Algorithm of Smart Parking Planning

Team 8 : Shi Hu, Jixiong Deng, Liyu Chen

Problem Description

Smart Parking System always collects real-time data (pricing, parking availability, etc.) from sensors and navigates users to proper parking spaces with their preferences.

- Reduce Traffic Congestion
- Increase Urban Mobility
- Improve City Environment
- Raise Parking Revenue

Our bio-inspired algorithm should fulfill its duty on approprivate navigation.

- Attempt to solve existing technical difficulties
- Tradeoff with current state-of-art work

Problem Description

Importances of Smart Parking System:

- Economics Impact
  - Time value
    - ~ 30% vehicles wastes 7.8 minutes on seek of parking spot
  - Land use (% of total CBD area)
    - 18% in New York City
    - 31% in San Francisco
    - 81% in Los Angeles
- Environmental Impact
  - Vehicle cruising for parking spot (in small area of L.A.)
    - Burn 47000 gallons of gasoline
    - Produce 730 tons of carbon dioxide

Problem Description

Challenges of Smart Parking System:

- Basics Infrastructure
  - As unified as possible
- Poor Compatibility
  - Linear Assignment vs Generalized Assignment
  - P vs NP
- Lack of User Engagement
  - The more engaged, the more controllable

Problem Formulation and Modeling

- The parking system receives queries in real time.
  - If each query is treated separately, greedy algorithm can be used.
  - Fail to achieve system-level efficiency
- Hold a number of queries in a time slice and process them altogether.
  - Assign each vehicle a parking lot.
- Linear assignment problem
  - Two sets of equal size A, T
  - A cost function C
  - Find a bijection A -> T to minimize C
  - Can be solved in polynomial time

Problem Formulation and Modeling

- Assume, in a time slice, there are M vehicles and N available parking lots.
  - V = {v1,...,vM}, P = {p1,...,pN}
  - vi.start, vi.dest, vi.hours; pi.hr, pi.limit, pi.max;
  - A driving time matrix D, Dij = drive time between vi and pj
  - A walking time matrix W, Wij = walking time between pj and vi.dest
  - A rate matrix R, Rij = rate for vi to park at pj

$$D_{M,N} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,N} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,N} \end{pmatrix}$$

$$r_{ij} = \begin{cases} v_i.hours * p_j.hr & \text{if } v_i.hours * p_j.hr < p_j.max \\ p_j.max & \text{otherwise} \end{cases} \tag{2}$$

- Define a cost matrix C
  - Cij = the cost for vi to park at pj

$$c_{ij} = \begin{cases} d_{ij} + w_{ij} + \alpha * r_{ij} * w_{ij} & \text{if eligible } (v_i.h <= p_j.limit \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

  - *α * rij * wij* is the penalty term to model the trade-off between parking rate and walking time

Problem Formulation and Modeling

- Define a solution matrix X

$$x_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is guided to } p_j \\ 0 & \text{otherwise} \end{cases}$$

- Define the total cost

$$cost = \sum_{i=1}^{M} \sum_{j=1}^{N} c_{ij} \times x_{ij}$$

- Find a X to minimize the total cost subject to

$$\begin{cases} \sum_{j=1}^{N} x_{ij} = 1 \\ \\ \sum_{i=1}^{M} x_{ij} \leq 1 \end{cases}$$

State of Art Solution

- Hungarian method
  - The well-known Hungarian method can solve linear assignment problems in O(n^4), which was later improved to O(n^3)

H. W. Kuhn, “THE HUNGARIAN METHOD FOR THE ASSIGNMENT PROBLEM,” IBryn Yaw College, 1955.

Solutions using bio-inspired algorithms

- Genetic Algorithm
  - Is inspired by the process of natural selection
  - Generate high-quality solutions by relying on biologically inspired operators such as mutation, crossover and selection.
  - Is suitable for discrete optimization.
  - Needs a proper objective function with constraints built in

Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press. ISBN 9780585030944.

