

基于并行计算的微观交通仿真模型标定

汇报人：唐揽月

2025年1月9日星期四



同濟大學
TONGJI UNIVERSITY



目录

目录 content



01 绪论



02 仿真搭建与参数选取



03 参数标定与并行计算



04 评价体系与实验对比



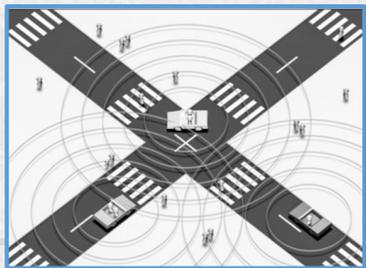
05 总结与展望



一. 绪论

一. 绪论

1.1 研究背景



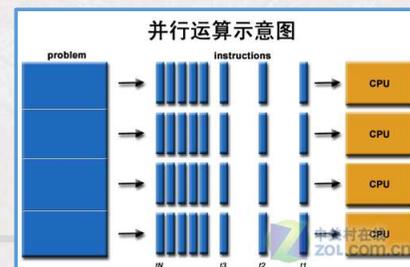
微观交通仿真

- 微观交通流仿真已成为评估各种交通规划管理方案性能的**必要技术支撑工具**。
- 微观仿真工具中的**默认值**往往与特定的仿真场景不匹配，导致**仿真精度较低**。因此，在实际应用中，**参数标定**成为了所有后续工作的前提。



启发式算法

- **启发式算法**因其更广泛的适用性逐渐成为微观交通流仿真参数标定的常用方法。
- 在启发式算法中，一旦参数集中的参数值发生变化，就需要**重新仿真**计算优化后的目标函数。这个过程通常会耗费大量时间，具有**较大的被加速潜力**。



并行计算

- 并行计算的**基本思想**是将一个**大问题分成小任务**，在一个称为并行执行或并行化的进程中，**同时在多个处理器**上解决。
- 目前关于将并行计算应用于仿真标定的研究较少

一. 绪论

1.2 工作内容



01

按照**并行框架选择**、**计算瓶颈识别**和**负载均衡设计**三个步骤，设计并实现**GA和PSO的并行化**，实现微观交通流仿真的参数标定。

建立一个包含高密度交通流的**SUMO仿真模型**，将建立的并行标定算法应用于该案例研究，验证了标定算法在**准确性方面**的性能

02



03

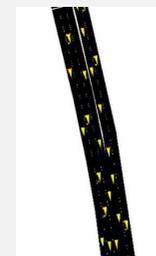
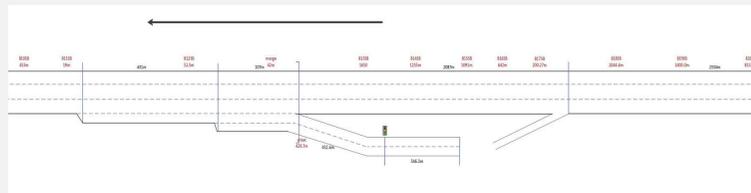
本研究从**标定计算时间**和**可扩展性**两个维度对两种标定算法的并行计算性能进行评估和比较



二. 仿真搭建与参数选取

二，仿真搭建与参数选取

01研究对象概述



本项目以澳大利亚 Bruce 高速公路的 Deception Bay 大街出口匝道 Anzac 大街入口匝道上下游路段作为研究路段。基于实际检测数据，利用 SUMO 仿真搭建5km长的仿真模型，属于典型的**车道减少 (Lane-drop) 常发性瓶颈点**。

