实 验 报 告

实验名称：基于 DSOL 的仿真轮船货运模型实现

一、实验目的与任务

目的：理解并掌握课堂所学知识，并将其运用到实际中

任务：基于 DSOL 平台设计一套物流系统并分析实验结果

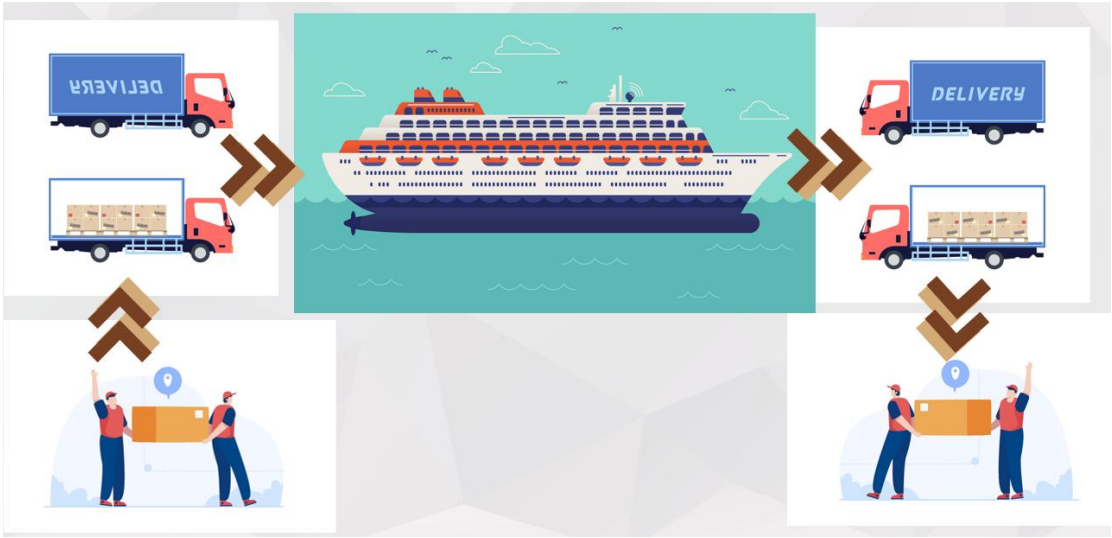
二、实验原理

基于 DSOL 平台建立仿真模型，使用 Java 编程语言构造不同的原子模型类，创建不同对象的 class，完成整个耦合模型的构建。

三、实验内容及要求

基于 DSOL 平台实现一个仿真模型。

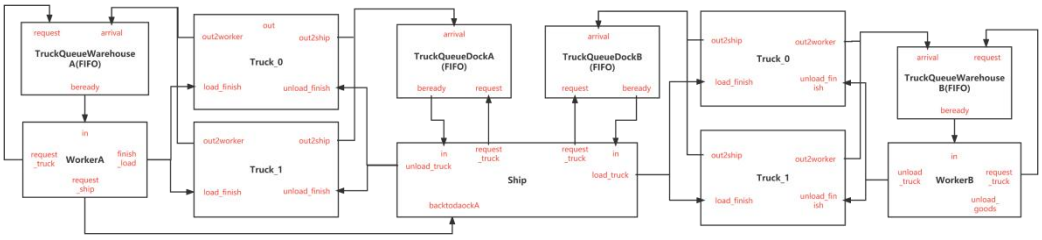
四、想定场景和研究问题



如图，A 地与 B 地相隔一条河，在 A 地仓库的货物需要运往 B 地的仓库，模型主要解决的是货物在两地之间的运输问题，A 地工人先将货物运到 A 地卡车，卡车前往轮船，由轮船携带货物前往 B 地码头，到达码头后将货物装载到 B 地卡车上，B 地卡车携带货物前往 B 地仓库，由 B 地仓库工人卸载到 B 地仓库。

