

Courtesy Umbrellas

張允隆
0756053

林榆軒
0756614
s410385018@gmail.com

林柏諺
0656163
lin.cs06g@nctu.edu.tw

ABSTRACT

公共建築物的入口通常會放置傘桶，讓人們在下雨天時，可以將雨傘放進去，以免弄濕建築物內部。而原本在建築物裡的人，離開建築物時，若看到外面下雨，則會從傘桶裡面拿傘，在傘數量與總人數不均時，會導致後面的人無傘可用。因此會發生了一部份人因為擔心從建築物出來時無傘可用，選擇無視傘桶，直接將傘帶到建築物內部的情況。

本題目假設在一個封閉的環境下，有數座建築物、若干人。人會在建築物之間移動，有些可能帶著傘，也有可能不帶著。每座建築物都有傘桶，一開始傘桶內有一定數量的傘，隨著人在建築物之間移動改變數量。最終希望藉由模擬建築物、人流與傘的變化，提供人們在進出建築物，將傘放進傘桶，或是帶在身上的最佳策略。¹

Keywords

Keywords are your own designated keywords separated by semicolons (“;”).

1. INTRODUCTION

1.1 雨傘共用

公共建築物的入口通常會放置傘桶，讓人們在下雨天時，可以將雨傘放進去，以免弄濕建築物內部。原本在建築物裡的人，離開建築物時，可以從傘桶裡面拿傘，然而在傘數量與總人數不均時，會導致後面的人無傘可用。因此會因為擔心從建築物出來時無傘可用，選擇無視傘桶，直接將傘帶到建築物內部，造成地板濕滑、建築物內部潮濕的情況。

若是能在公共場所提供適量的雨傘，在下雨時提供給未帶傘的人使用，進入建築物時將公共雨傘歸還，可省去平時為了預防下雨而隨身攜帶雨傘的麻煩，亦可避免建築物在下雨時常常因將雨傘帶入室內而造成的髒亂。

1.2 需要多少愛心傘

為了解決下雨時所造成的麻煩，有了愛心傘的出現，無論是善心人士將多餘的傘放置於建築出入口的傘桶，亦或是建築的管理員特意將傘置於傘桶，對於忘記帶傘的人來說都是如同救星一般。

若時能在所有建築物前放置無數的愛心傘，理所當然地可以解決雨天時忘記帶雨傘所帶來的困擾。然而這並不

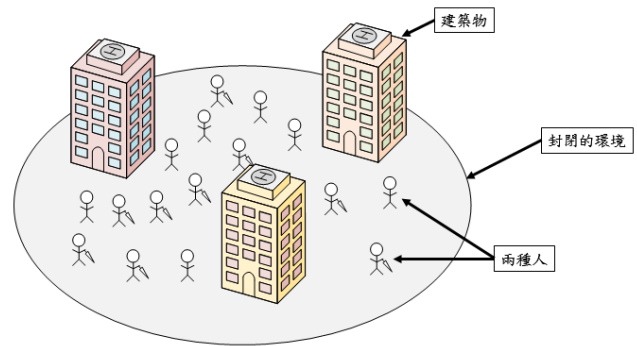


Figure 1. 封閉的環境示意圖，本模擬模擬一封閉的環境，裡面有固定數量的建築物、人口數、雨傘。

可能，實際上人們沒辦法在建築物前放置無數的愛心傘，無論是成本、空間都不足夠。

退而求其次，我們可以選擇放置足夠/適量的愛心傘，然而多少愛心傘可以滿足需求呢？本實驗即是以此為出發點，希望藉由程式模擬愛心傘的使用狀況，分析人口以及足夠多雨傘數量之間的關係，以及愛心傘使用後是否歸還對於整體愛心傘使用循環的影響。

2. BACKGROUND

2.1 OpenMP

OpenMP 是由 OpenMP Architecture Review Board 牽頭提出的，並已被廣泛接受的，用於共用記憶體並列系統的多執行緒程式設計的一套指導性注釋（Compiler Directive）。OpenMP 支援的程式語言包括 C 語言、C++ 和 Fortran；而支援 OpenMP 的編譯器包括 Sun Studio 和 Intel Compiler，以及開放原始碼的 GCC 和 Open64 編譯器。OpenMP 提供了對並列演算法的高層的抽象描述，程式設計師通過在原始碼中加入專用的 pragma 來指明自己的意圖，由此編譯器可以自動將程式進行並列化，並在必要之處加入同步互斥以及通訊。當選擇忽略這些 pragma，或者編譯器不支援 OpenMP 時，程式又可退化為通常的程式（一般為串行），程式碼仍然可以正常運作，只是不能利用多執行緒來加速程式執行。[1]

