**1. 我做到了3顆星，加Dashboard跟GUI**

**2. how to build my model**

首先建立一個Globaltable，叫demand\_location跟server\_location來計錄村落地址、醫院地址、跟各個lambda。

各拉一個Source、Queue、Processor、Sink、Operator。

在Source內的triger寫一個script 使每個由Source(NUM)出來的物件都叫做(NUM)，此用意是未來方便做判斷。Source的expontential輸入連線時再弄。

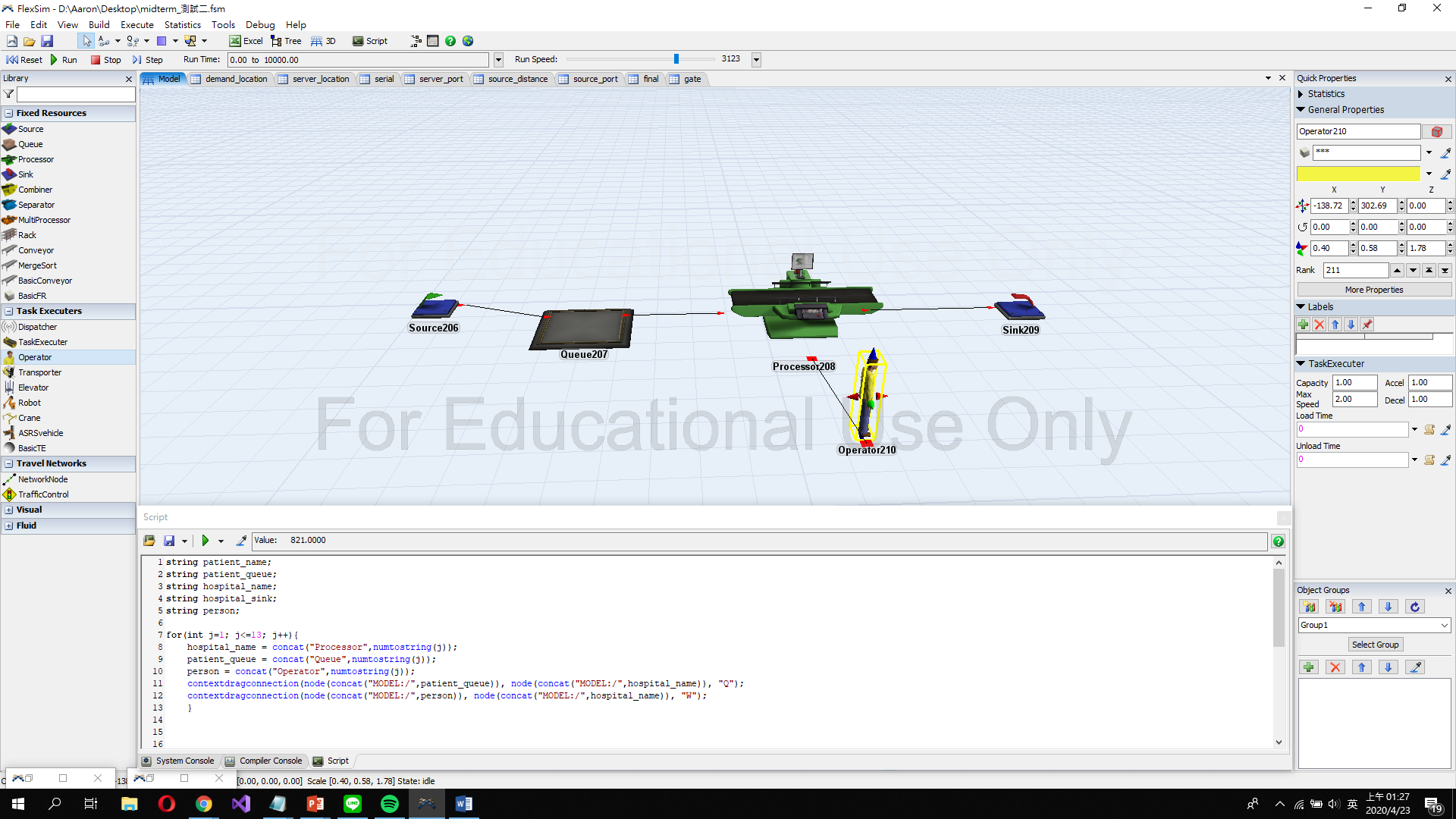
再來是要用最短路徑的設定，把Source的flow按下去，並打開send to port 的script 打開在裡面打關於最短路徑的code在此不多做說明(主要是給最近的available port，如果都沒有available port 就給最近的port )。