仿真场景搭建

流量预估与输入

Sobol敏感性分析

参数选取结果

02交通流状况分析

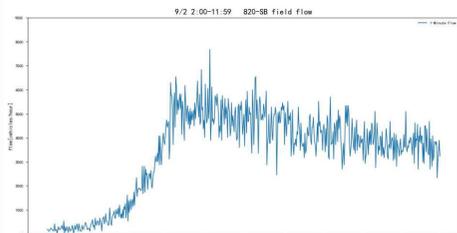


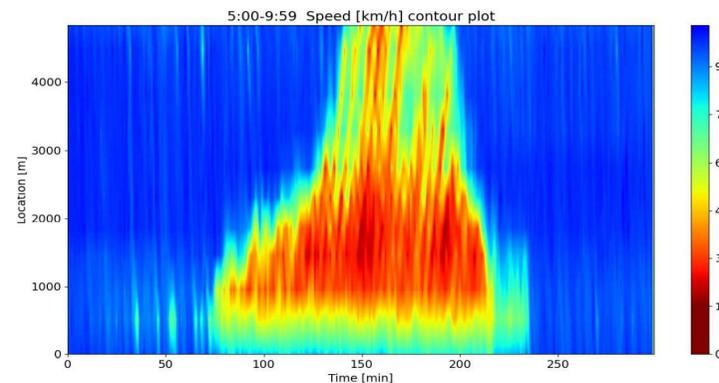
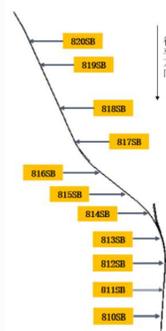
图 2-2 0:00-11:59 820-SB 流量变化图

820流量变化曲线图

- 不同于国内的交通流状况，研究地区的**早高峰开始的较早**，从 **5:00** 左右开始输入的流量开始出现**明显的增加**。到 **9:00** 左右输入的流量出现平缓减少的趋势，但输入流量总体维持在一个**较高的水平**。

速度时空图

- 由速度时空图左右时，入匝道汇入处，即车道减少瓶颈点开始可见 **6:10**出现**明显的拥堵**，并在 **6:40-8:30** 高峰期间，拥堵有自车道减少瓶颈点**向上游路段延伸**的趋势。



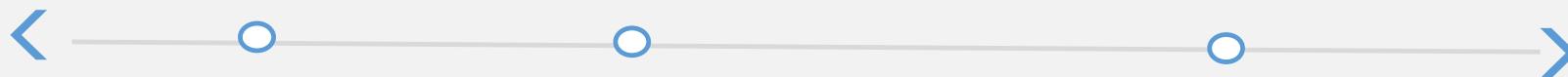
二.仿真搭建与参数选取

Sobol敏感性分析

参数选取结果

01预估流程

由于SUMO的flowrouter.py工具存在一定的问题，本项目中根据高速公路OD推导法，自主编写了route与flow的生成工具，较之于原工具，考虑了**车辆通行时间**。



检测器生成

根据线圈检测器的经纬度生成检测器文件
820-810.add.xml

路径生成

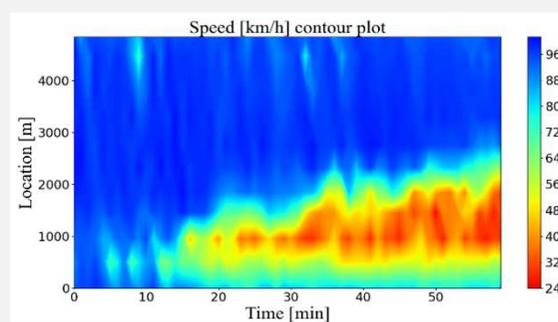
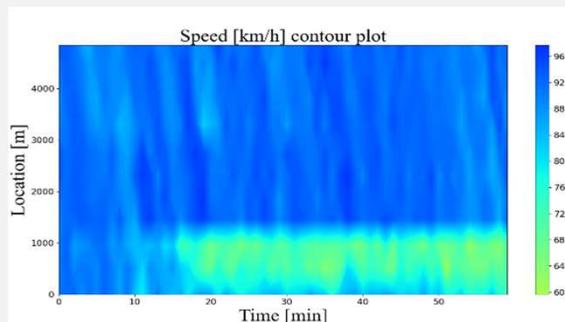
对检测器之间的路径进行选择保留
Route.rou.xml

计算流量

考虑车辆在两相邻检测器之间的行驶时间。利用820-SB、817-SB与816-SB流量差、814-SB与813-SB流量差预估了三条路径的输入流量。

上述流量输入方法仍较为粗糙，无法准确预估拥堵时车辆的流量，在拥堵时的输入流量进行了一定的调整。

02实际速度时空与默认仿真



- 本研究选取了**6:00AM-6:59AM**时间段来进行标定实验，
- 右图为实际线圈速度时空图，左图为默认参数设置下，仿真模型的速度时空图。

二.仿真搭建与参数选取

仿真场景搭建

流量赋值与输入

参数选取结果

01参数选择及范围

参数敏感性分析的过程中，
确定以下参数进行敏感性分析：

- IDM跟驰模型参数
- LC2013换道模型参数
- 2个速度分布参数 (SpeedDev,SpeedFactor)