五、模型设计

模型架构如下图所示。



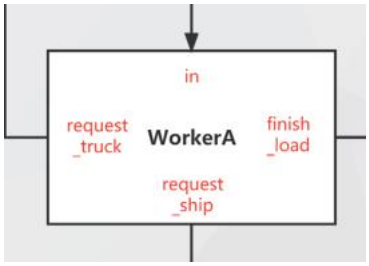
模型对象分析：

基础对象类：

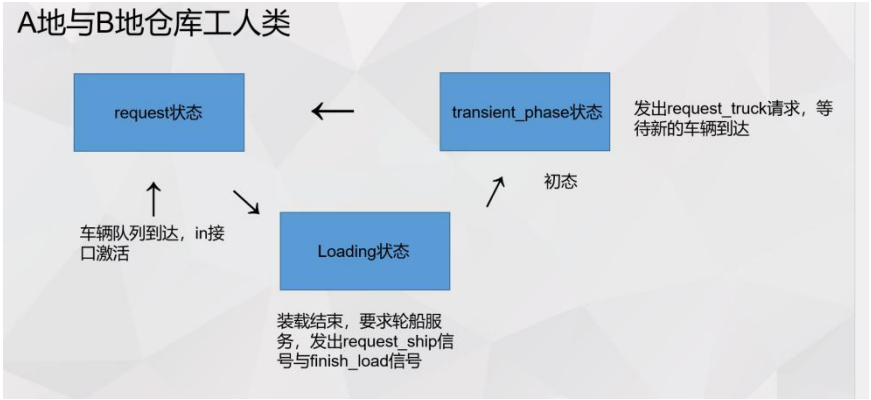
1.仓库工人类：仓库工人的行为主要是装载与卸载货物。

在 A 地的仓库工人装载货物的工作需要 A 地空卡车的到达；在 B 地的仓库工人负责卸载货物，其行为受限于 B 地的卡车

原子模型：



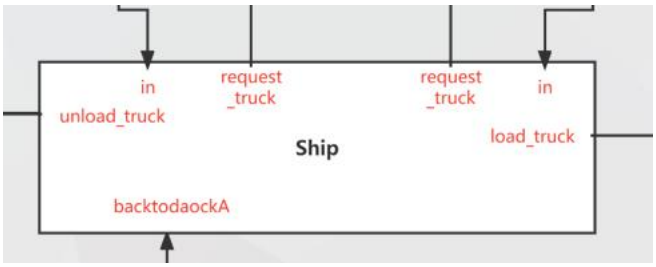
状态分析：



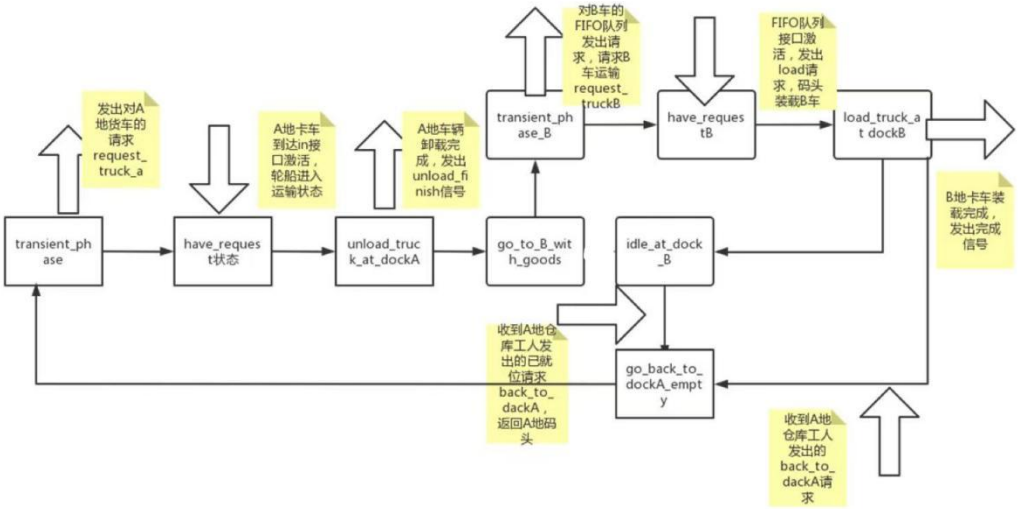
仓库工人有三个状态，分别是 request 状态、Loading 状态和 transient_pha 的暂态。仓库工人在开始时处于 transient_phase 状态，发出对 truck 的 request_truck 请求，进入 request 状态，等待卡车到达。卡车到达后,in 接口激活，工人状态转变为 Loading 状态，，经过一定时长后，装载完成，对轮船发出 request_ship 信号，对指定卡车发出 finish_load 信号。