Queue的聯動頗多先在triggers的Onentrys 內打script設定關於” serial ”這個globaltable的值，使serial( (各個Source) , 1 )代表需求item的數量，serial( (各個Source) , 2 ) 代表此Source被解決的item數量，serial( (各個Queue) , 3 )代表此Source失敗的次數。並把全部MAP的輸出總數、失敗總數寫進final這個globaltable的column1、column2，column3代表失敗率。

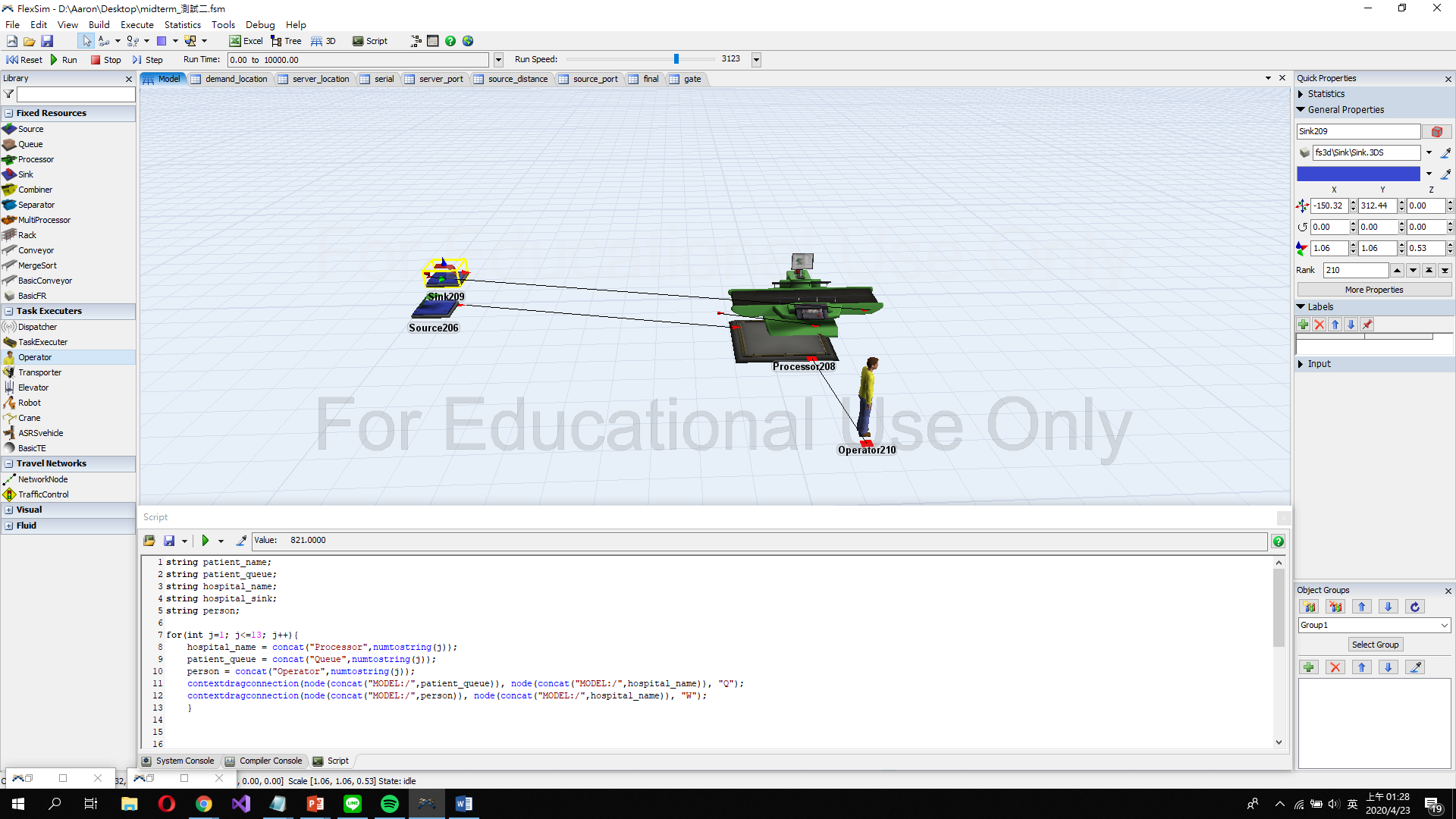
Processor要先在process time的地方輸入根據物件名稱對應Globaltable的expontential lambda，把use operator for process 勾起來(之後解釋)。然後把flow 打開按send to 的script，在裡面寫這個物件要送到哪一個SINK。把Processor 的 Use Transport 打開使Operator 可以運送物件算路程時間，也要把Use operator for processes打開使加工不會趁Operator 外出送貨時偷跑。