规则	参数取值范围选取
1	符合sumo官方文件取值范围
2	考虑与其他参数的限制关系
3	参考以往sumo/vissim标定研究中意思相近的参数取值范围
4	参考sumo中给定的默认值大小，使范围尽量在默认值周围

02Sobol敏感性分析

本项目采取较为通用的**基于方差的敏感性分析法**，具体采用**sobol**敏感性分析法。该方法的基本思想是**基于方差可以很好地代表输出中的不确定性的假设**。

Sobol序列

Sobol 序列本质上是以最小素数 2 为基数的确定性序列，由有限领域线性递推关系构建的。

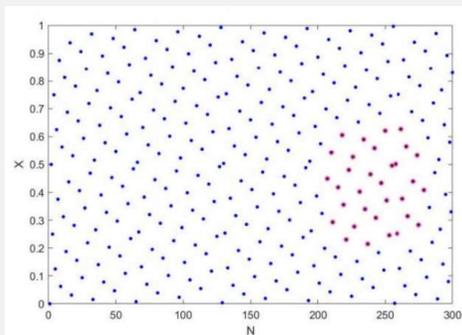


图 2-7 Sobol 序列采样

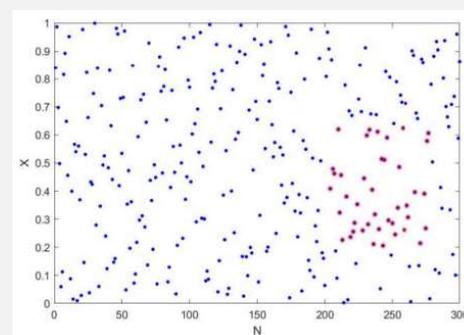


图 2-8 LHS 采样

由图可以看出 Sobol 采样产生的样本点在**空间上分布更加均匀，覆盖范围广**，产生的采样点多样性效果更好。

二.仿真搭建与参数选取

仿真场景搭建

流量预估与输入

Sobol敏感性分析

参数选取结果

Sobol敏感性分析流程

sobol序列取值

样本生成

运行仿真

计算敏感性指标

如N=4, D=3, 参数[0, 1], 在区间内取4个值
[0.25, 0.5, 0.75, 0.875]

N=20, D=18

$$A = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.75 & 0.25 & 0.25 \\ 0.25 & 0.75 & 0.75 \\ 0.375 & 0.375 & 0.625 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.25 & 0.75 & 0.75 \\ 0.75 & 0.25 & 0.25 \\ 0.875 & 0.375 & 0.125 \end{bmatrix}$$

+

$$AB^1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.25 & 0.25 & 0.25 \\ 0.75 & 0.75 & 0.75 \\ 0.875 & 0.375 & 0.625 \end{bmatrix}$$

$$AB^2 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.75 & 0.75 & 0.25 \\ 0.25 & 0.25 & 0.75 \\ 0.375 & 0.375 & 0.625 \end{bmatrix}$$

$$AB^3 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.75 & 0.25 & 0.75 \\ 0.25 & 0.75 & 0.25 \\ 0.375 & 0.375 & 0.125 \end{bmatrix}$$

生成共N* (D+2)
400个参数组合样本

$$S_i = \frac{Var_{X_i}(E_{X_{\sim i}}(Y|X_i))}{Var(Y)}$$

$$S_{Ti} = \frac{E_{X_{\sim i}}(Var_{X_i}(Y|X_{\sim i}))}{Var(Y)}$$

speed均方根误差

Sobol敏感性分析结果参考值

- 根据参考文献, 当全局敏感性指数ST大于2%时, 可以考虑选择为调整的参数。结合实验数据, 本项目中选择将所有参数的ST进行排序, 前7选择为要调整的参数集。

