网约车平台下自动驾驶车队规模与人工司机配备优化模型构建与求解

在萝卜快跑等无人驾驶服务的新挑战下，传统网约车平台为增强其竞争力也考虑购入一定量的自动驾驶车辆为乘客提供出行服务，为增强其乘客的市场占有率并最大化其利润，网约车平台管理公司应如何确定其自动驾驶车队的规模以及保留多大的人工司机数量呢？本研究尝试通过数学建模以及数值模拟的方式回答这个问题。主要任务包括：前期调研、模型构建、求解与算例分析。

符号	说明
$\mathrm{N_{AV}}$	平台自动驾驶车辆数量
$\mathrm{N_{HV}}$	平台人工司机数量
N_{HV}^{A}	平台潜在的人工司机数
α	人工司机的工资比例
N_t^{ts}	t 时段内运输的司机数
P_t^{AV}	t 时段内自动驾驶车辆的单位运价
P_t^{HV}	t 时段内人工司机的单位运价
W_t^{AV}	t 时段内自动驾驶车辆的等待时间
W_t^{HV}	t 时段内人工司机的等待时间
T	运输途中的行驶时间
D_t^{AV}	t 时段内对自动驾驶车辆的需求数量
D_t^{HV}	t 时段内对人工司机的需求数量
u_t^{AV}	t 时段乘客选择自动驾驶车辆的预期收益
u_t^{HV}	t 时段乘客选择人工司机的预期收益
D_t^{A}	t 时段平台潜在的乘客数量
N_t^{AV}	t 时段空闲的自动驾驶车辆的数量
N_t^{HV}	t 时段空闲的人工司机的数量
β_{AV}	自动驾驶车辆的工作时长比例
β_{HV}	人工司机的工作时长比例
V_t	t 时段内车辆行驶的平均速度
L_t	t 时段内平均行驶距离
V_c	速度缩减系数
V_f	自由流速度
N_P	t 时段的背景车辆数量

决策变量：

（1）自动驾驶车辆数量$\mathrm{N_{AV}}$

（2）人工司机工资（人工司机的工资比例）

（3）自动驾驶车辆的运价P_t^{AV}和人工司机的运价P_t^{HV}

司机与自动驾驶车辆空车状态的界定：

车辆处于运输状态、匹配状态均认为被占用。

模型约束：

（1）人工司机数量约束，人工司机与自动驾驶车辆数量之和为常数

$$N_{HV} = N_{HV}^{A} e^{b\alpha}$$

人工司机数量N_{HV}与潜在的司机数量N_{HV}^{A}和平台给人工司机的报酬比例 α 成正相关，b为参数。

（2）乘客选择自动驾驶车辆或人工驾驶的数量约束

$$D_t^{HV} = D_t^{A} \frac{\exp(\theta u_t^{HV})}{1 + exp(\theta u_t^{HV}) + \exp(\theta u_t^{AV})}$$

$$D_t^{AV} = D_t^{A} \frac{\exp(\theta u_t^{AV})}{1 + exp(\theta u_t^{HV}) + \exp(\theta u_t^{AV})}$$

$$u_t^{HV} = u - P_t^{HV} - \varepsilon(W_t^{HV} + T)$$

$$u_t^{AV} = u - \gamma - P_t^{AV} - \varepsilon(W_t^{AV} + T)$$

乘客选择自动驾驶车辆或者人工驾驶的数量与乘客选择自动驾驶车辆或者人工驾驶的收益有关，通过 Logit 模型联系起来。

θ为 Logit 模型的灵敏度参数，控制效用差异对选择概率的影响。

ε表示乘客对等待时间和行驶时间的敏感系数。

t 时段乘客选择自动驾驶的收益与在 t 时段自动驾驶车辆的单位运价P_t^{AV}、t 时段自动驾驶的等待时间W_t^{AV}、运输途中的行驶时间 T 有关；

t 时段乘客选择人工驾驶的收益与 t 时段人工司机的单位运价P_t^{HV}、t 时段人工驾驶的等待时间W_t^{HV}、运输途中的行驶时间 T 有关。

u 为乘客的固定收益，γ为乘客选择自动驾驶承担安全隐患的风险参数。

（3）空闲人工司机数量以及空闲自动驾驶车辆数量约束

$$\widetilde{N}_t^{AV} = \beta_{AV} N_{AV} - D_t^{AV}(T + W_t^{AV})$$

$$\widetilde{N}_t^{HV} = \beta_{HV} N_{HV} - D_t^{HV}(T + W_t^{HV})$$

若时段 t 的总需求为D_t^{AV}，则总占用时间为$D_t^{AV}(T + W_t^{AV})$。

空闲自动驾驶车辆数量等于所有可用的自动驾驶车辆数量减去正在进行运输任务的自动驾驶车辆数量，β_{AV}为自动驾驶车辆可用状态的比例；

空闲人工司机数量等于所有可用的司机数量减去正在进行运输任务的司机

数量，β$_{HV}$为司机为可用状态的比例。

（4）等待时间约束

$$W_t^{AV} = \frac{\omega}{\sqrt{\widetilde{N}_t^{AV}}} = \frac{\omega}{\sqrt{\beta_{AV} N_{AV} - D_t^{AV}(T + W_t^{AV})}}$$

$$W_t^{HV} = \frac{\omega}{\sqrt{\widetilde{N}_t^{HV}}} = \frac{\omega}{\sqrt{\beta_{HV} N_{HV} - D_t^{HV}(T + W_t^{HV})}}$$

