



云南大学
YUNNAN UNIVERSITY

云南大学硕士论文答辩

车联网环境下能耗优先的任务调度算法研究

答辩人：李涵

导师：张学杰教授

专业：计算机软件与理论

学号：12021115016

1. 研究背景

2. 主要工作

2.1 问题建模与基因算法求解

2.2 基于贪心法调整的离散侏儒猫鼬算法的任务调度

3. 总结与未来展望



研究背景

未来将迎来“万物互联”的时代。车联网是“万物互联”时代的典型应用，汽车的功能已经超出了传统的交通工具范畴，变成了一个智能互联的计算系统。因此需要大量的稳定的计算资源来保证车辆的服务质量。

可以通过将任务调度到其他车辆的方式来节约资源，利用任务量与能耗之间的非线性关系，来降低所有车辆的能耗。

任务调度的最优化目标有：

- （1）最小化调度时延。时延是车联网十分重要的指标，不仅关乎用户的体验还关乎用户的安全。
- （2）最小能量消耗。减少能耗可以提高车辆的行驶里程，缓解“里程焦虑”。



研究背景

任务调度的常用算法包括基于强化学习的任务调度、基于博弈论的任务调度和基于启发式算法的车任务调度。

文献[1]提出了一种基于深度强化学习任务调度方案调度目标未为最小化能耗。文献[2]使用深度强化学习方法用于解决分配网络资源问题。文献[3]提出了一种基于深度强化学习的方案来解决重定向和调度决策问题。

- [1] CHENG M, LI J, NAZARIAN S. Drl-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers[C/OL]//2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). 2018: 129-134. DOI: 10.1109/ASPDAC.2018.82 97294.
- [2] NING Z, DONG P, WANG X, et al. Deep reinforcement learning for vehicular edge computing: An intelligent offloading system[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(6): 1-24.
- [3] NING Z, DONG P, WANG X, et al. Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme[J]. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(4): 1060-1072.



研究背景

文献[4]将任务调度子问题建模为精确势博弈并提出了一种多智能体分布式深度确定性策略梯度来实现纳什均衡。文献[5]将问题表述为潜在的博弈并证明它存在纳什均衡，设计了一种去中心化算法用于在博弈中找到纳什均衡。文献[6]将多用户调度决策问题建模为非合作博弈，证明该问题存在纳什均衡并提出了一种基于机器学习技术的调度算法来寻找该纳什均衡点。

[4] XU X, LIU K, DAI P, et al. Joint task offloading and resource optimization in noma-based vehicular edge computing: A game-theoretic drl approach[J]. Journal of Systems Architecture, 2023, 134: 102780.

[5] HE Q, CUI G, ZHANG X, et al. A game-theoretical approach for user allocation in edge computing environment[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 31(3): 515-529.

[6] CAO H, CAI J. Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach[J]. IEEE Transactions on Vehicular Technology, 2017, 67(1): 752-764.

。



研究背景

文献[7]提出了一种高效的启发式算法解决移动云计算系统中总体成本的非凸二次规划问题。文献[8]开发了一种基于粒子群的启发式算法，用来解决异构网络中多用户多服务器的两层计算调度策略问题。文献[9]使用蚁群算法来计算出到达目的地的最佳路径。

- [7] CHEN M H, DONG M, LIANG B. Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints[J]. IEEE Transactions on Mobile Computing, 2018, 17(12): 2868-2881.
- [8] HUYNH L N, PHAM Q V, PHAM X Q, et al. Efficient computation offloading in multi-tier multi- access edge computing systems: A particle swarm optimization approach[J]. Applied Sciences, 2019, 10(1): 203.
- [9] BUI K H N, JUNG J J. Aco-based dynamic decision making for connected vehicles in iot system [J]. IEEE Transactions on Industrial Informatics, 2019, 15(10): 5648-5655.

研究背景



现有的研究中没有考虑存在不可调度任务的情况以及任务调度公平情况。在进行任务调度时，必然存在一些不可调度的与安全相关的任务。本文旨在存在不可调度任务的情况下，以提高能源利用效率为目标的同时兼顾公平，研究如何利用启发式算法解决车联网中的任务调度问题。