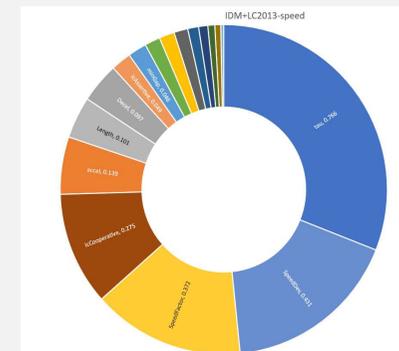
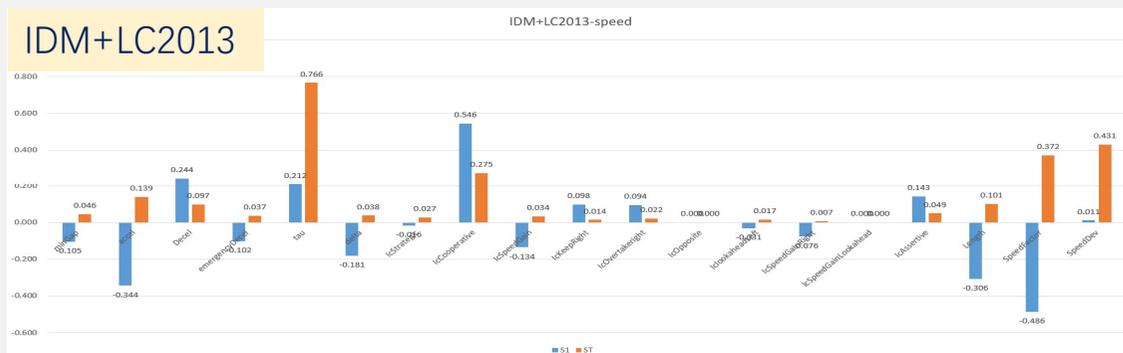
二.仿真搭建与参数选取

仿真场景搭建

流量预估与输入

Sobel敏感性分析

03敏感性分析结果



参数名	参数含义	ST	取值范围
tau	驾驶员期望的（最小）行车间隔时间。	0.766	[1, 4]
SpeedDev	定义为车辆速度的偏差值。	0.431	[0, 1]
SpeedFactor	定义为车辆速度期望值。	0.372	[2, 9]
lcCooperative	合作换道的意愿。较低的值导致合作减少。	0.275	[0, 0.5]
accel	各类车辆的加速能力(m/s ²)	0.139	[0.5, 1.5]
Decel	各类车辆的减速能力(m/s ²)	0.097	[0.8, 1.5]
lcAssertive	愿意接受目标车道上较低的前后间距。所需间隙除此值	0.049	[1, 7]



三.参数标定与并行计算

三.参数标定与并行计算

遗传算法
参数标定

并行计算

最终标定
算法框架

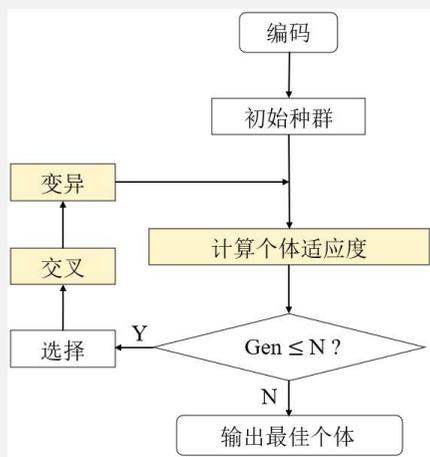
01问题建模

微观交通流仿真的参数标定本质上来说是一个**带有目标函数的组合优化问题**，本项目中采用**速度的平均绝对归一化误差**以及**反映瓶颈范围匹配程度的指标C1**做为参数标定的目标函数。

$$\text{最小化 } RMSE_{\text{speed}} = \sqrt{\frac{\sum_{i=1}^N (d\text{speed}_i - f\text{speed}_i)^2}{N}}$$

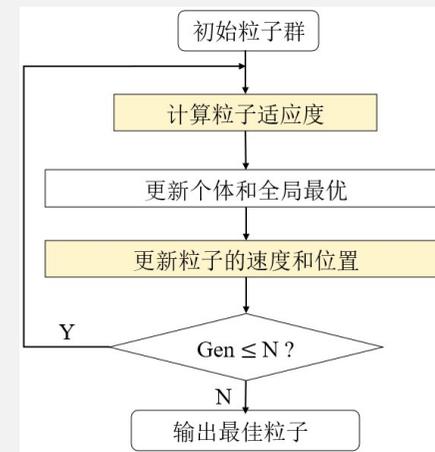
$$\text{最大化 } C_1 = \frac{2 \sum_{i=1}^N \left\{ \left(\sum_{t=1}^T [BS_S(i,t) \wedge BS_r(i,t)] \right) \cdot (x_{i+} - x_i) \right\}}{\sum_{i=1}^N \left\{ \left(\sum_{t=1}^T [BS_S(i,t) \vee BS_r(i,t)] \right) \cdot (x_{i+} - x_i) \right\}}$$