2.轮船类：轮船负责在 A 地和 B 地的运输活动，设计相对而言是耦合模型中比较复杂的一个原子模型。轮船需要到达 A 地收到 A 地卡车运送的货物，然后送往 B 地，同时需要唤醒 B 地的卡车，进行装载。

原子模型：



状态分析：

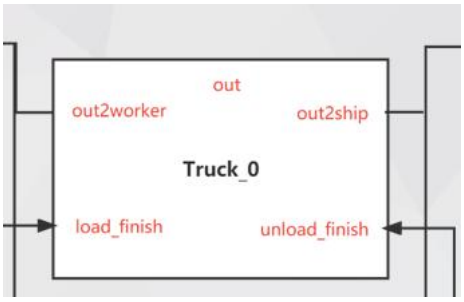


轮船初始状态为 transient_phase 状态，发出 request_truck 信号，进入 have_request 状态，A 地卡车到达后，轮船卸载卡车，进入 unload_truck_at_dockA 状态，卸载卡车完成后，对指定卡车发出 unload_truck 信号，之后开始进行运输，进入 go_to_B_with_goods 状态，到达 B 地后，轮船进入 transient_phaseB 状态，发出对 B 地车辆的请求信号 request_truck1，进入 have_requestB 状态，等到 B 地卡车到来后，右侧的 in 接口激活，码头开始装载卡车，进入 load_truck_at_dockB 状态，等 B 地卡车装载完成后，轮船发出 finish_load 信号，此时需要查看之前是否有 A 地仓库工人的轮船请求，如果有的话。轮船进入 go_back_to_dackA_empty 状态，返回 A 地，返回后进入 transient_phaase 状态，要求 A 地卡车装载完成后到来，如果装载完 B 地卡车后，没有收到 A 地的货运请求，则进入空闲态 idle_at_dockB 状态。

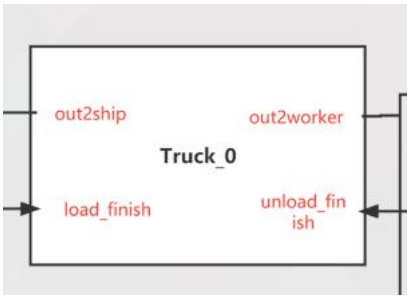
3.卡车类：　卡车类有两部分，从 A 地仓库到达 A 地码头的卡车类与从 B 地码头到 B 地仓库的卡车类，二者的主要区别是初态时 A 地的卡车需要自动运行到达 A 地仓库，唤醒 A 地仓库工人进行装载，是整个模型先开始运行的部分；B 地卡车需要轮船先运送货物再装载，之后运送到 B 地仓库卸载