¹ Source code: <https://github.com/s410385015/PP-FinalPRJ->

2.2 POSIX Thread

跳至搜尋 POSIX 執行緒（英語：POSIX Threads，常被縮寫為 Pthreads）是 POSIX 的執行緒標準，定義了創建和操縱執行緒的一套 API。實現 POSIX 執行緒標準的庫常被稱作 Pthreads，一般用於 Unix-like POSIX 系統，如 Linux、Solaris。但是 Microsoft Windows 上的實現也存在，例如直接使用 Windows API 實現的第三方庫 pthreads-w32；而利用 Windows 的 SFU/SUA 子系統，則可以使用微軟提供的一部分原生 POSIX API。[2]

3. PROPOSED SOLUTION

本研究將分析雨傘量與人口之間關係的工作分為兩個階段：1. 定義模擬的模型，一個包含建築物、人、雨傘的封閉環境 Figure 1，以及其相關要素 2. 依據定義的模型描述並假設一個輸入-輸出-處理定義明確規範的模擬過程 3. 使用模型藉由調整輸入參數來模擬各種情況，從模擬結果分析雨傘的用量與人口的關係。

3.1 模擬模型

此模擬目標為給定初始狀態，包含儲存特定數量[雨傘]的[建築]、能夠分成好幾類的[人群]在建築之間的分布、[天氣]狀態、各類人在建築間移動的規則，利用這些狀態來進行人群在建築之間移動的模擬。其中雨傘的使用會受天氣影響，天氣和人群在建築間移動的方式將會受[時間]影響。整個模擬的模型屬於封閉模型，具有固定建築物數量、人口數量、雨傘，並且此模型的過程可不斷循環。

模擬環境、物件與假設：

3.1.1 環境

環境是模型中影響物件行為的因素，但是它們本身不會受到物件影響。主要為下列三個要素：

- 時間：模型在進行模擬時，是以一個基礎的時間單位進行，每一個模擬循環即是從當前時間模型中的環境和物件，計算出下一個時間的環境和物件。在每個時間單位進行一次人物移動。在本研究中，以一個基礎的時間單位為一小時計算，。
- 天氣：模型僅將天氣分為晴天和雨天，其中晴天即表示人們離開建築時不會帶走雨傘，雨天即表示人們離開建築時會帶走雨傘，若沒有雨傘可帶，則會被迫駐留在建築中。
- 建築：建築是人們可以躲雨、儲存雨傘的地方。

3.1.2 關注物件

關注物件是我們在模擬過程中需要觀察的對象，模擬的目的即是觀察物件的行為並進行統計。共有兩類主要物件，分別為人、雨傘。目的即是為了關注人在建築物之間的流動，以及雨傘的分布。

- 人：在我們的模擬中，在每個時間點於各個建築物中移動，根據天氣情況決定是否攜帶雨傘離開建築

物。本模型將人分類，不同類型的人在各個建築物間移動、活動週期各不相同。

- 雨傘：雨傘是用於讓人們在移動時不受雨天天氣影響的工具，會隨著人們移動一起移動到其他建築。

3.1.3 假設

為了進行模擬，我們依據現實情況建立我們的模型，並為了方便進行模擬，我們假設下列事項永遠成立：

- 人進入建築時都將攜帶的雨傘放入傘桶，此傘桶是所有人共有的，亦即其他人離開建築物時可以隨意拿取。
- 人在建築間移動時，若是晴天則不會從傘桶中取走雨傘，雨天時從傘桶中取傘，若未能從傘桶中取得雨傘則無法離開建築，同樣的若是傘桶中有雨傘，則必定可以離開建築物。
- 在模擬的每個時間點，人必定會出現在建築物中。亦即是假設每次移動人物可直接從原本所在建築物移動至下一目標建築物，不再路途中逗留。