Solutions using bio-inspired algorithms

- Objective function for GA

Input : Decision vector x of length M where x_i is denoting that the v_i is assigned to p_{x_i}. Cost matrix $C_{M,N}$

Output: Total cost $cost$

Initialize a decision matrix $Y_{M,N}$ with zeros

while i *in range of* M **do**

  $Y_{i,x_i} = 1$

end

$cost = \sum y_{ij} * c_{ij}$

$penalty = 0$

for *each* j **do**

  if $\sum_{i=1}^{M} y_{ij} > 1$ **then**

    $penalty = \infty$

  end

end

return $cost + penalty$

Solutions using bio-inspired algorithms

- Particle Swarm Optimization
  - Optimizes a problem by iteratively trying to improve a population of candidate solutions
  - Uses a position-velocity update method.
  - Each particle's movement is influenced by its known local best position and global best position.
  - More suitable for unconstrained continuous problem.

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]$$

J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

Dataset - On-street Metered Parking in LA City

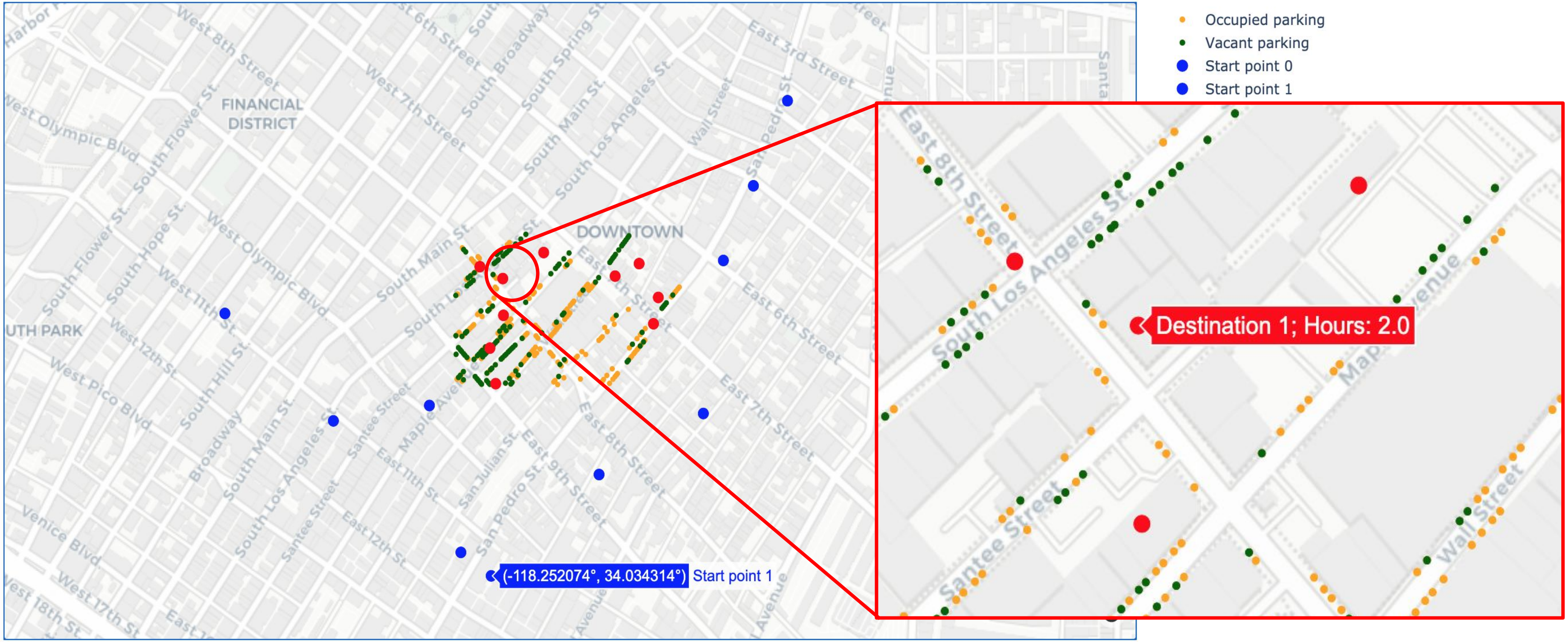
LADOT Metered Parking Inventory & Policies Transportation

