

SAMPLE POSTERS


SHANGHAI 纽约大学

An Advanced Parking Navigation System for Downtown Parking A Practical Implementation

Zheng Zhang, Fangqing He, Prof. Zhibin Chen | Transportation Planning and Engineering

Abstract

Private vehicle penetration rate in urban area is rising at a higher rate than ever. Consequently, finding parking space has become drivers' daily headache. This research aims to mitigate the problem through establishing a real-time and intelligent navigation system between drivers and parking spaces. Adopting a two-sided matching algorithm, the system matches the drivers to their most appropriate parking spaces based on their real-time locations and parking preference, and thus preventing multiple drivers from being guided to the same parking space. This research will first build a fully functional, performance optimized backend to implement the two-sided matching algorithm and then develop a native APP on iOS mobile platforms. The APP will guide drivers to parking spaces according to their preference by considering their real-time position and traffic condition information. For demonstration, parking space occupancy will be simulated through a randomized algorithm since real-life parking data fetching requires sensor installation and serial communication, which is beyond the scope of this research.

Though many studies have focused on developing an advanced parking navigation system, most of the existing systems are not ready for practical implementation.

Problems of current solutions

- 1. Expensive computation
- 2. Strong assumption
- 3. Disclosing drivers' private information

Methods

The Two-Sided Match

We consider a downtown area where there is a finite number of parking spaces. Let $V = \{v_1, v_2, \dots, v_m\}$ and $S = \{s_1, s_2, \dots, s_n\}$ denote the set of drivers cruising for parking spaces and the set of open spaces at time t . Based on the current location, final destination and other personal favors (e.g., parking price, safety, etc.), open spaces are ranked based on each driver's preference. On the other hand, each open space has a preference ranking for cruising drivers, which is measured by the travel time for drivers to arrive at the space. That is, a space would prefer a driver who is closer to the one who is farther away.

Specifically, it consists of two procedures, including driver and space procedures (see Algorithm 2 and Algorithm 3), and they are executed iteratively and asynchronously.

Algorithm 2 Driver procedure

```

 $\begin{aligned}
& s \leftarrow \text{unmatched}; \\
& \text{terminal} \leftarrow \text{false}; \\
& \text{while } \text{terminal} \text{ do} \\
& \quad \text{if } s \text{ is unmatched and } P(v) \neq \emptyset \text{ then} \\
& \quad \quad v = \text{first}(P(v)); \\
& \quad \quad s.\text{dis} = \text{dis}(v, s); \\
& \quad \quad \text{sendMsgRequest}, v, s; \\
& \quad \quad v = v - 1; \\
& \quad \text{msg} = \text{getMsg}(); \\
& \quad \text{switch msg.type} \\
& \quad \quad \text{accept: do nothing;} \\
& \quad \quad \text{reject: } v = P(v) - \text{msg.sender}; \\
& \quad \quad v = \text{unmatched}; \\
& \quad \text{stop} : \text{terminal} \leftarrow \text{true}; \\
\end{aligned}$ 

```

Algorithm 3 Space procedure

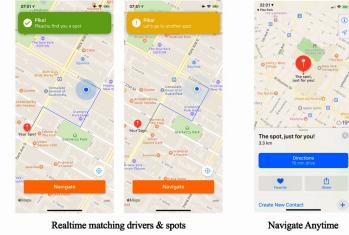
```

 $\begin{aligned}
& s \leftarrow \text{unmatched}; \\
& \text{terminal} \leftarrow \text{false}; \\
& \text{while } \text{terminal} \text{ do} \\
& \quad \text{if } s \text{ is unmatched} \text{ then} \\
& \quad \quad \text{sendMsgRequest}, s, v; \\
& \quad \quad s = s + 1; \\
& \quad \text{msg} = \text{getMsg}(); \\
& \quad \text{switch msg.type} \\
& \quad \quad \text{request: do nothing;} \\
& \quad \quad \text{if } v.\text{dis} > s.\text{dis} \text{ then} \\
& \quad \quad \quad \text{sendMsgAccept}, s, v; \\
& \quad \quad \quad s = v; \\
& \quad \quad \text{stop} : \text{terminal} \leftarrow \text{true}; \\
\end{aligned}$ 

```

Implementation

The two-sided matching algorithm ensures the user gets the most appropriate spot by real-time matching drivers with parking spots. If the newly-added user is closer to a parking spot that is already assigned to a further driver, the system will rematch the drivers and spots because the parking spot prefers a closer driver. This mechanism not only serves the parking spot's preference but also prevents the further driver finding his spot has been already taken by the closer driver by the time he arrives.



Project Title: Pika Park - Intelligent Parking Navigation System Zheng Zhang & Fangqing He



Background

Public Key Scheme $\xrightarrow{\text{Achieved by}} \text{Key Exchange Protocol} \xrightarrow{\text{Current Schemas}} \text{RSA-like} \text{ or } \text{Lattice} \xrightarrow{\text{An alternative}} \text{CA chaotic equations}$

Private Key for decryption, Public Key for Encryption, A set of calculation of two agents to enable communication, Brain by quantum, Still in research, A Nonlinear Equation Generator

Key Exchange Protocol

Alice:

Secret: $R = \langle P_1, M_1, N_1, \dots \rangle$
 $E(x) = \langle P(x), E(x) \rangle$
 $E(x) = \text{Gen}(R)$
 $E(x) = \text{Gen}(R)$
 $E(x) = E(R)$

Bob:

Secret: $k = E(x)$

Security Assumption

Assumption 1: One cannot distinguish an equation $E(x, m)$ generated by G from a randomly generated equation in $E(m)$ in polynomial time.

Assumption 2: One cannot solve a random $E(x, m)$ Equation in polynomial time.

Security Analysis

Theoretical Security: Average-case assumption, Nonlinear Equations, CA Rule 30, K-LIN, 3-SAT

Concrete Security: A graph showing the number of operations increasing exponentially with the number of variables.

2. GPU parallel search: One of the most significant improvements using this algorithm is that, the number of equations in a system does not affect the total time complexity. Instead, one should measure $\text{Grid}[E]$ (defined below). The algorithm is designed for high order equation system with only a few equations. Moreover, more equations require more advanced GPU. Yet the adversary time is still about 2 seconds according to their tests.

Efficiency Analysis

Theoretical Efficiency: Time for $\text{Gen} = \text{Gen}' = O(n^2)$

Theoretical analysis on space efficiency showed: Scheme E is space efficient

Concrete Efficiency: The space complexity in 10000 test is shown left below.

Future Study

Key generation speed: So far the public key generation speed is rather slow, even if with polynomial space efficiency. It is because in currently implementation, the operation of algebra, and equations used Computer Algebra System (CAS), which takes a longer time than numerical operation. Moreover, the generation of a random invertible $R \times n$ matrix in $GF(2)$ is also not efficient, as the current algorithm first generate a $R \times n$ matrix in $GF(2)$, then calculate the determinant, if determinant = 0 generate a new one.

Security proof: Although the promising result of concrete security, the scheme still lacks rigorous mathematical proof due to lack of study in average-case cryptography field. As discussed above, the study of its relation to random 3-SAT might be worth. Yet, the best way might be in the investigation of the theoretical applicability of average-case cryptography assumption.

Project Title: A New Public Cryptography Design Based on CA Chaotic System Name: Gengyu Chen