主要工作——问题建模

任务不可分情况

在任务不可分时，问题的优化目标是通过调度各个车辆的任务，最小化所有能耗的平方，该问题最终被建模为如下形式：

$$\begin{aligned} \min \quad & \sum_{t=1}^T \sum_{i=1}^N (E_{it}^{blnc})^2 \\ \text{s. t.} \quad & E_{it}^{blnc} = \sum_{j=1, j \neq i}^N x_{ij}^t \cdot P_0 \cdot \frac{d_{jt}}{W \log(1 + \text{SNR})} + \sum_{j=1, j \neq i}^N x_{ji}^t \cdot P_0 \cdot \frac{d_{it}}{W \log(1 + \text{SNR})} \\ & \quad + \lambda_i \cdot f_{it}^3 \cdot \Delta T, \quad i = 1, \dots, N \\ & \sum_{j=1}^N x_{ij}^t \cdot r'_{jt} \leq C_{it} \\ & \sum_{i=1}^N x_{ij}^t \geq 1 \\ & f_{it} \geq 0 \\ & x_{ij}^t \in \{0, 1\} \end{aligned}$$

能耗与功率、任务量、频率的关系

分配的任务不能超过车辆本身的计算容量

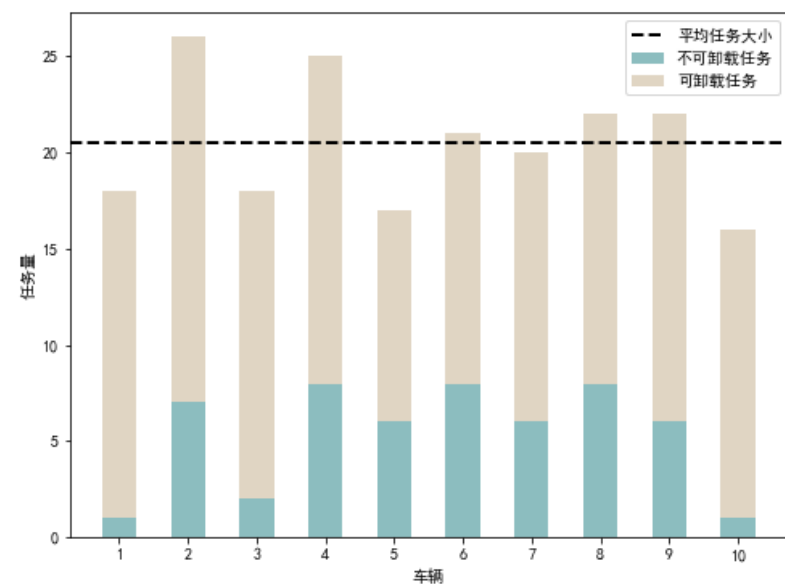
每个任务都必须能被完成

主要工作——问题建模

任务可分情况

任务可分情况下，可以分两种情况讨论，一种是大任务情况，另一种是小任务情况。

两种情况下，该问题都能够直接求解数学规划的方式解决。



小任务情况示意图

根据均值不等式： $\sqrt[n]{\frac{\sum_{i=1}^n a_i^n}{n}} \geq \frac{\sum_{i=1}^n a_i}{n}$ ，可得： $\frac{\sum_{i=1}^n a_i^3}{n} \geq \left(\frac{\sum_{i=1}^n a_i}{n} \right)^3$

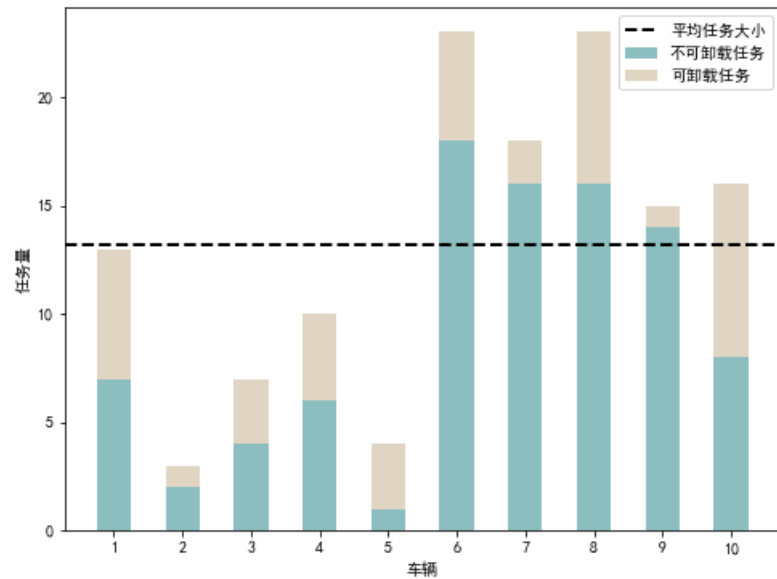
其中当 $a_1 = a_2 = \dots = a_n = \frac{C}{n}$ 时，等号成立。