LADOT Parking Meter Occupancy Transportation

Occupancy - Live Feed

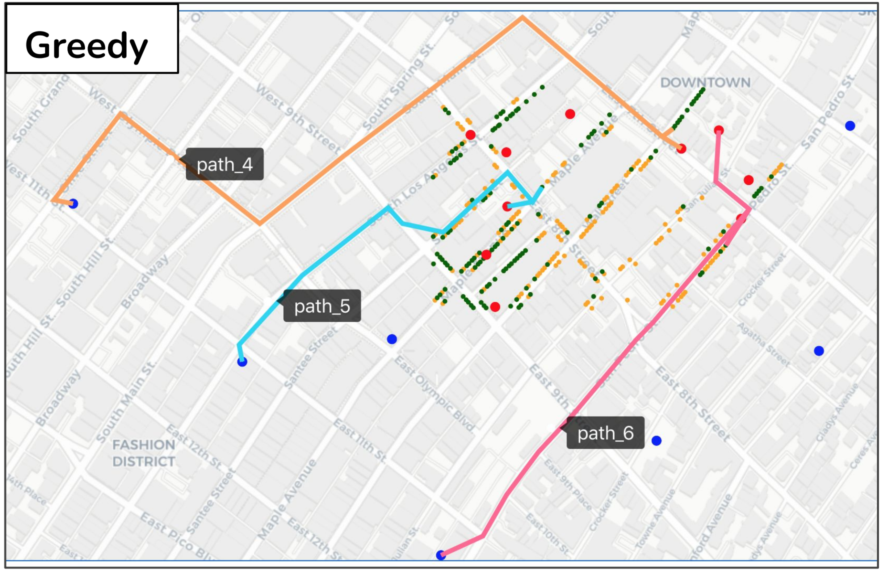
Field Name	Value/Type
SpaceID	String
Latitude/Longitude	Float/Float
OccupancyState	0 - Vacant; 1 - Occupied
HourlyRate	$/hour
HourLimit	#hours
RateType	FLAT, JUMP, SEASONAL, Time-of-Day (TOD)
MaxRate	Max $ within hour limit

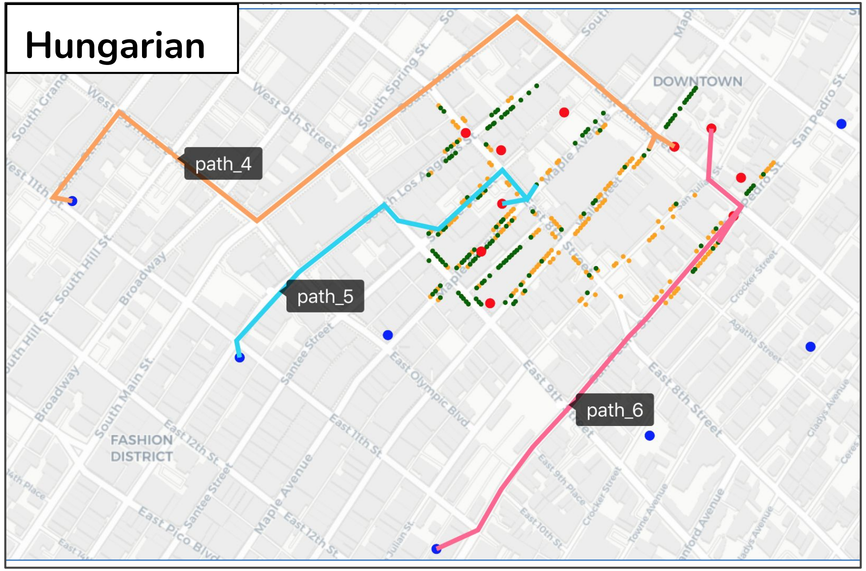
"LA Express Park™", LA Express Park™, 2020. [Online]. Available: http://www.laexpresspark.org/la-city-open-data/.