为防止**车道间的速度差对 RMSE 的值产生影响**，故本项目中的对仿真输出的速度以及实际场地速度均采用了**流量加权平均**。



- 就更新方法而言，粒子群优化算法比遗传算法更简洁。
- 而遗传算法较之于粒子群优化算法能更有效地避免局部最优陷阱。

02遗传算法与粒子群优化

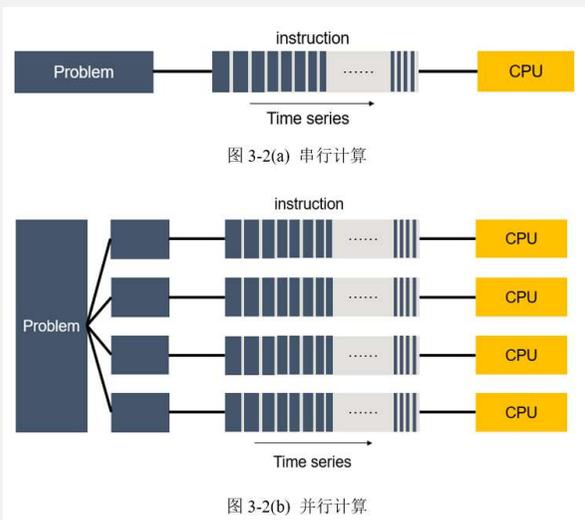


三.参数标定与并行计算

遗传算法
参数标定

最终标定
算法框架

01概述



- 并行计算是指同时利用**多个计算资源**执行**多个进程或多个计算的算法**。
- 从**计算资源供给**的角度出发，并行计算技术的应用意味着可以**更好的利用**计算机已有的**硬件资源**。
- 从**需求**角度出发，并行计算从最本质的建模逻辑上来说更适用于对现实世界的**复杂现象**进行建模、模拟。
- 总体来说，并行计算技术起到**节约问题计算的时间和成本**的作用。

并行计算算法应用条件:

该问题能够被**分解为离散片段**，且这些离散片段需要是能够被**并发执行的**



该问题分解得到的不同离散片段在**执行顺序没有要求**，即是**互相独立**的片段



要求采用多个计算资源的花费时间需要小于单个计算资源所花费的时间。



02问题适应性分析

- 遗传算法：计算个体适应度、交叉、变异。
- 粒子群优化：计算粒子适应度，更新速度与位置。

三.参数标定与并行计算

遗传算法
参数标定

并行计算

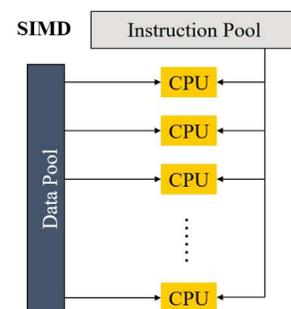
最终标定
算法框架

03算法设计

本项目使用**单指令多数据 (SIMD)** 架构:

- 首先**主进程**对数组进行初始化, 生成初始化参数集种群,
- 将各个参数集信息**发送给子进程**, **并接收**子进程运行仿真计算收敛指标的**结果**。
- 子进程在收到信息后, 分别执行仿真计算任务, 并将最终的计算结果发送给主进程

并行计算框架



算法瓶颈识别



控制两者评价次数相同

- x 轴表示抽样数, 一个函数在 **x 轴占据的宽度越宽**, 表征其执行的时间越长。
- 由此图可见, 两个算法中都是仿真运行、计算适应度这一步占用了整个标定过程中大多数的计算资源, 且极为耗时。
- 故本项目将并行计算技术主要应用于**仿真运行适应度**计算这一部分的离散片段