因此，要使 $\sum_{i=1}^n a_i^3$ 最小，需要让 $\sum_{i=1}^n a_i$ 的值尽可能

平均分配给 n 个数，即令 $a_1 = a_2 = \dots = a_n = \frac{C}{n}$ 。

主要工作——问题建模---任务可分

当任务高于平均任务大小（大任务）时，该问题需要进行讨论，是否还需要将再给有大任务的汽车分配任务。



大任务情况示意图

根据拉格朗日乘数法，构造带拉格朗日乘数的拉格朗日函数，并对其求偏导且等于零，得到以下结果：

$$\begin{cases} \frac{\partial L}{\partial a_i} = 3a_i^2 - \lambda_i + \mu = 0, & i = 1, \dots, m \\ \frac{\partial L}{\partial a_i} = 3a_i^2 + \mu = 0, & i = m + 1, \dots, n \\ \frac{\partial L}{\partial \lambda_i} = a_i - x_i = 0, & i = 1, \dots, m \\ \frac{\partial L}{\partial \mu} = \sum_{i=1}^n a_i - C = 0 \end{cases}$$

最优的分配方案是：

- （1）对于本地任务是大任务的车辆，将他们所有可调度的任务调度出去。
- （2）其余车辆再均分剩余的任务。

评测指标

1. 总能耗。

优化目标

3. 性能指标，即节能的百分比。

$$P = \min \left(100 \cdot \left(1 - \frac{\sum_{i=1}^N E_{it}^{blnc}}{E_{loc}} \right) \right), t = 1, \dots, T$$

其中， E_{loc} 是本地执行任务所需要消耗的能量。

2. 算法运行时间。

固定迭代次数的时间

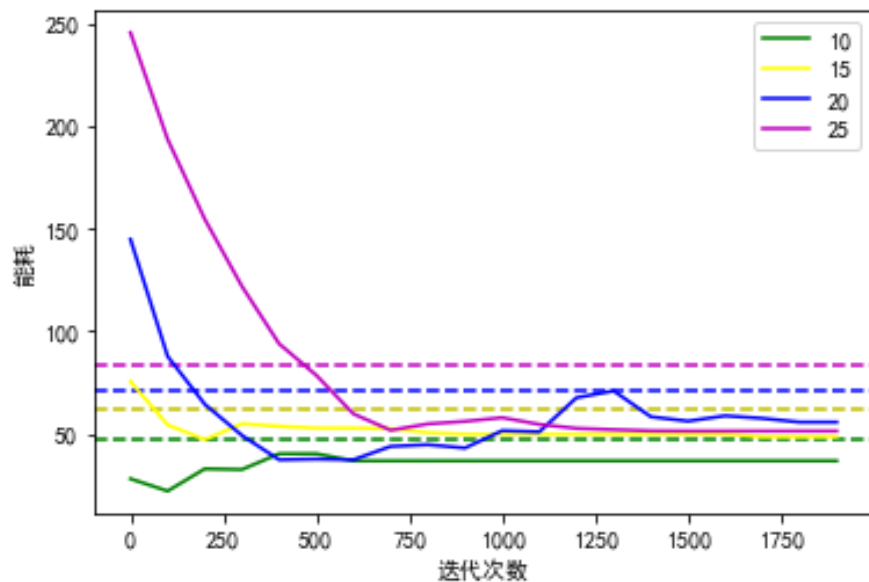
4. 公平系数。

$$FC = \max \sqrt{\frac{1}{N} \sum_{i=1}^N \left(E_{it}^{blnc} - \bar{E} \right)^2}, t = 1, \dots, T$$