Operator要設定的比較少，只需在Trigers 的unload寫script，叫Operator送完貨要記得回醫院。

Sink沒什麼太特別的設定



**拆解版的model連線大致長這樣**



**組合過後符合題目要求的排列**

**此是為了方便看，實際上Source跟Sink會貼合再一起，processor跟Queue也是**

接下來重頭戲是寫一個用來創造各個物件的Script 叫demand\_server\_map，內容主要是把物件創造複製，並且設定地點。其中，創造Source的同時把各個需求點的expontential輸入。在創造Operator時也把各個operator設成reset position。

我的Model主要有80個Sink跟Source， 13個Processor、Operator、Queue，要把Source當成某個村落，Queue當成醫院的等候室，Processor當成急救室，最後再由Operator把item由Processor送回位在Source同地點的Sink。

建立完物件，我建了一個叫connection的script，主要把所有物件連結起來。

首先是直接把同地點的Queue與Processor用” A”連線(因為一個等候室只屬於一間醫院)，再用” S “ 把Operator與Processor連線 。

根據距離判斷是否<=250判斷，(Source , Queue)，(Processor , Sink)是否要連線。

然後段同時也順便把80個村莊到13間醫院的距離表建到source\_distance的globaltable(再用shortest server的時候會用到，主要是給source\_port做一個判斷依據)，然後再根據source\_distance建一個globaltable叫source\_port用來給Source判斷最近的port。

我額外又再建立一個script叫disconnect是用來方便解連結除錯的，其code跟connect 很像只是做反向操作。

我又有加寫一個Sript叫destroy\_all\_object是用來清除場上的地圖用的。

最後就是Optimier找最佳解的地方，先創一個( 13 X 1 ) 的table叫gate，並且把此13個變數輸入到Experiment control 的scenario內。Performance measure 以globaltable final( 1 , 2 )總失敗次數為量測值。到Optimizer design內把13個變數設為Binary，並且在旁邊限制式設定13個變數相加等於7( 代表一次挑7個server)，最後把Run Time設成10000跟replicate設成5就可以跑結果了。

我是把MAX Solution設成400，取到的結果後面第四段會提到~~~

**3. Instructions to use your model**

首先把demand\_server\_map打開並run，再打開connection按run此時基本上已經大功告成，剩下要做的就是自己去按opermizer run 求出可行解。

現在檔案內已經將地圖建起來並已成功連線，但如果助教您要把excel的測資改掉的話，就要先把destroy\_all\_object這個script執行(清除場上物件)，再來做前面提到的步驟。

至於績效的部分，把statistics的Experimenter打開我基本上已經設定完了，剩下就是設定Run Time 跟 Max solution ，我自己Run Time設10000，Max solution 設400，所以系統取到400組解時就會停下來了。(結果在後面第四段呈現)