原子模型：

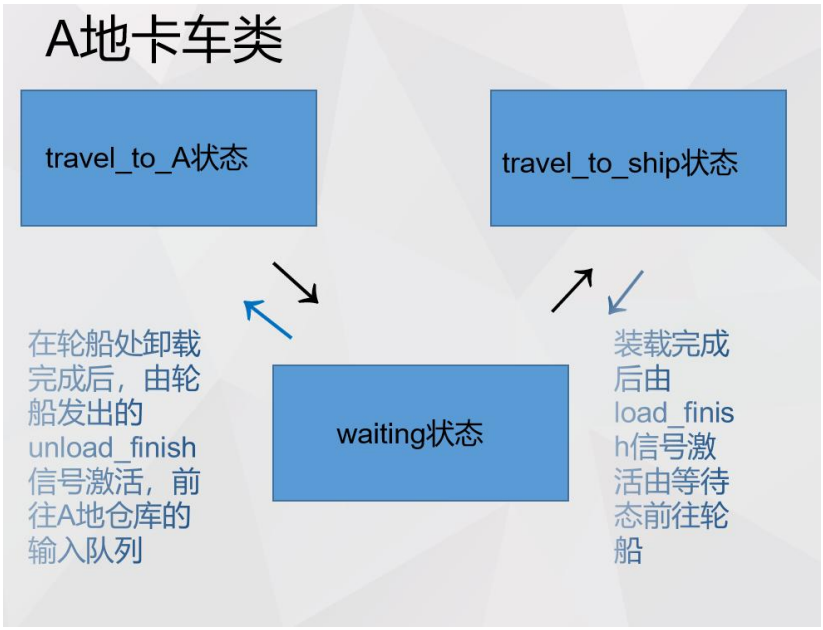
A 地卡车：



B 地卡车：



状态分析：

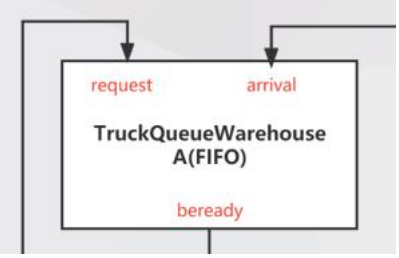


两地卡车共三个状态，分别是 travel_to_A/B、travel_to_ship 状态和 waiting 状态，状态转换如图所示，主要取决于卡车是否装载或者卸载完成，需要船或者仓库工人发出相应的信号激活。

补充对象类：

考虑到车辆不止一辆，不能与轮船或者仓库同时交互，同时车辆的交互顺序符合 FIFO 队列的排队过程，所以在两个基本对象类的交互过程中需要设置 FIFO 对象类，通过队列模型完成两个基本对象类的一对一的交互过程

FIFO 队列类：遵循先来先服务原则，先来的车辆先得到服务。



状态分析：FIFO 队列采用了三个接口 request、arrival、beready，request 接口决定了服务对象，arrival 决定了卡车的到达顺序，同时根据到达顺序的先后输出 beready 接口，采用队列的形式使卡车的输出顺序由 arrival 的顺序决定，当 request 与 arrival 同时被激活时，就输出队列头部的元素，即先到达的卡车，同时记录卡车序号。

六、仿真实现

主要是对 Java 类的定义，构建相应的原子模型，需要考虑输入接口激活之后的状态转换，输出接口的发出条件以及模型内部之间的状态转换。搭建完原子模型之后可以构造出相应的耦合模型。

七、实验运行及结果分析

1.仿真结果输出

仿真进程中各个对象存在因内部转移函数、外部转移函数导致的状态转移，以及对外部的输出函数，对产生这些动作的仿真器实时时间和存在时长进行输出，据此表现各个对象对应的实体的行为的发生时间和持续时间，同时，针对队列类，将队列发生变化时队列中存在的对象输出，如图所示 7-1 所示，仿真时间为 0 时，初始化对象，并根据状态转移，出现卡车到达队列等待和工人向队列发出请求事件，根据 FIFO 队列先进先出，即先进入队列先接受服务的设定，此时刻 Truck_a_1、Truck_a_0 和 Truck_b_1、Truck_b_0 分别先后进入 TruckQueueLaborLoad 和 TruckQueueAtDockB 等待，表示 A 地卡车排队等待在仓库装载货物，B 地卡车在港口等待船到达后卸下货物并装载上车。同时，Ship 向 TruckQueueAtDockA 发出请求，请求一辆满载的卡车到港口卸货上船，Labor_a 和 Labor_b 分别向 TruckQueueLoad 和 TruckQueueUnload 发出请求，请求卡车前来装货或者卸货。TruckQueueLoad 队列中有等待卡车且有来自工人的请求，因此队列中的卡车开始接受服务，产生服务时间，经过服务时间后，卡车、工人和队列的状态发生转移。据此说明仿真初始化运行不存在问题。