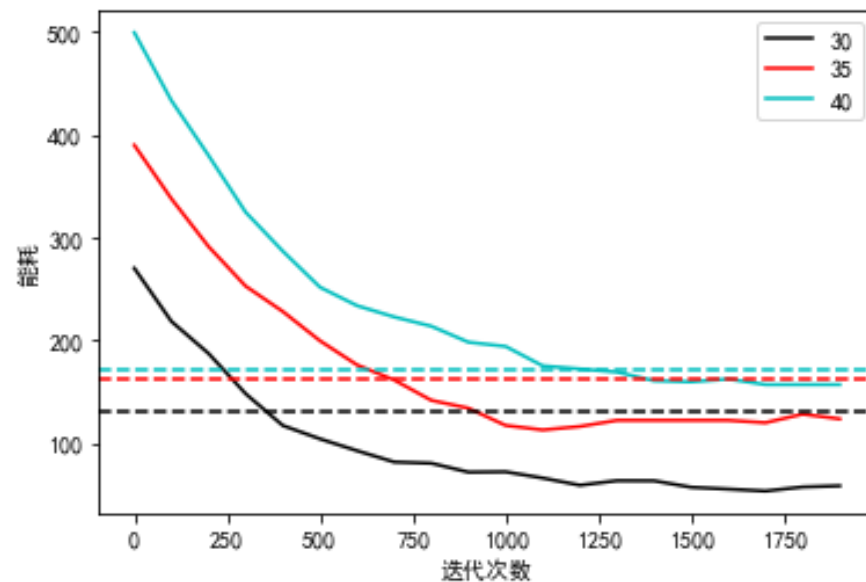
其中， \bar{E} 是所有车辆能耗的平均值

实验一评估

随着车辆数目的增加，收敛所需要的迭代次数也在增加。因此需要合理地调整迭代次数，因为车辆数量的变化会显著的影响算法的运行时间。



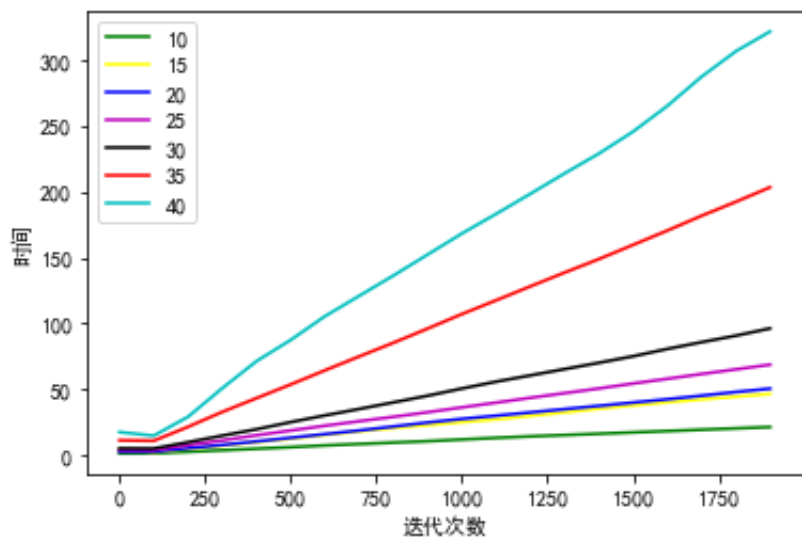
车辆能耗结果 1



车辆能耗结果2

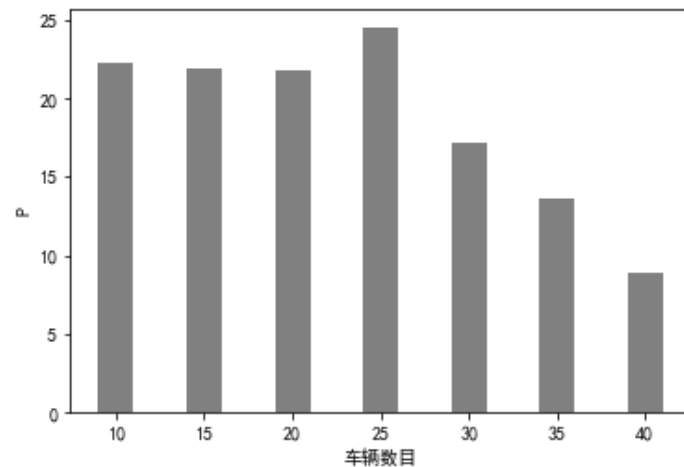
实验一评估

随着车辆数量的增加，迭代的次数也在增加。
经过 100 次迭代以后，所有数目运行时间和迭代次数呈线性关系。但是算法的运行时间和车辆之间的关系是非线性的。



算法运行时间结果图

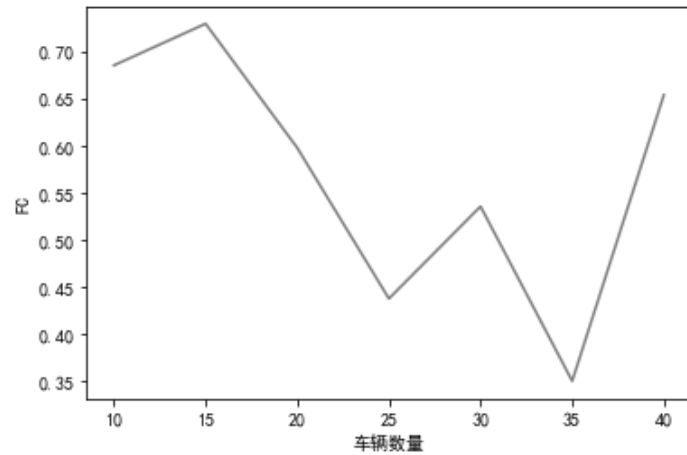
总体而言，随着车辆数目的增加，节能的作用也越来越小。



性能度量 P 值图

实验一评估