**EXTRA POINT**

**Dash board:**

案statistics內的Dashboard我有建一個叫Dashboard的表，內有Processor status(醫院的使用狀態)、

current waiting patient(現在該醫院正在等候的病患數)、

Source Output Per Hour(每個村庄的輸出數率)、

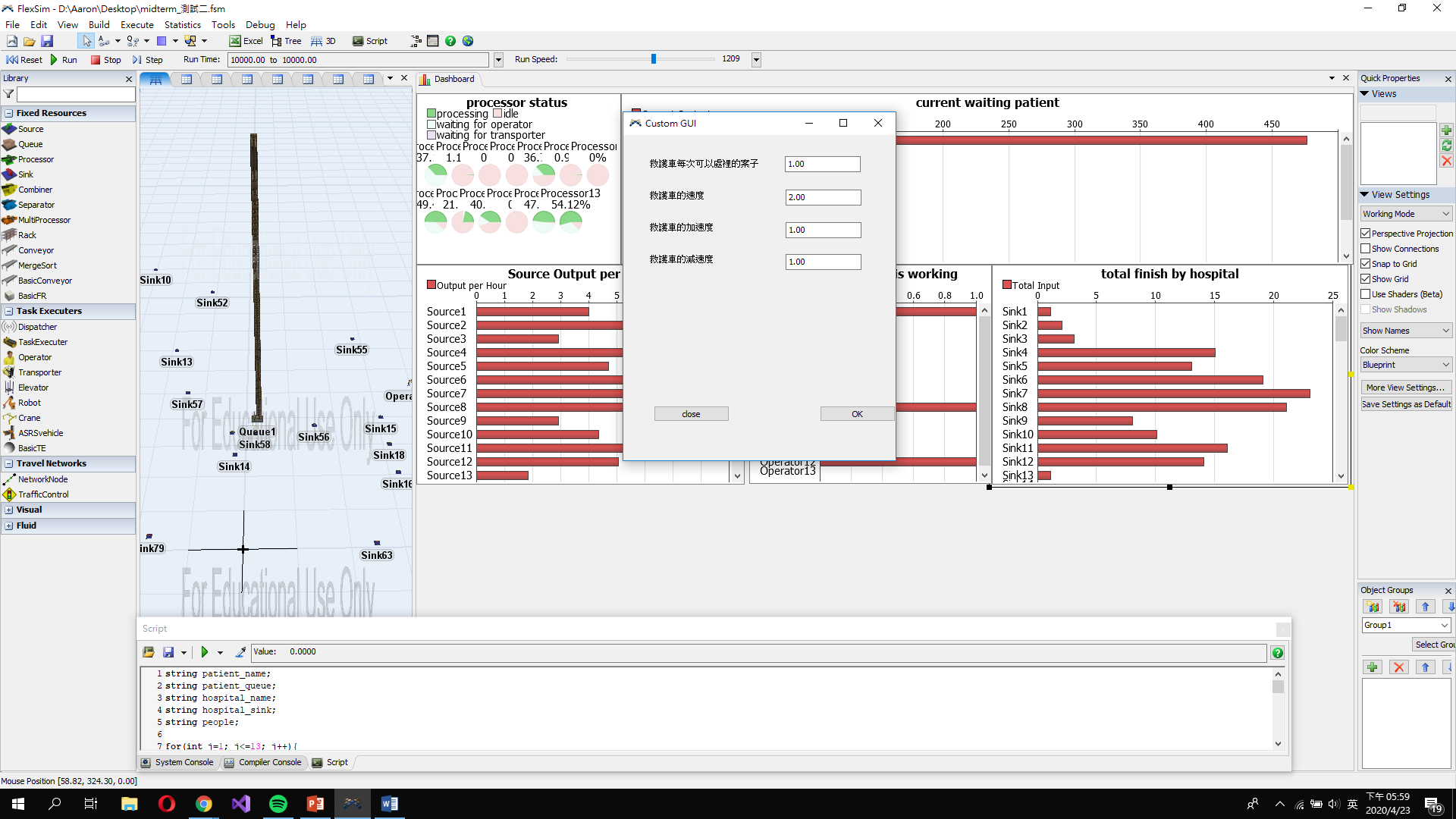