三.参数标定与并行计算

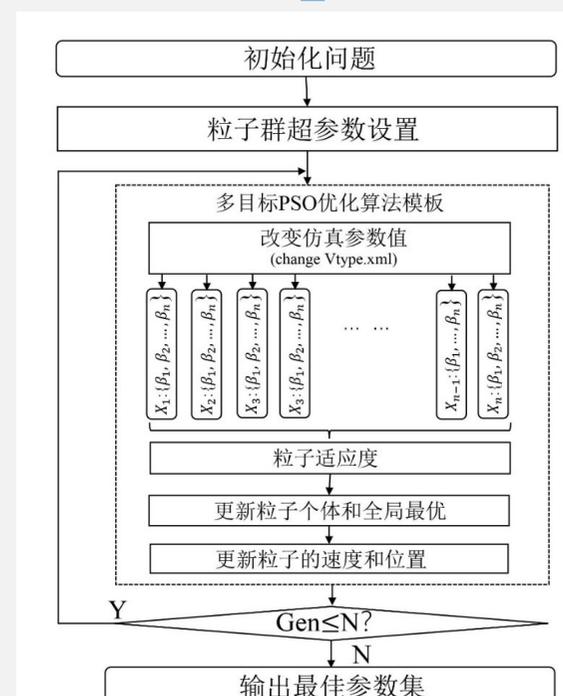
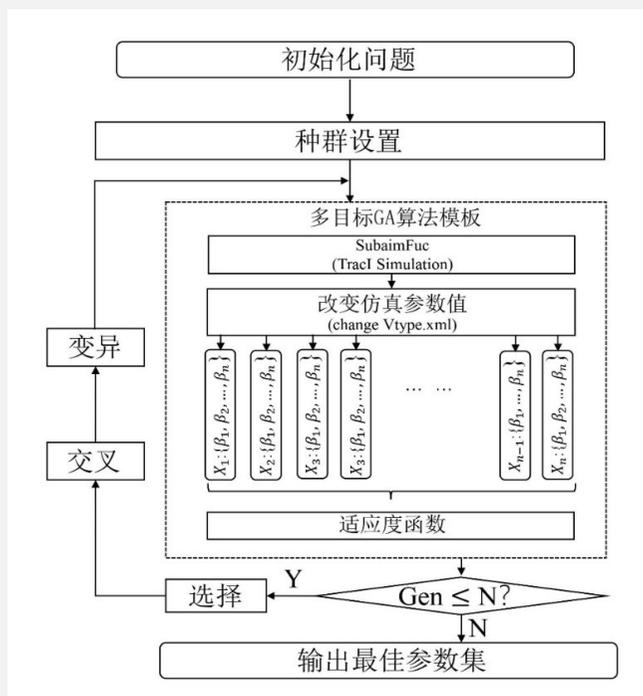
遗传算法
参数标定

并行计算

负载均衡

- **负载均衡**是指在各个子进程分配大约相等数量的计算工作，以便所有处理器在所有时间保持繁忙，使所有子任务空闲时间最小化。
- 故本项目中利用调度任务池（Scheduler-task pool）的方法实现任务的动态分配

最终算法框架





四.评价体系与实验对比

四.评价体系与实验对比



1

仿真精度
评价体系
对比不同跟
驰模型的仿
真适应性

速度流量匹配

$$(1) \text{ speed: } \sqrt{\frac{\sum_{n=1}^{120} (d \cdot \text{speed} \times 3.6 - f \cdot \text{speed})^2}{120}}$$

$$(2) \text{ flow: } \sqrt{\frac{\sum_{n=1}^{120} (d \cdot \text{flow} - f \cdot \text{flow} / 60)^2}{120}}$$

瓶颈范围匹配

二进制速度时空图

其中:
 $S(i, t)$: t时刻检测器 i 处的速度加权平均值,
 $\text{if } S(i, t) \leq V_{th}, BS(i, t) = 1, \text{ else } BS(i, t) = 0$

C1

$$C1 = \frac{2 \sum_{i=1}^N \{(\sum_{t=1}^T [BS_S(i, t) \wedge BS_r(i, t)]) \cdot (x_{i+} - x_i)\}}{\sum_{i=1}^N \{(\sum_{t=1}^T [BS_S(i, t) \vee BS_r(i, t)]) \cdot (x_{i+} - x_i)\}}$$

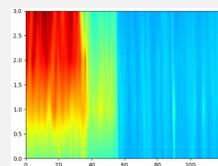
C1=1 则意味着实际与仿真结果的瓶颈区域完全匹配

C2

$$C2 = 1 - \frac{2 \sum_{i=1}^N \{(\sum_{t=1}^T [BS_S(i, t) \wedge BS_r(i, t)] \cdot |S_S(i, t) - S_r(i, t)|) \cdot (x_{i+} - x_i)\}}{\sum_{i=1}^N \{(\sum_{t=1}^T [BS_S(i, t) \wedge BS_r(i, t)]) \cdot (x_{i+} - x_i)\}}$$

