Design of Intelligent Parking System in Urban Environment

Yanfeng Geng and Christos G. Cassandras

Abstract
We are interested in designing an intelligent parking system for an urban environment, where all drivers can easily find a satisfactory spot based on their personal criteria while also ensuring that the city's parking capacity is efficiently utilized. This is formulated as a dynamic resource allocation problem with many random events, such as drivers generating parking requests, parking spots becoming vacant/occupied, traffic events, etc. We make use of Receding Horizon Control (RHC) to solve this problem. At each decision point, we solve a Mixed Integer Linear Program (MILP) to make a potential allocation. When the state changes, we solve a new problem at the beginning of the receding horizon. An efficient method is proposed to deal with conflict resolution when a parking spot is optimal for more than one driver. We have also built a simulation environment for the complete parking system operating under our proposed optimization problem solution.

Motivation

Necessity
• Convenient for drivers to find better parking spots
• Efficient & intelligent management of parking resources

Feasibility
• GPS to detect car position and destination location
• Technology to detect parking spot status, reserve spots, etc
• Communication between drivers and central system

Intelligent Parking Problem Set Up

User request
• Cost upper bounded
• Walk distance upper bounded

System's objective
• Satisfy as many requests as possible
• Optimize a given system utility function

Step One: relax \( \sum_{i \in F_t} x_i \leq 1 \forall i \in F_t \) and add penalty to cost function

\[
\min \sum_{i \in F_t} \sum_{j \in J} J_{ij} x_{ij} + \sum_{i \in F_t} \sum_{j \in J} J_{ij} x_{ij} + \sum_{i \in F_t} \gamma \left( x_{ij} - 1 \right)
\]

Original problem becomes feasible. In the relaxed solution, remove "conflict" resources in \( \sum_{i \in F_t} x_i > 1 \forall i \in F_t \)

Step two: change conditions from

\[
\sum_{i \in F_t} x_i = 1 \forall i \wedge W_i \leq R_i
\]

and re-solve the original problem.

Simulations

• Simulate the central system control panel
  • Random request time and locations
  • Random resource occupancy time
  • CPLEX to solve MILP problem

Future Work

• Extensions
  • Estimate future event times
  • Decile decision interval T
  • Resource pricing control (e.g., to prevent overfilled parking lot)

Realization in our Robotic Urban-like Environment (RULE)