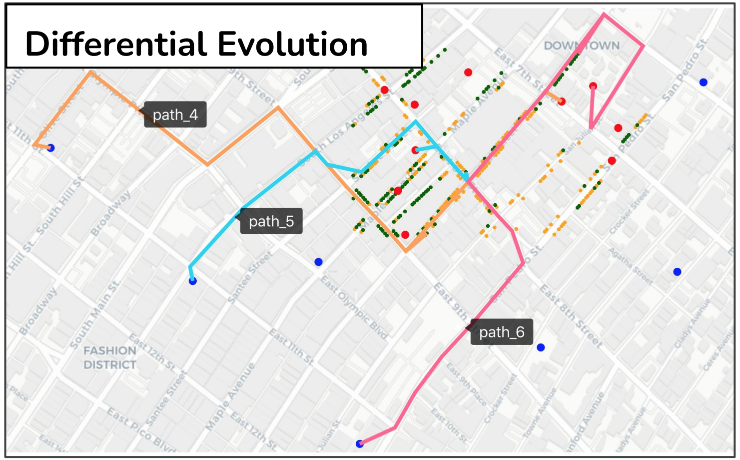
Simulation of 10 drivers given the following information:
(start point coordinates, destination coordinates, parking hours)

Experimental results of the comparative study

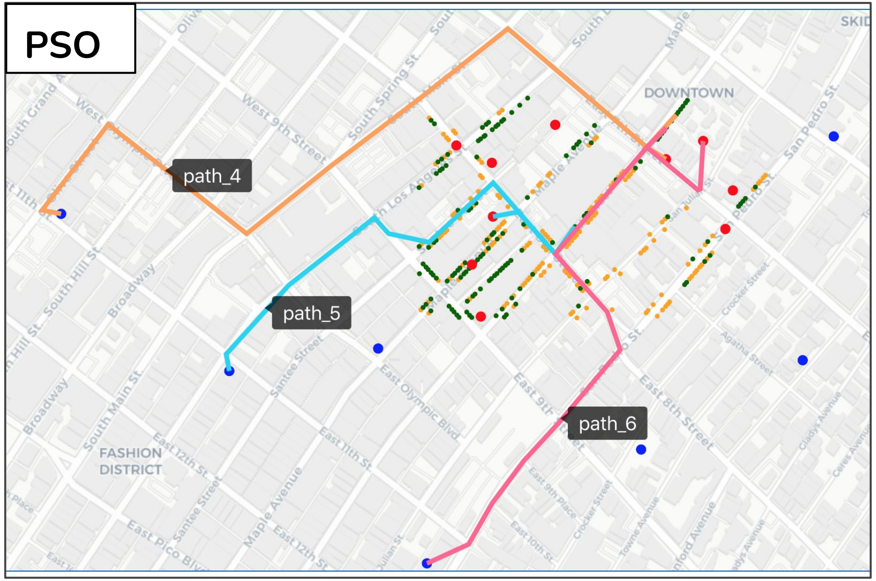




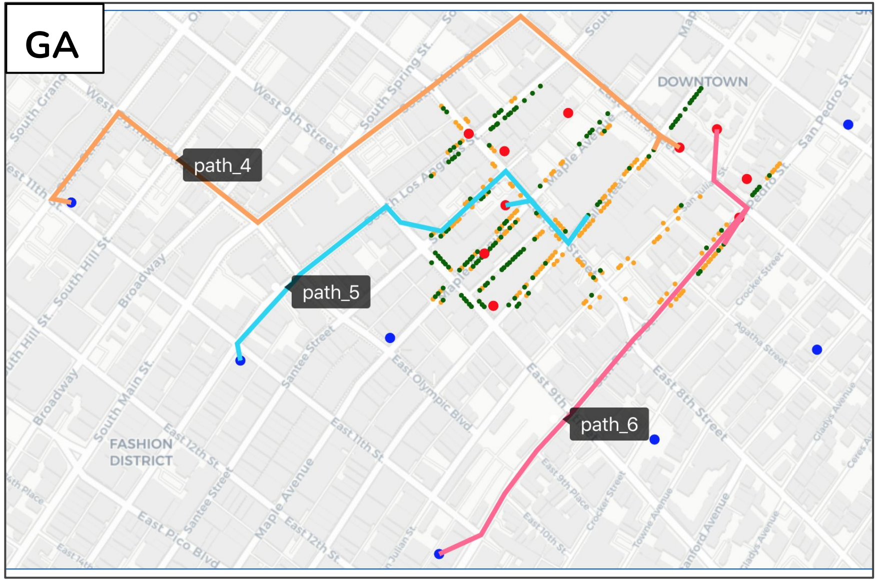
Continue...



Differential Evolution
path_4
path_5
path_6
FASHION DISTRICT



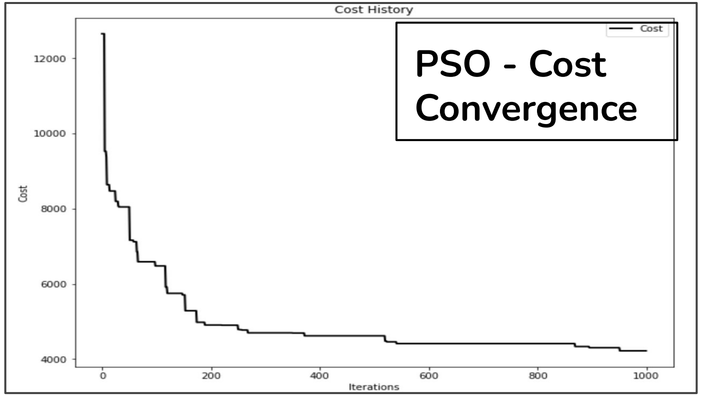
PSO
path_4
path_5
path_6
DOWNTOWN
FASHION DISTRICT

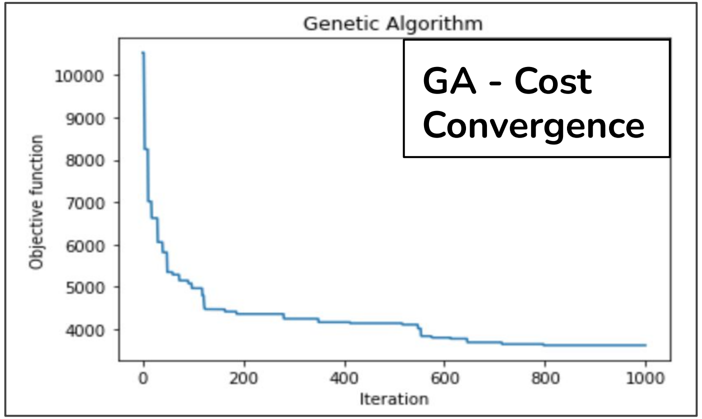


GA
path_4
path_5
path_6
DOWNTOWN
FASHION DISTRICT

Performance of algorithms

Algorithms	Total Cost	Run Time (sec)
Greedy	3490.92	0.00031972
Hungarian	3479.15	0.00041604
DE	8890.26	13.7536
PSO	4219.9	40.3403
GA	3619.14	59.606





Conclusion and future work

- Hungarian has the best performance on solving linear assignment problem
- PSO and GA are relatively doing well on cost optimization, but can be time consuming

Future Recommandation:

- Assigning vehicles to parking lots with capacity greater than 1 is a generalized assignment problem which is NP-hard
- Bio-inspired algorithms are expected to do better than linear algorithms on solving generalized assignment problem
- More infrastructures are needed in the smart parking system to collect real-time availability data in each parking lot

Q&A