C2=1意味着实际与仿真的瓶颈区域和速度值完全一致

速度时空图



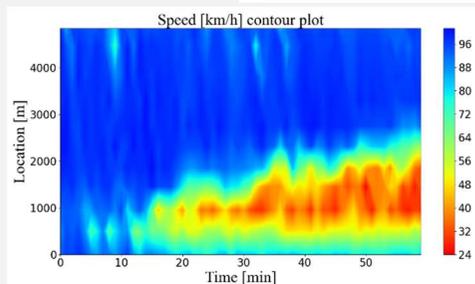
四.评价体系与实验对比

精度评价体系

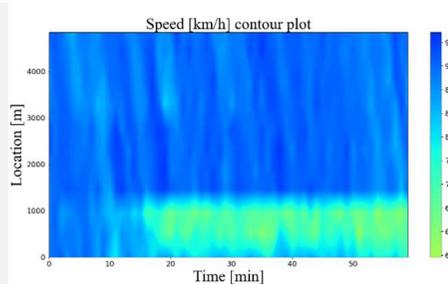
标定精度

在并行GA标定算法中，本文将种群大小设置为32，最大迭代次数设置为30。在并行PSO标定算法中，粒子群数量设置为32，最大迭代次数也设置为30。基于建立的精度评价体系，评价算法的标定结果。

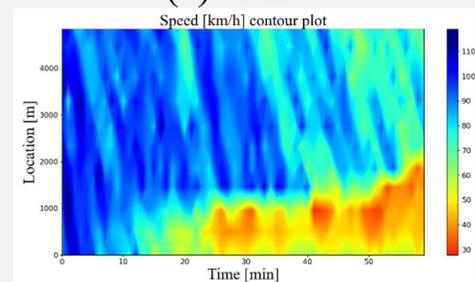
Indicator	$RMSE_{speed}$	$RMSE_{flow}$	C_1	C_2
Default	27.71	30.35	0	0
Parallel GA	16.70	21.10	0.74	0.62
Parallel PSO	19.80	25.04	0.78	0.60



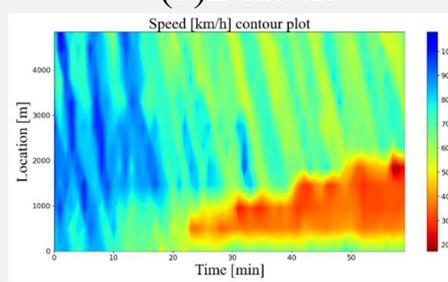
(a)Field



(b)Default



(c)Result of parallel GA



(d)Result of parallel PSO

标定结果

- 在SUMO仿真平台中使用默认值时，仿真模型**根本无法模拟**实际交通流中的瓶颈状态。
- 标定后，模型的仿真精度大大提高，实验结果验证了本文提出的并行标定算法的**有效性**。

效率评价体系

效率实验结果

四.评价体系与实验对比



效率量化
评价体系
并行计算应用的有效性

IDM+LC2013

仿真执行时间

比较串行, 与n个pool池的运算时间

加速度比指标

加速比的计算方法为单处理器上的最优串行化算法计算时间与使用 n 个处理器并行计算时间的比。

并行效率

并行效率计算方法如下:

$$E_N = \frac{S_N}{N}$$

N: 并行计算程序使用的处理器个数。

S_N : 当使用 N 个处理器时, 并行计算程序的加速度比。

并行效率受**程序可并行化比例、硬件**, 包括**网络通讯质量和速度、应用算法、通讯开销**以及**解决的具体问题特征**等因素复杂的影响。

可拓展性

可拓展性是指整个并行算法**通过增加更多的计算资源成比例增加并行速度的能力**, 是并行算法在多大程度上能够有效利用多处理器数量增加的能力的一个度量。

随着并程序处理器的增加, 若**并行效率曲线基本保持不变或略有下降**, 则认为该并行算法的**可拓展性良好**, 反之, 若效率曲线**下降的很快**, 则认为该并行算法**可拓展性较差**。