3.2 模擬過程

有了 2.1 節對整體模型的描述，我們進一步描述使用這個模型模擬實際人口流動的過程，透過輸入-處理-輸出的方式表達，並描述模擬過程的演算法。

3.2.1 輸入

- 模擬時間總長度、循環週期： T 、 R
- 每個時間點的天氣狀態： $w[i]$, $0 \leq i < T$
- 人的種類數量： K
- 每種人的數量： $m[i]$, $0 \leq i < K$
- 建築物數量： B
- 每棟建築物初始的人口分布： $d[b][k]$, $0 \leq i < B$ 、 $0 \leq i < K$
- 每棟建築物初始的雨傘數量： $u[i]$, $0 \leq i < B$
- 每種人在建築物之間的移動規律 $p[k][r][i][j]$, $0 \leq i < K$ 、 $0 \leq r < R$ 、 $0 \leq i, j < B$ ，表示第 k 種人在 $r+nR$ 時間點從建築物 i 到建築物 j 的機率，若 $i=j$ 則係是該時間點不離開原本建築的機率。

3.2.2 處理

從初始狀態開始進行 T 次模擬，以 R 為一個循環。

3.2.3 輸出

各個時間點的人群、雨傘分布、多少人因為沒有雨傘而無法移動以及這之中又有多少人是原本帶傘進入建築，離開時卻沒有傘。

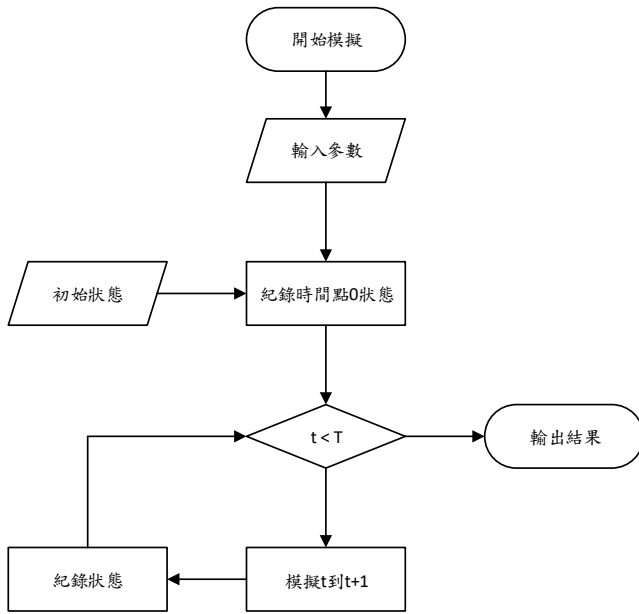


Figure 2. 模擬演算法，輸入參數後進行模擬，重複 T 個回合。

3.2.4 模擬演算法

我們針對每個人分別進行處理，根據其人種、所在建築物與當前時間，首先我們利用移動規律矩陣 p ，得到他前往各個建築物的機率，並透過亂數決定其最終去向。再來根據那個時間點的天氣狀態以及所在建築物的雨傘數量決定是否能出發前往目的地。如此不斷重複，直到 T 時間點為止 Figure 2。

我們會統計在模擬過程中有多少人成功移動到目的地，以及多少人因為沒有雨傘而被迫滯留。

4. EXPERIMENTAL METHODOLOGY

這裡我們描述如何將 2.2.4 小節的模擬演算法平行化。我們將憑下化的工作分為兩個階段，在兩個階段之間需要一個記憶體 barrier。假設在時間點 t ，每棟建築物內的人數標記為 $s[b]$ ，表示第 b 棟建築物有哪些人。

4.1 分配工作

假設共有 L 個 thread，我們將所有建築物平均分配給 L 個 thread，每個 thread 負責處理被分配至建築物中的人，計算下一個時間點這些人分配到各個建築物的人數： $q[h][b]$ 表示第 h 個 thread 在時間點 $t+1$ 有多少人在第 b 個建築物中 Figure 3a。在處理得同時將沒有雨傘可用的人留在原建築物。

4.2 合併結果

將上一階段的結果 q 以第二維度，也就是建築物維度平均分配給 L 個 thread，每個 thread 負責將被分配至建築物 $t+1$ 時間點的人數加總 Figure 3b，即可計算出時間點 $t+1$ 的 $s[b]$ 陣列訊息。