分配方案较为公平，其数值都不大，且不同的车辆数目下差别不大。



平衡因子 FC 值图

第一部分工作对问题完成了建模，并使用基因算法解决了该问题，但基因算法存在计算量太大、计算时间过长等问题。

第二部分工作通过重新定义运算符的方式，将侏儒猫鼬算法修改为解决离散问题的算法，解决本文提出的离散形式的问题。

重新定义的运算符有双目加法运算符、双目减法运算符、双目乘法运算符、单目调整运算符。



主要工作——算法设置

双目加法运算符 (\oplus)。该运算符借鉴了基因算法中的交叉操作。

双点交叉的具体操作过程：假设有两个父代个体分别是：

$$S1 = [2, 1, 2, 1, 2, 1]$$

$$S2 = [1, 2, 1, 2, 1, 2]$$

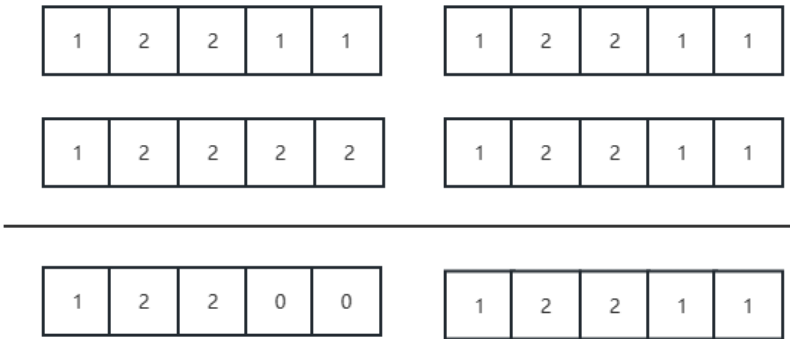
随机选择两个交叉点位置 $k1 = 2$ 和 $k2 = 5$ ，然后将两个父代个体在这两个位置之间的基因信息进行互换，得到新的子代个体 $S3$ 和 $S4$ ：

