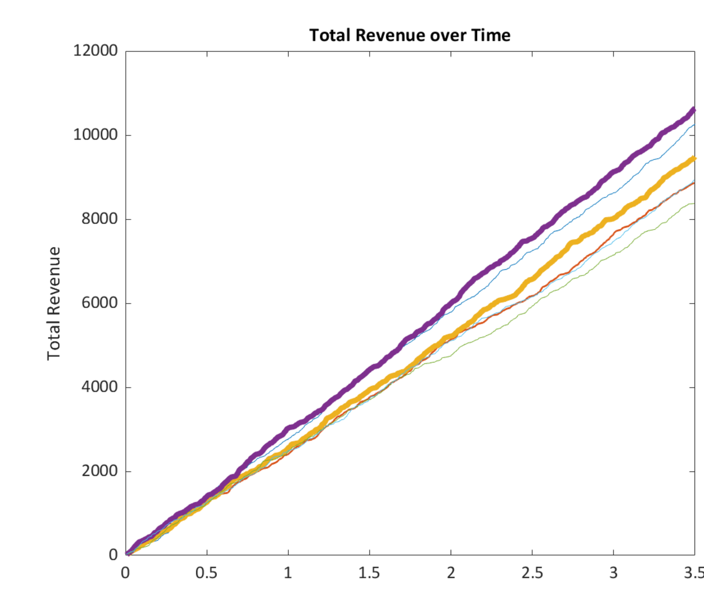


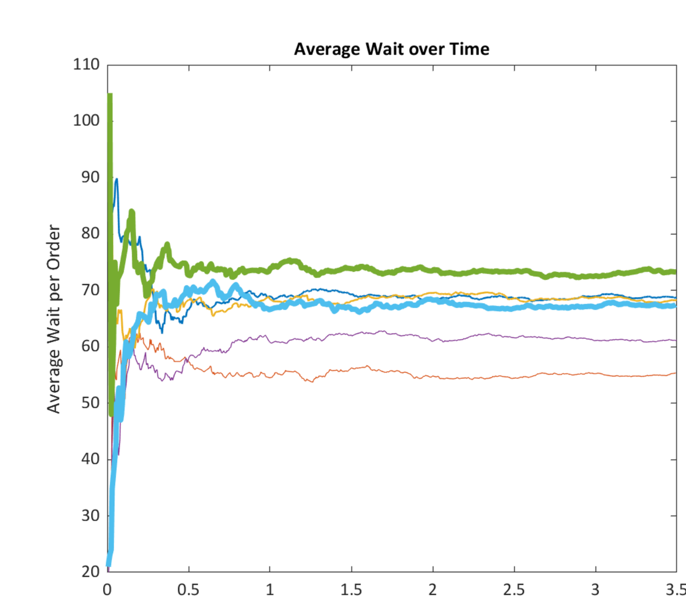

TD, Min Distance greatly outperforms the other algorithms, including Greedy, Min Distance

All TD algorithms have lower distances than greedy algorithms

TD, Max Price gains the most revenue, outperforming Greedy, Max Price

Greedy, Min Distance gains second most revenue




TD, Max Wait outperforms Greedy, Max Wait

TD, Min Distance yields lowest average wait

Both greedy and TD max wait algorithms fall in middle of the pack

## Conclusion & Future Work

- Overall best performing algorithm is **TD, Min Distance:** lowest average distance and lowest average wait time
- **TD, Max Price** is optimal for strictly maximizing revenue
- **TD algorithms** perform more optimally than greedy algorithms for autonomous vehicle routing on 7 of 9 tested metrics for SDPC efficiency and customer experience, but, as none of the test controls worked best universally, a multi-factor TD algorithm is likely needed to perform optimally across all tested metrics.

### Future Work

- Increase robustness of model with more factors
- Develop algorithm with optimal performance on all metrics
- Expand model to apply to SDPC fleets


**Acknowledgements:** This research was conducted under the UC Santa Barbara Department of Electrical and Computer Engineering. The authors would like to thank them and a number of individuals for their contributions to the research process: Dr. Lina Kim, our program director, provided a number of invaluable resources to guide our research, and both Dr. Michael Hughes and A S M Iftekhar, our TA, gave us specialized feedback to fine tune our research and its presentation. Finally, our lab mates Adit Shah and Siddharth Ganesan were valuable partners in navigating the research process. Without these people's support, this report would not have been possible.



**References:**
[1] "Pizza delivery: consumer spending US 2018," Statista. [Online]. Available: https://www.statista.com/statistics/259168/pizza-delivery-consumer-spending-in-the-us/. [Accessed: 14-Jul-2019].
[2] P. Holley, "Domino's will start delivering pizzas via an autonomous robot this fall," Washington Post, 17-Jun-2019.
[3] "Order Delicious Pizza and Have it Delivered Zume Fast," Zume Pizza, Order Online. [Online]. Available: https://zumepizza.com/. [Accessed: 14-Jul-2019].
[4] M. Han, P. Senellart, S. Bressan, and H. Wu, "Routing an Autonomous Taxi with Reinforcement Learning," in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16, Indianapolis, Indiana, USA, 2016, pp. 2421–2424.
[5] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-Finder: A Recommender System for Finding Passengers and Vacant Taxis," IEEE Trans. Knowl. Data Eng., vol. 25, no. 10, pp. 2390–2403, Oct. 2013.
[6] M. Pavone, "19 Autonomous Mobility-on-Demand Systems for Future Urban Mobility," p. 16.
[7] D. Bertsekas, Dynamic Programming and Optimal Control, 3rd ed., vol. 1. Athena Scientific, 2005.
[8] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving," arXiv:1610.03295 [cs, stat], Oct. 2016.
[9] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep Reinforcement Learning framework for Autonomous Driving," Electronic Imaging, vol. 2017, no. 19, pp. 70–76, Jan. 2017.
[10] K. Treleaven, M. Pavone, and E. Frazzoli, "Asymptotically Optimal Algorithms for Pickup and Delivery Problems with Application to Large-Scale Transportation Systems," p. 36, 2012.