which ambulance is working(救護車是否忙線)、

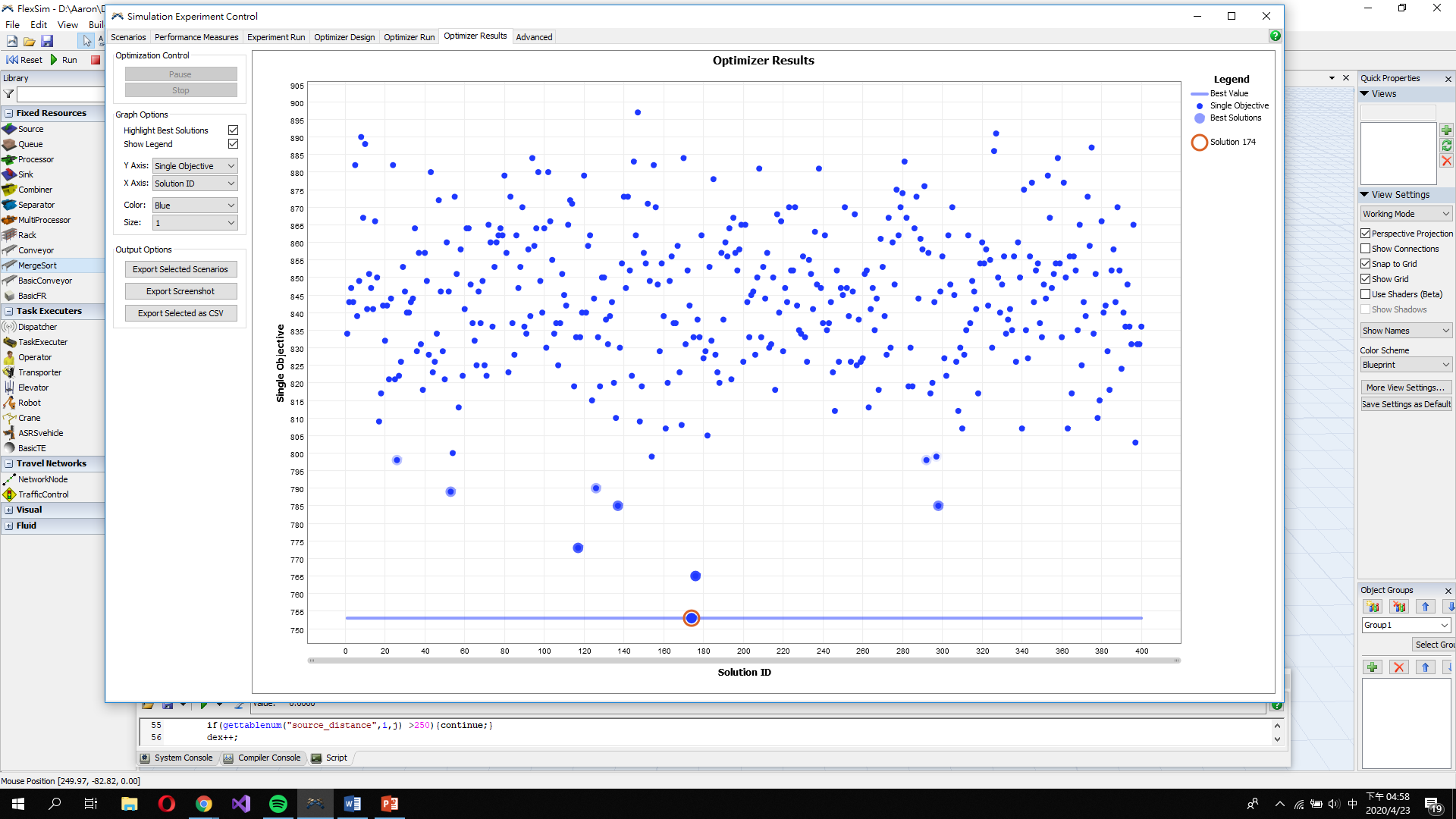
total finish by hospital(該醫院看完診的病患)

**GUI:**

這個我不太確定裡面要放甚麼參數

先在View 打開model control GUI裡面有一個我設定好的使用者介面，可以設定救護車的速度、加速度、減速度、每次搬運輛，(初始值已經根據題目要求設定好了)。