$$S3 = [2, 2, 1, 2, 1, 1]$$

$$S4 = [1, 1, 2, 1, 2, 2]$$

双目减法运算符 (\ominus)。解中每一个变量相应元素的值是否不同。

如果是，则获得其值，如果不是，则置为 0。也就是说，获得两个解中相同的部分，不同的部分置为 0。

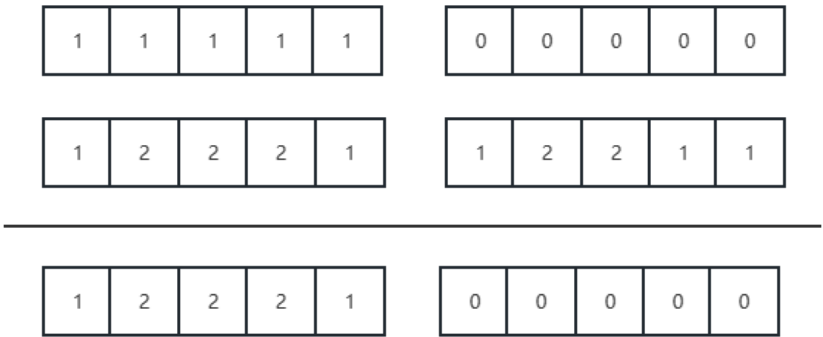


双目减法运算符示意图

主要工作——算法设置

双目乘法运算符 (\otimes)。

每个变量的取值分别相乘，第一个变量的取值是 n 维的向量，向量每个维度其值只有 0 和 1。

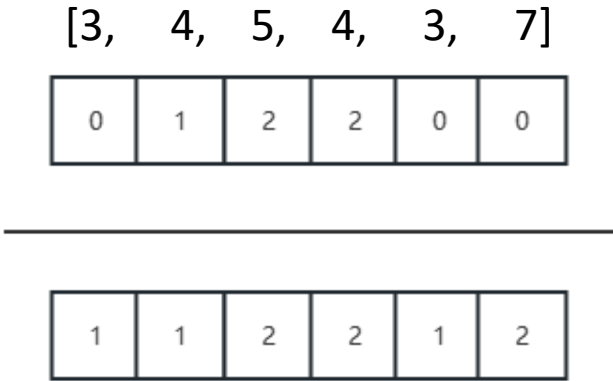


双目乘法运算符示意图

单目调整运算符 \odot 。对 变量中所有为 0 的元素进行处理时间最短调整。

单目运算符的优先级低于双目运算符。

以下任务的所需时间分别为[3, 4, 5, 4, 3, 7]。



单目运算符示意图

主要工作——算法设置

Algorithm 1: 离散侏儒猫鼬算法流程图

输入: 种群数量 n , 目标函数 $f(x)$, 最大迭代次数 \max_iter , 变量的维度 $ndim$

输出: x_{best}

```
1 初始化种群数量  $n$ , 初始化目标函数  $f(x)$ , 初始化种群  $X$ , 初始化最大迭代次数  $\max\_iter$ , 初始化保姆的数量  $bs$ 
2  $n = n - bs$ 
3 设置保姆交换参数  $L$ 
4 for  $i = 1 : \max\_iter$  do
5   计算种群的适应度
6   设置计时器  $C$ 
7   通过下列公式, 选择出  $\alpha$  群体中的雌性首领  $X_\alpha$ 
8     
$$\alpha = \frac{f(u_\alpha)}{\sum_{i=1}^n f(u_i)}$$

9   对种群中的每一个个体, 通过下列公式, 计算候选的觅食位置
10     
$$X_{t+1} = \odot RVS(ndim, p_t) \otimes X_t \oplus X_\alpha$$

11   评估每个新值  $X_{t+1}$ 
12   用下列公式计算睡眠丘陵
13     
$$sm_t = \frac{f(u_{t+1}) - f(u_t)}{\max(|f(u_{t+1})|, |f(u_t)|)}$$

14   用下列公式计算睡眠丘陵的平均值
15     
$$\varphi = \frac{\sum_{i=1}^n sm_t}{n}$$

16   用下列公式计算移动向量
17     
$$p_t = e^{-\frac{C}{L}}$$

18   如果  $C \geq L$ , 交换保姆并重新初始化
19   用如下公式进行更新
20     
$$X_{t+1} = \begin{cases} \odot RVS(ndim, \max(\varphi, CF)) \otimes X_t \oplus X_\alpha & \text{if } \varphi_{t+1} > \varphi_t \\ \odot RVS(ndim, \min(\varphi, CF)) \otimes X_t \oplus X_\alpha & \text{else} \end{cases}$$

21   更新  $X_{best}$ 
22 end
23 return  $X_{best}$ 
```

离散侏儒猫鼬算法流程如下:

- (1) 进行参数初始化。
- (2) 对于每一轮迭代进行更新:
 - (2-1) 计算适应度, 选出雌性首领。
 - (2-2) 通过更新公式, 进行更新。其中, RVS 是维度为变量维度, 其值只有0或者1的向量。 $p_i = e^{-\frac{\varphi}{10}}$
 - (2-3) 计算睡眠丘和睡眠丘平均值。
 - (2-4) 计算出概率 pi , 其中, $CF = \frac{C}{L}$ 。
 - (2-5) 如果 $C \geq L$, 交换保姆并重新初始化。
- (3) 输出最佳的种群个体。

实验二评估指标

1. 总能耗。
2. 算法运行时间。
3. 性能指标，即节能的百分比。
4. 公平系数。

和实验一相同的评价指标

5. 种群多样性。

$$D = \sqrt{\sum_{j=1}^m D_j^2}, D_j = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$$