四.评价体系与实验对比

精度评价体系

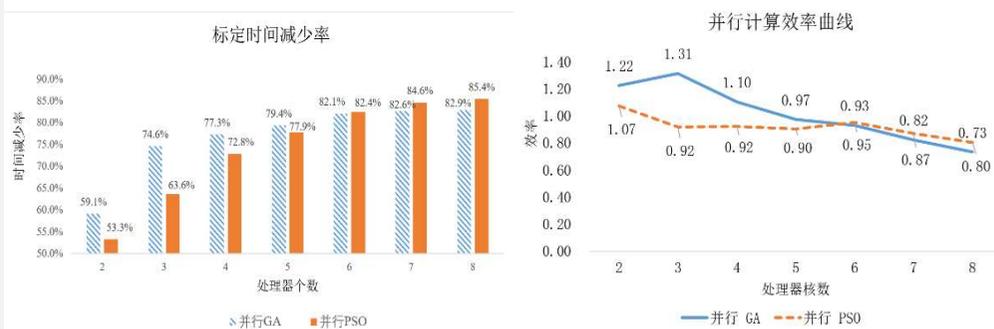
标定精度

效率评价体系

效率实验结果

本文将两类标定算法的总仿真评价次数控制为960次。

Calibration algorithm	Parallel GA			Parallel PSO		
	Number of processors	Calibration time (h)	Time reduction ratio (%)	Calibration time (h)	Time reduction ratio (%)	Acceleration ratio
Serial		5.65	-	4.84	-	-
2		2.31	59.1%	2.26	53.3%	2.14
3		1.43	74.6%	1.76	63.6%	2.75
4		1.28	77.3%	1.32	72.8%	3.68
5		1.16	79.4%	1.07	77.9%	4.52
6		1.01	82.1%	0.85	82.4%	5.69
7		0.98	82.6%	0.80	84.6%	6.09
8		0.97	82.9%	0.76	85.4%	6.42



- 并行计算的应用大大缩短了标定时间。在单线程串行标定算法中，标定需要**5.65小时**。采用并行计算技术后，标定时间能够缩短到**1小时以内**，**降低了80%**。实验结果验证了PCT在标定问题中应用的重要意义。
- 从两种算法的效率曲线可以看出，并行PSO算法的**并行效率曲线**比较**平缓**。这表明其可拓展性更好，使用增加的**计算资源的能力更强**。



五. 结论与展望

五. 结论与展望

主要结论

1

- 本文应用**PCT**来优化计算瓶颈。本文按照**并行框架选择**、**算法瓶颈识别**和**子任务负载均衡设计**三个步骤，实现了GA和PSO标定算法的**并行化**。

2

- 本文验证了将并行计算技术应用于参数标定可以大大加快标定速度，在实例研究中加速效果达到了**80%**以上。

3

- 研究结果并行算法利用增加的资源的能力，即可**拓展性**，是评价并行算法的一个**重要而必要的指标**。从结果可以看出，并行PSO的可扩展性优于并行GA。

审稿意见总结

Review 1

- 并行计算并行计算技术是否可以结合到现有的微观仿真软件中去？
- 建议进一步研究在不同大陆的不同交通场景下的实验，因为跟车、换道、速度分布参数会有很大差别。

Review 3

- 仿真模拟的过程最为耗时是交通领域的常识。计算瓶颈识别的讨论和图表应该更加简洁明了，火焰图处叙述有点冗长。
- 将并行GA与并行PSO标定后的与SUMO默认值输出的仿真模型相比较不合理。与别的不依赖并行的标定算法的结果相比较才比较合理

五. 结论与展望

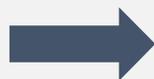
Review 1

- 并行计算并行计算技术是否可以结合到现有的微观仿真软件中去？
- 建议进一步研究在不同大陆的不同交通场景下的实验，因为跟车、换道、速度分布参数会有很大差别。

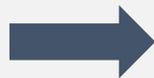
Review 3

- 仿真模拟的过程最为耗时是交通领域的常识。计算瓶颈识别的讨论和图表应该更加简洁明了，火焰图处叙述有点冗长。
- 将并行GA与并行PSO标定后的与SUMO默认值输出的仿真模型相比较不合理。与别的不依赖并行的标定算法的结果相比较才比较合理
- 涉及很多计算机名词，缺少解释。

后续改进计划



- 在VISSIM中实现一个简单应用，实现VISSIM中PSO标定算法的并行化，表明并行计算技术的可移植性。



- 写入研究展望。



- 删减这一部分的冗长内容，只展示图表



- 感觉评论者误会了，并行计算对于精度没有影响，只对速度有影响。文中比较精度只是为了说明展示一下标定的结果。



- 补充相关概念的解释

谢谢!

汇报人：唐揽月

2025年1月9日星期四



同濟大學
TONGJI UNIVERSITY