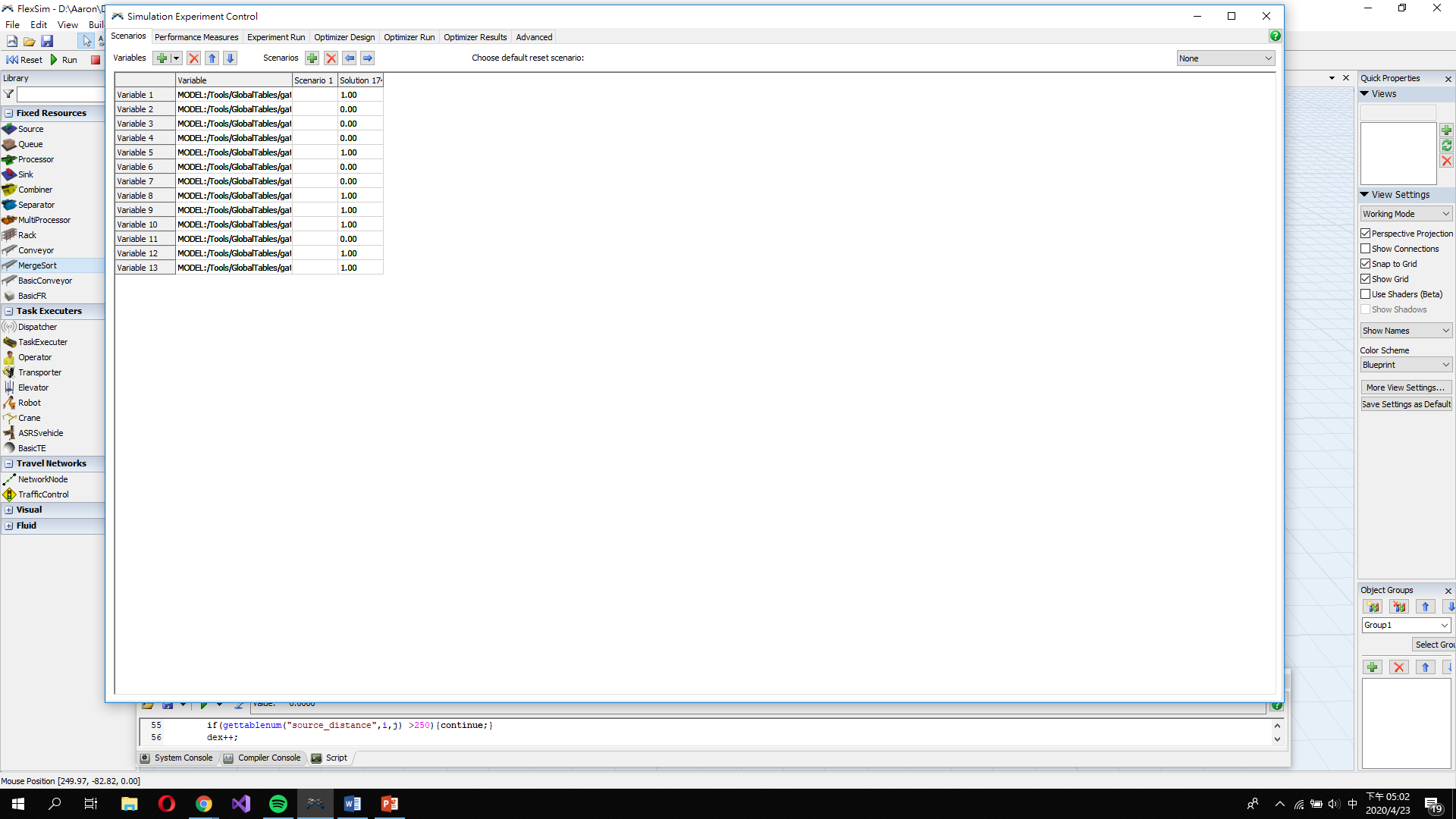


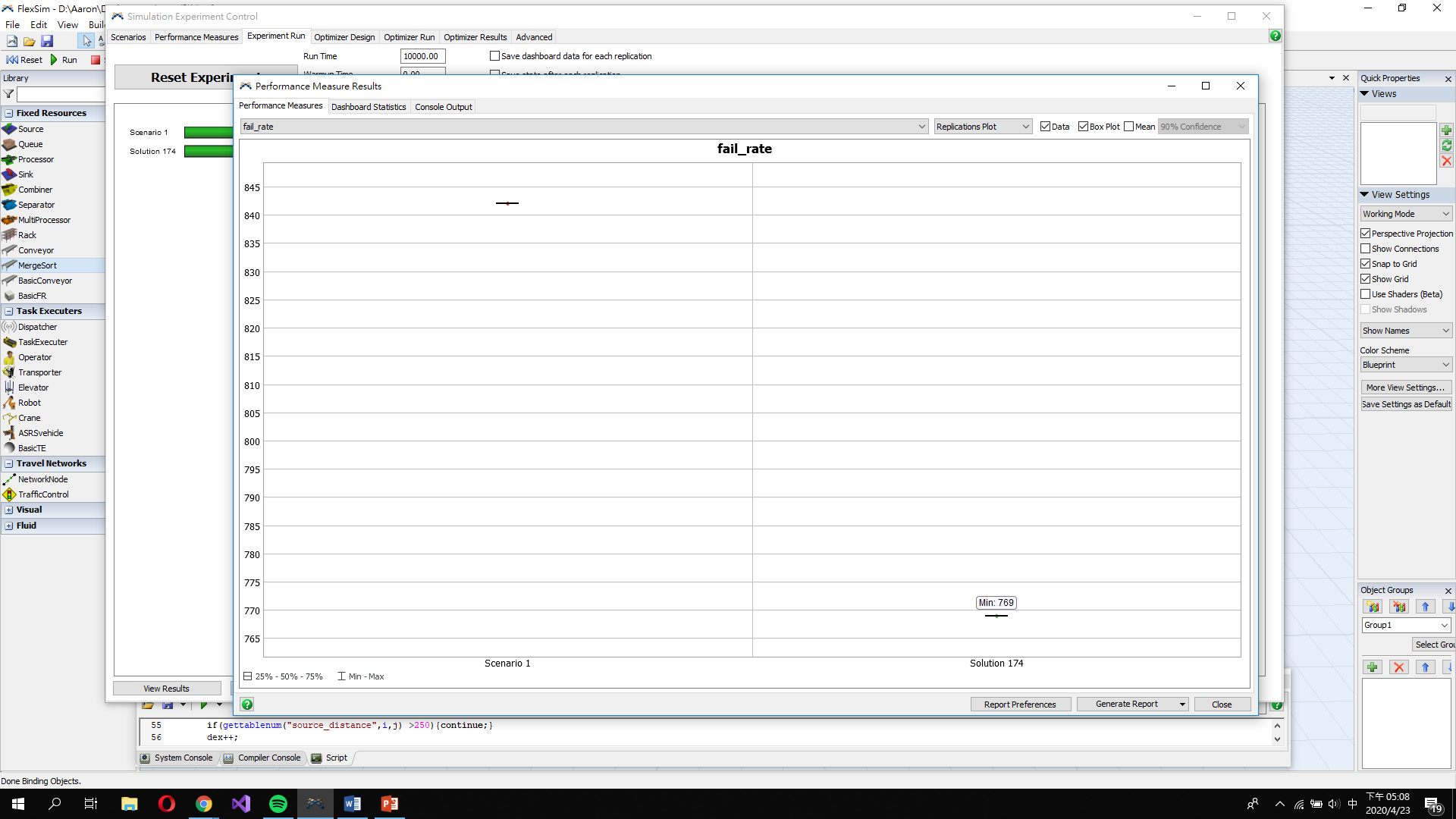
**4. If you did optimization or any other data analysis of running result, put on your insights.**