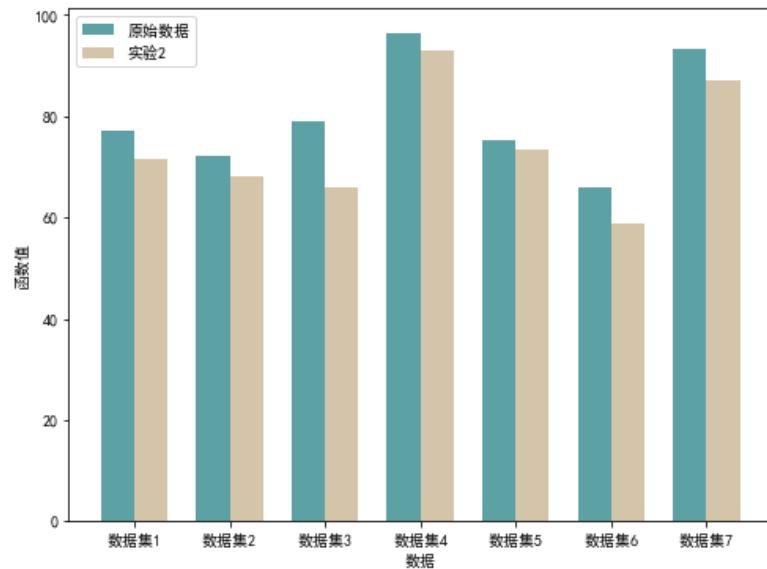
和实验一不同的评价指标

其中， m 是解的维度， n 是种群的数量， x_{ij} 是第 i 个粒子的第 j 个维数。

实验二评估——消融实验

消融实验一：验证解中某些变量的取值发生改变对目标函数的影响。

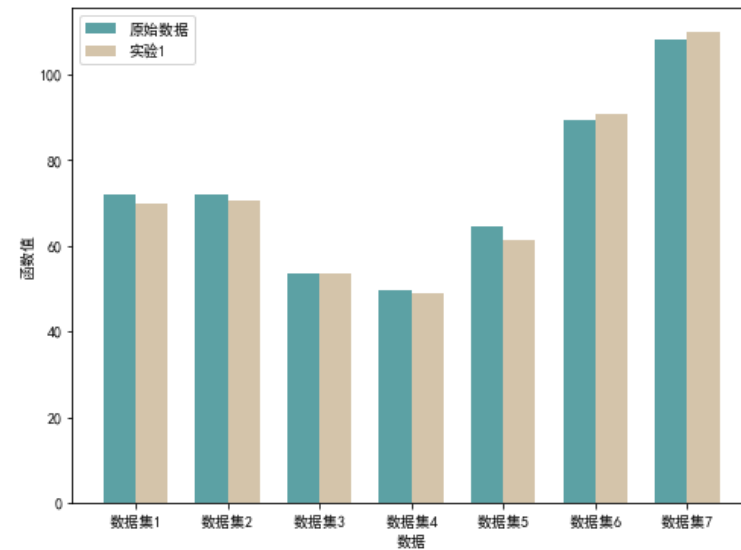
- (1) 随机生成初始解，且保证解的合法性，并计算目标函数值。
- (2) 对每个解的某些位置的变量，交换他们的取值。
- (3) 计算完成操作后的解所对应的目标函数，生成图像。



消融实验一实验结果图

消融实验二：两个解进行交叉行为对目标函数的影响。

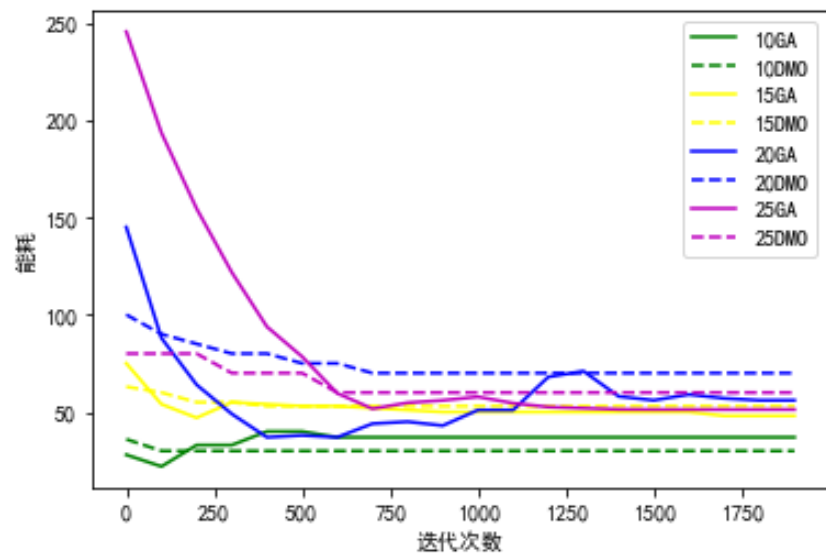
- (1) 随机生成初始解，且保证解的合法性，并计算目标函数值。
- (2) 解两两配对，进行交叉操作。
- (3) 计算完成操作后的解所对应的目标函数，生成图像。