车辆的等待时间与其相对应车辆的空闲车辆数成反比，ω为参数。

（5）平均行驶时间、平均速度约束

$$\mathrm{T} = \frac{\mathrm{L_t}}{\mathrm{V_t}}$$

$$V_t = V_f - V_c * [(\widetilde{N}_t^{HV} + \widetilde{N}_t^{AV} + D_t^{HV}(T + W_t^{HV}) + D_t^{AV}(T + W_t^{AV}))\rho + N_P]$$

V_f为自由流速度，无交通干扰时的理论最大速度。

V_c为速度缩减系数，表示每单位交通密度对速度的衰减量。

ρ 为将平台车辆活动转换为等效交通密度的比例因子。

$\mathrm{N_P}$为时间段 t 内不属于平台的其他车辆数。

目标函数：

$$\underset{P_t^{HV}, P_t^{AV}, \alpha, \mathrm{N_{HV}}}{Max} U = \sum_{t \in T} P_t^{HV} D_t^{HV} * (1 - \alpha) + P_t^{AV} D_t^{AV} - N_{AV} c$$

c 为单辆自动驾驶车辆的购置成本。

忽略平台的运营成本，构建以平台的利益最大为目标的目标函数。平台的利益等于平台进行运输活动获取的净收益减去购买自动驾驶车辆的成本。

算法设计：

1.基于遗传算法来求解平台最优策略模型的具体步骤

第 1 步：

输入种群规模N，问题维度D，最大迭代次数G，种群个体每个维度的最大值X_MAX和最小值X_MIN以及交叉概率$CROSSOVER_RATE$和变异概率$MUTATION_RATE$

第 2 步：

初始化种群$P=\{X_1,X_2,\dots,X_N\}$，每个个体的每个维度在X_MIN到X_MAX范围内随机生成

第 3 步（遗传算法主循环）：

$\boldsymbol{For\ generation\ g\ =\ 1\ to\ G}$ **do**

step 1（适应度评估）：

$\boldsymbol{For\ Each\ individual\ X_i}$ **do**

依据公式，通过牛顿迭代法计算$Trst$，并求解由D_t^{AV}，D_t^{HV}，N_t^{es}，N_t^{ts}，W_t^{AV}，W_t^{HV}构成的六元非线性方程组，最后计算适应度值$F(X_i)$ =目标函数值

step 2（轮盘赌选择）：

构造概率分布$p_i=\frac{F(X_i)}{\sum_{k=1}^{N}F(X_i)}$，生成新种群$P^*$：

$\boldsymbol{For\ i=1\ to\ N}$ **do**

生成随机数 $r\sim U(0,1)$

选择满足 $\sum_{k=1}^{N}p_i\ge r$的最小 m

用新种群 P^* 替换种群 P

step 3（算术交叉）：

$\boldsymbol{For\ i=1\ to\ N/2\ \ do}$

$\boldsymbol{If\ rand()<CROSSOVER\_RATE}$：

随机选择父代个体X_a，X_b

生成子代：

$$X_a' = aX_a+(1-a)X_b$$

$$X_b' = aX_b+(1-a)X_a$$

其中 $\alpha\sim U(0,1)$

step 4（自适应变异）：

$\boldsymbol{For}$ **每个个体** X_i $\boldsymbol{do}$

$\boldsymbol{For}$ **每个维度** j $\boldsymbol{do}$

为每个维度 j 设置初始标准差$\sigma_j^{init}=0.05*(X_{max}^j-X_{min}^j)$

其中 X_{max}^j 和 X_{min}^j 分别是个体第 j 维的上界和下界。

$If\ rand() < MUTATION\_RATE$:

根据当前进化代数 g 和最大进化代数 G，计算当前标准差：

$$\sigma_j(g) = \sigma_j^{init} * (1 - \frac{g}{G})$$

从高斯分布$N(0, \sigma_j(g))$中取一个随机数Δ，对当前个体的第 j 维进行更新：

$$X_{i,j} \leftarrow X_{i,j} + \Delta$$

step 5（移民策略）：

$\boldsymbol{If\ g\ mod\ 10 = 0}$ ：

找到种群中适应度最差的 20%的个体，对于每一个最差个体，用随机生成的新个体进行替换。

step 6（记录最优解）：

更新全局最优个体和最优适应度

End

2.基于 Kriging SBO 模型算法来求解平台最优策略模型的具体步骤

Kriging SBO（基于 Kriging 代理模型的优化） 是一种结合高斯过程回归（Kriging）和序列优化策略的高效优化方法，尤其适用于计算成本高昂的问题。其核心思想是通过构建目标函数的代理模型，减少对真实模型的直接调用次数，从而以较低的计算代价逼近全局最优解。