**最佳scenerio174**

**400組解大致呈現這樣**

**這組解挑選了醫院1、5、8、9、10、12、13(7間醫院)**

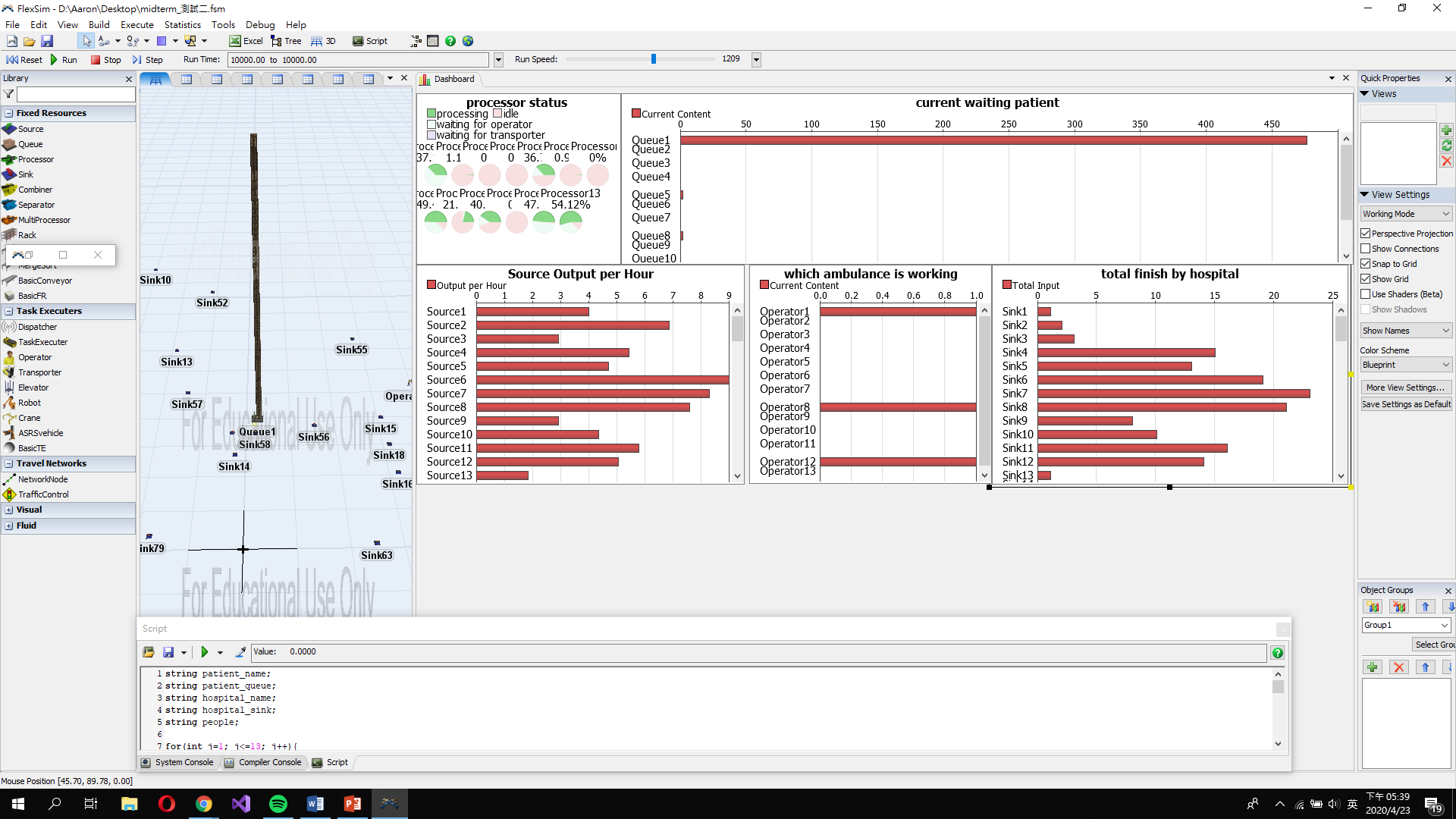


****

**總失敗次數為769次**

**這組解挑選了醫院1、5、8、9、10、12、13(7間醫院)**

但是最佳解scenerio174有個缺點，之所以總失敗次數這麼低是因為犧牲了Queue1(醫院一)的能力負荷(這也跟我的操作策略有關)

**我的操作策略: 如果某村莊發出需求針對距離該村狀最近的醫院排序給此病患派送，但，如果此時每間醫院都忙線，就把這個病患放在最近的醫院給他排隊(如此一來犧牲這一類病患，就不會因為要給這些病患長途跋涉到其他很遠地區的醫院，而延誤了其他地區的救護需求)，如上圖可知醫院一附近的需求超多，其他地區雖然排隊的人很多，但相對醫院一真的是相形失色。**

我對此次參數的見解，我發現80個村落當中每個村落的lambda都頗大(意味著單位時間內需求大)，一開始7間醫院幾乎都可以正常服務病患，但一有排隊的情況發生後面幾乎都會屬於fail的情況發生了，所以越到後面fail rate 就會一直往上飆升，這次我做的成果fail rate 為0.88相對於其他逼近0.99算非常好的結果了，也因此代表著這80個村落只有7間醫院7台救護車，是很吃力的!!