图 7- 1 仿真输出结果

2.实验数据记录

思考后我们认为可以通过对卡车等待时间进行分析，从而得出一些影响等待时间的因素，可以通过分析这些因素得出是否可以优化运输效率的结果，因此，此次实验将对卡车进入等待状态的仿真时间和等待时长进行记录并分析，通过建立并使用 FileWrite 类将得到的实验数据写入文档，以便后续在 matlab 中进行分析，实验结果记录示例如图 7-2 所示。

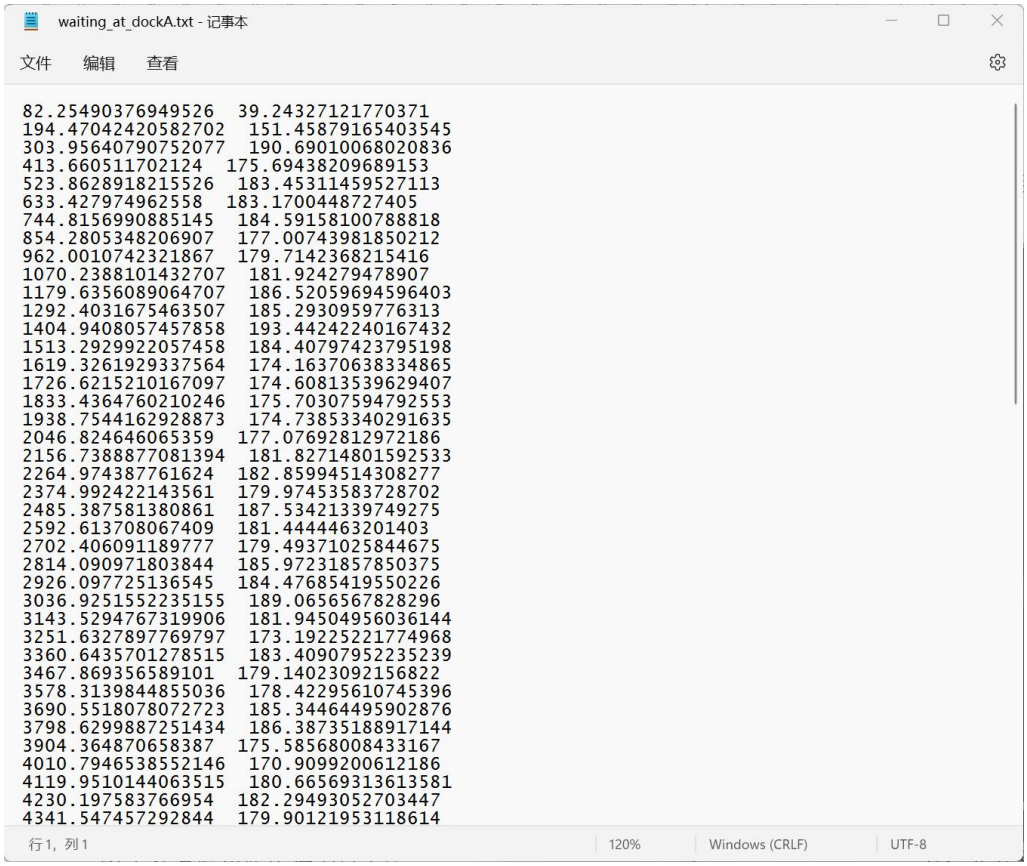
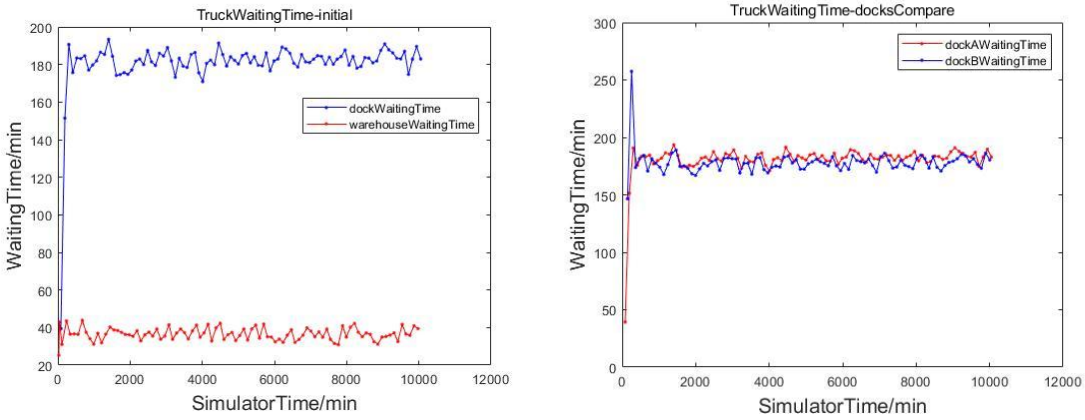