消融实验二实验结果图

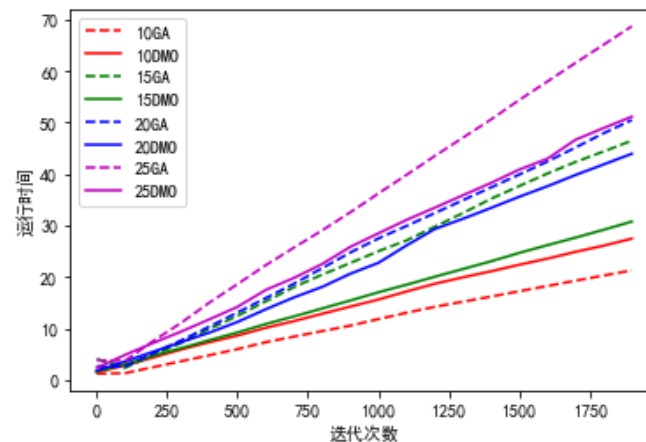
实验评估

尽管侏儒猫鼬算法在最终收敛时可能得到比基因算法差的解，但由于调整策略的优势，它只需进行少数迭代即可获得较好的结果。也就是说，侏儒猫鼬算法具有较强的探索能力（广泛的搜索范围），但在寻找局部最优解方面不如基因算法。



能耗结果图

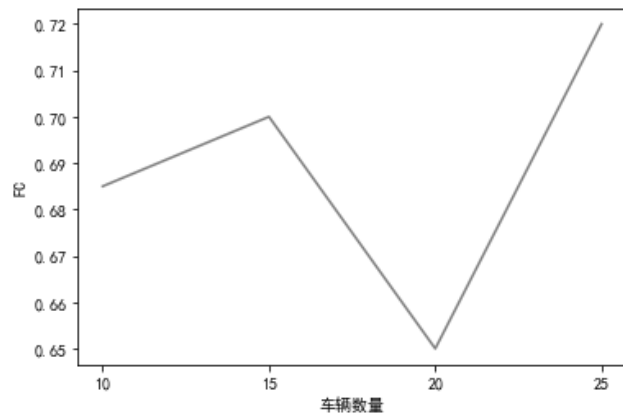
虚线代表基因算法求解问题的时间，而实线则是侏儒猫鼬算法的运行时间。标签前面的数字表示车辆的数量，而同颜色的线代表着相同数量的车辆。侏儒猫鼬算法的运行时间明显比基因算法的运行时间要短。从图中可以清晰地看出侏儒猫鼬算法在解决问题时具有更高的效率和优势。



算法运行时间结果图

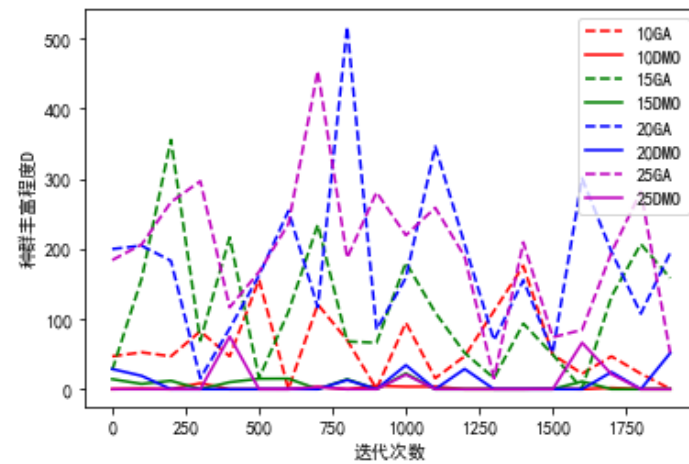
实验评估

数值都在 0.6 至 0.7 之间浮动，车辆任务调度的分配方案相对较为公平。在平衡方面，侏儒猫鼬算法表现良好。



平衡因子 FC 值结果图

在下图中，侏儒猫鼬算法种群多样性程度的变化幅度较大。在算法运行过程中仍在产生新的解的种群多样性程度较大。



种群多样性程度结果图



总结与未来展望

论文的工作及贡献：

1. 研究了车联网中任务卸载的问题模型，将能耗模型分为两部分来描述，将其描述为数学规划的形式。使用了基因算法来求解该问题。
2. 本文在侏儒猫鼬算法的基础上提出了基于贪心法调整的侏儒猫鼬算法，核心思想是通过贪心法（处理时间最短法）以及重定义运算符来实现对种群进行更新。

未来的工作：

1. 考虑加入一些其他的近似算法来加快这个基因算法的过程。
2. 考虑后期动态的不确定性，迭代前期加大不确定性，迭代后期减小不确定性，进一步提高问题的解决效率。

在校期间成果

1. 发表论文

[1] 第一作者, A Genetic Algorithm for Task Offloading problem in Vehicular Edge Computing[C]//2022 China Automation Congress (CAC). IEEE, 2022:6242-6247 (EI)

感谢各位专家的聆听与指导！