第 1 步（输入）：

输入变量维度 D，最大评估次数N_{max}和初始样本点数 n，Kriging 代理模型参数σ和θ_d，样本每个维度的最大值X_MAX和最小值X_MIN，

第 2 步（初始化）：

利用拉丁超立方采样生成初始数据集

第 3 步（序列优化主循环）：

For* 总评估次数 $\boldsymbol{N < N_{max}}$ *do

① 训练 Kriging 模型

1. 构建协方差矩阵K

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ . & . & . & . \\ . & . & . & . \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{pmatrix}$$

其中元素$k(x_i, x_j)$表示样本点x_i和x_j的相关性

$$k(x_i,x_j)=\sigma^2\exp\left(-\sum_{d=1}^{D}\frac{(x_i^d-x_j^d)^2}{2{\theta_d}^2}\right)$$

2. 计算权重向量a_i

$$a_i=K^{-1}y_i$$

K^{-1}表示协方差矩阵K的逆矩阵，y_i表示第i个样本点计算出的函数值

3. 计算当前最优目标函数值f_{best}

$$f_{best}=\max(y_i)$$

② 优化采集函数（EI）来寻找下一个点

1. 全局随机采样 1000 个点$\{x_c\}_{c=1}^{1000}$,并计算每个采样点的 EI 值

$$EI(x_c)=\left(f_{best}-\mu(x_c)\right)\Phi(z_c)+\sigma(x_c)\phi(z_c)$$

其中$\mu(x_c)$和$\sigma(x_c)$分别为预测点x_c的均值和方差

$$\mu(x_c)=k_c^T K^{-1}y_c$$

$$k_c^T=[k(x_c,x_1),k(x_c,x_2),\dots,k(x_c,x_n)]$$

$$\sigma^2(x_c)=k(x_c,x_c)-k_c^T K^{-1}k_c$$

$\Phi(z_c)$和$\phi(z_c)$分别表示预测点x_c的标准正态分布的累积分布函数和标准正态分布的概率密度函数

$$z_c=\frac{f_{best}-\mu(x_c)}{\sigma(x_c)}$$

选择 EI 值最大的候选点作为初始点x_{init}

2. 以x_{init}为初始点通过梯度下降法寻找使 EI 值最大的点x^*

③ 评估新点并更新数据集

调用真实目标函数计算目标函数值$f(x^*)$，并将$(x^*,f(x^*))$加入数据集，生成 n+1 维的协方差矩阵，重新训练 Kriging 模型

End

3.基于 RBF 模型算法来求解平台最优策略模型的具体步骤

第 1 步（输入）：

输入变量维度 D，最大评估次数N_{max}和初始样本点数 n，选定 BRF 模型中的核类型为高斯核，并输入核函数参数ε和探索参数ϵ，样本每个维度的最大值X_MAX和最小值X_MIN，

第 2 步（初始化）：

利用拉丁超立方采样生成初始数据集

第 3 步（序列优化主循环）：

$\boldsymbol{For}$ **总评估次数** $\boldsymbol{N < N_{max}\ do}$

④ 训练 RBF 模型

4. 构建计算权重ω_i的线性方程组

$$\begin{bmatrix} \varphi(||x_1 - x_1||) & \varphi(||x_1 - x_2||) & \dots & \varphi(||x_1 - x_n||) \\ \varphi(||x_2 - x_1||) & \varphi(||x_2 - x_2||) & \dots & \varphi(||x_2 - x_n||) \\ . & . & . & . \\ . & . & . & . \\ \varphi(||x_n - x_1||) & \varphi(||x_n - x_2||) & \dots & \varphi(||x_n - x_n||) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ . \\ . \\ \omega_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix}$$

x_n为第 n 个样本点的值，y_n为第 n 个样本点的观测值，$\varphi(\cdot)$表示以高斯核为核函数的径向基函数表示为：$\varphi(r) = e^{-\varepsilon r^2}$。$\omega_n$表示第 n 个样本点的权重

5. 构建 RBF 模型

RBF 模型的预测值是所有中心点处径向基函数的加权线性组合：

$$\mu(x) = \sum_{i=1}^{n} \omega_i * \varphi(||x - x_i||)$$

6. 计算当前最优目标函数值f_{best}

$$f_{best} = \max(y_i)$$

⑤ 优化采集函数（EI）来寻找下一个点

3. 全局随机采样 1000 个点$\{x_c\}_{c=1}^{1000}$，并计算每个采样点的 EI 值

$$EI(x_c) = Max\{\mu(x_c) - f_{best} - \epsilon, 0\}$$

选择 EI 值最大的候选点作为初始点x_{init}

4. 以x_{init}为初始点通过梯度下降法寻找使 EI 值最大的点x^*

⑥ 评估新点并更新数据集

调用真实目标函数计算目标函数值$f(x^*)$，并将$(x^*, f(x^*))$加入数据集，重新训练 RBF 模型

End