图 7- 2 实验数据记录

3.实验数据分析

要对卡车等待时间进行分析，首先要判断记录的卡车等待时间是否平稳，即数据波动是否较小，在 matlab 中对实验记录的随仿真时间推进的卡车等待时间进行可视化，如图 7-3 所示。图 7-3(a)展示的是 A 地港口与仓库处的卡车等待时间，图 7-3(b)展示的是 A、B 两地港口处的卡车等待时间，可以看出除了仿真初期纵坐标有较大跃迁（由于第一辆卡车刚入队列就可以接受服务就不存在等待时间），随着仿真时间的推进，各处的卡车等待时间都是趋于平稳的，并无较大波动，因此可以在一定程度下对其影响因素进行分析。



(a) A 地港口与仓库卡车等待时间　　(b) A、B 两地港口卡车等待时间

图 7-3 初始条件下的卡车等待时间

本次实验对卡车在港口的等待时间进行分析，由于在模型设计时，对 A 地仓库工人的装货时间和 B 地仓库工人的卸货时间采用了相同的时间分布，故由 7-3(b)可以看出这种情况下 A、B 两地港口的卡车等待时间是差异极小的。

接下来改变 A、B 两地的卡车数量，即等同于改变 A、B 两地的等待队列长度，如图 7-4 所示，红线代表初始状态，即 A、B 两地卡车数量均为 2 时 A 港口的卡车等待时间，黄线代表 A 地卡车 2 辆、B 地卡车 4 辆时 A 港口的卡车等待时间，蓝线代表 A 地卡车 4 辆、B 地卡车 2 辆时 A 港口的卡车等待时间。可以看出只有 A 地卡车数量改变才能使 A 港口的卡车等待时间改变，考虑队列长度增加而服务台数量不变等待时间变长是理所当然的，进而考虑改变仓库装货时间观察其变化。

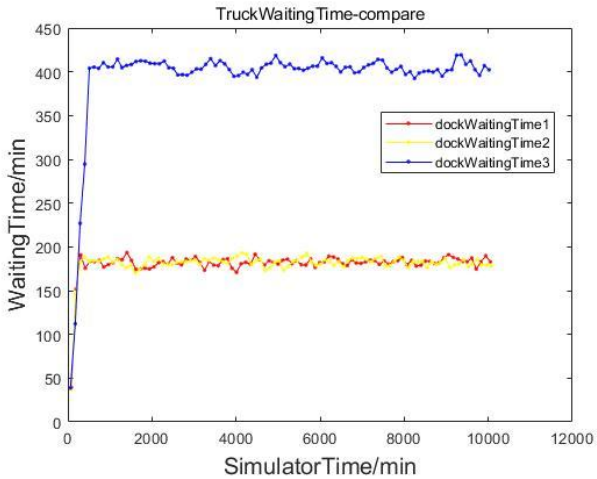


图 7-4 改变 A、B 两地卡车数量后 A 港口的卡车等待时间

改变 A 地工人在仓库时的装载时间分布，对时间分布区间端点值依次增加 10 分钟，得到的卡车等待时间分布如图 7-5 所示，蓝线代表初始状态，红线、紫线、黑线分别代表增加 10 分钟、20 分钟、30 分钟的卡车等待时间分布，可以看出卡车等待时间还是有明显降低的。接下来进行方差分析，检验装载时间对卡车等待时间的影响是否显著。