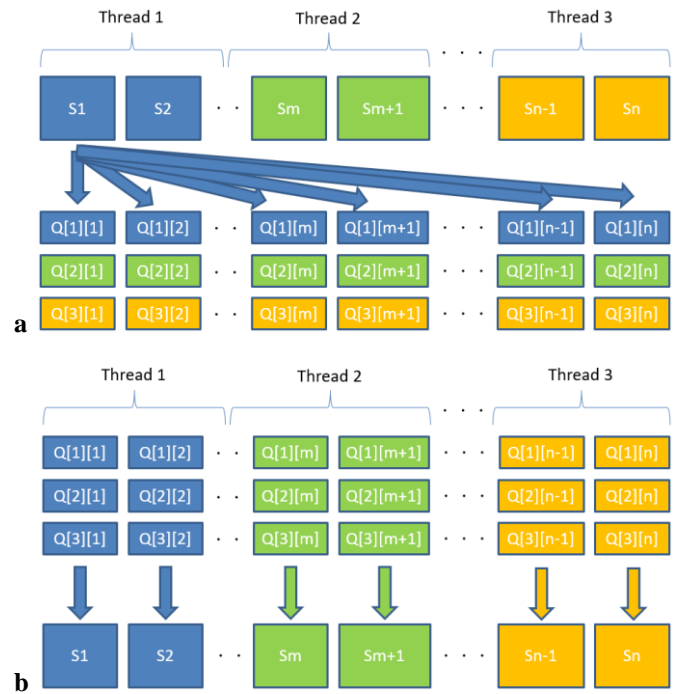


Figure 3. 平行化之分配、合併示意圖。

5. EXPERIMENTAL RESULTS

我們依據這個第 3 段，分別以 OpenMP、Pthread 實作演算法，比較使用不同 Thread 數量下 OpenMP、Pthread 之間的效能差異。並且比較不同建築物數量、總人口數量對模擬時間長短的影響。

Figure 4 測量不同 Thread 下 OpenMP、Pthread 模擬時間所花的長短，以及不同建築物數量對模擬時間的影響。我們可以發現到 Pthread 所花費的時間約為 OpenMP 的 80%，整體上使用 Pthread 比 OpenMP 快。建築物的數量成長與花費時間成線性關係。

Figure 5 測量不同 Thread 下 OpenMP、Pthread 模擬時間所花的長短，以及不同人口數量對模擬時間的影響。Pthread 所花費的時間約為 OpenMP 的 80% 與 Figure 4 相同，人口數量與花費時間同樣呈現線性關係。

6. CASE STUDY

我們以交大作為實際模擬例子，探討各棟建築物事先愛心傘的分配問題。

給定總長度 $T=16$ 、循環週期 $R=1$ ，代表一天中每小時的模擬狀態(扣除午夜少人外出的時間)，人口數為 1 萬 5 千人與 42 棟建築物。並且我們將建築物分為低中高熱度三類與兩種時間狀態(離峰與尖峰)，如表格 1。建築物類型的分配是依照交大真實建築物狀況，低熱度區指的是像中正堂、活動中心這種人煙較少的地方；中熱度區指的是一般的教學大樓，離峰時間人群較為集中、並長時間滯留的區域；而高熱度區則是指，在離峰時間會湧入大量人潮，像是籃球場、或是餐廳等等。

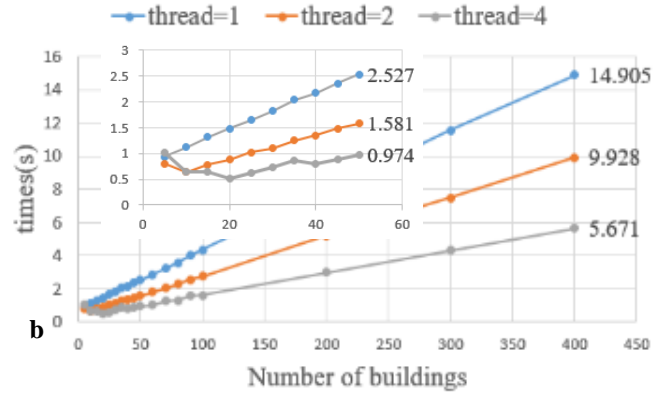
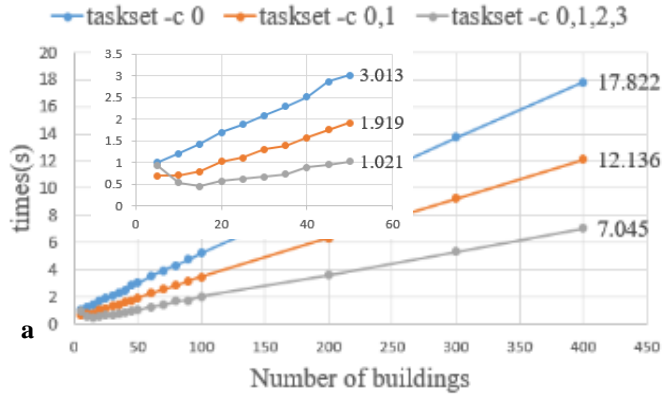