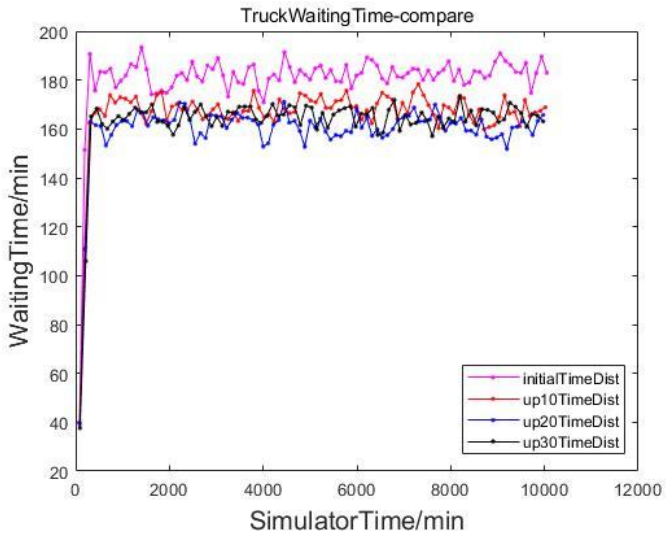


图 7- 5 不同装货时间分布下的卡车等待时间

方差分析的结果如表 7-1 所示，得到最终的检验值 $F = 6.007$，通过查表得到的 F 分布在显著性水平 $\alpha = 0.001$ 下的值为$F_{(3,60)} = 6.17$，$F_{(3,120)} = 5{,}79$，得到$F_{(3,120)} < F = 6.007 < F_{(3,60)}$，可以说明在显著性水平 $\alpha = 0.001$ 下装载时间对卡车等待时间有显著性影响，即说明相对情况下还是有一定影响。

方差来源	离差平方和	自由度	均方值	F 值
组间	$SS_A = 2.130$	3	$MS_A = 7.100$	$F = \frac{MS_A}{MS_E} = 6.007$
误差	$SS_E = 1.038$	81	$MS_E = 1.182$	
总和	$SS_T = 3.168$	84		

表 7- 1 ANOVA 表

本次实验经过分析，得到在本次实验模型设计的条件下，装载时间是卡车等待时间的一个影响因素，后期可以优化仿真模型，加入更多扰动，考虑更多影响因素，分析后可以建立预测模型，估计出参数，从而能够解决更多问题。

八、实验心得

1.模块化思想

解决一个复杂问题时，可以自顶向下把系统划分成若干模块。把完整的系统分割成可组合、分解和更换的单元。构建一个耦合模型时，可以把模型划分为各种原子模型。

2.对原子模型的构建

构建原子模型时，需要考虑模型的通用性，以保证相同模型可以多次使用，比如 FIFO 队列与 truck 类，在本模型中均具有一定的通用性。对于比较复杂的原子模型，需要仔细考虑其中的状态转化，符合模型内部的逻辑关系。

3.面向对象编程的训练

在编程过程中，可以体会到原子模型的封装性，比如对仿真器的使用，我们定义的主要是 input 与 output，以及原子模型中的状态转换条件的梳理，从而使原子模型得到运行，同时对于外部其他的原子模型来说，只了解模型接口，而不清楚其他类的实现过程。

4.关于 DSOL

DSOL 的分布式指的是一种拆分和聚合的思想，将一个复杂的仿真模型拆成易于实现的小部分，这使建模简单又省时。其次 DSOL 创建的模型中可以同时共存多种模拟形式，比如离散事件模拟，基于代理的模型和原子 DEVS，这使它适用范围更加广泛。