Figure 4. 建築物數量與運算時間比較圖，圖 a 使用 OpenMP，圖 b 使用 Pthread 方法

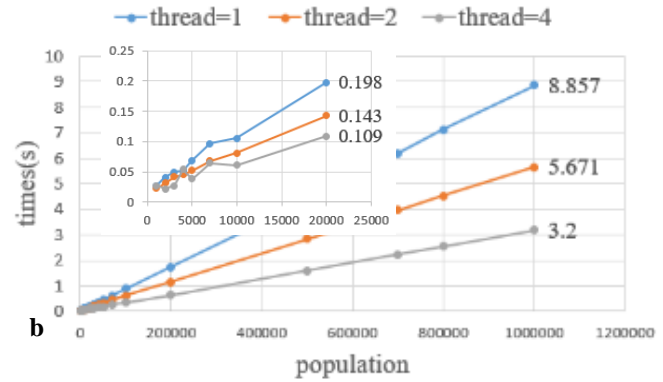
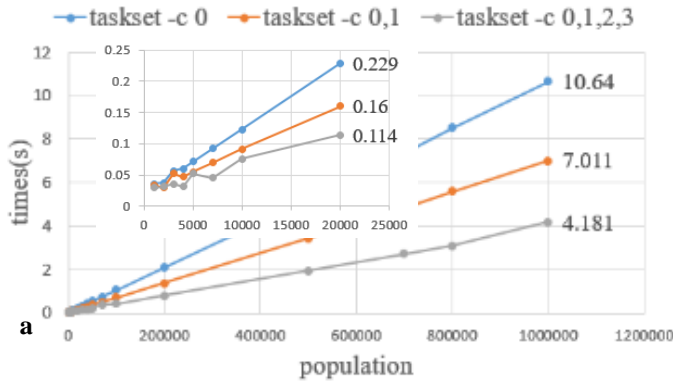


Figure 5. 人口數量與運算時間比較圖，圖 a 使用 OpenMP，圖 b 使用 Pthread 方法

表格 1. 建築物類型、與時間狀態

	離峰	尖峰
低熱度	低移入、高移出	低移入、高移出
中熱度	高移入、低移出	低移入、高移出
高熱度	低移入、高移出	高移入、高移出

初始的人口分布平均分布在中熱度區域。固定以上參數，改變初始愛心傘在各棟建築物的分配數量、愛心傘的數量是以人口相對的比例來做為初始愛心傘雨傘總數，以最少 50% 人口總數的傘數量(7500)隻，逐漸調整上升，模擬傘的使用狀況，考慮真實的操作狀況，一開始每棟建築物的雨傘數量是平均分配的(7500/42)。為了瞭解模擬實驗中，傘的數量是否足夠所有人使用，我們以 starve rate 作為指標，定義如下：

$$Starve\ rate = \frac{starve}{starve + transfer}$$

其中 starve 表示在這輪人口轉移，因為沒有傘可用，而受困的人數；transfer 則是有傘、成功移動到下棟目標建築的人數。我們希望 starve rate 可以越小越好，藉由調

高初始傘數量的比例(10%為一個單位)，找到 starve rate

小於人口數的 0.01%，作為滿足的收斂條件，其結果如 Figure 6。紅色線條表示不考慮晴天狀況，也就是模擬環境無時無刻都是處於下雨狀態；藍色線條表示考慮晴天狀態，使用者離開建築物並不需要傘。

從 Figure 6 中可以觀察到，當一直處於下雨狀態時，傘數量只要達到人口總數的 150%，就可以滿足每個人在每個時間點都能順利的使用到傘。但當我們加入考慮晴天狀況時，雖然一開始的 starve rate 因為有晴天的緣故相對較全是雨天的狀況來的低，但因為人在晴天移動的時候，並不會把傘帶回原處、或帶到下一棟建築物放置，造成傘的分布相當散亂，在下雨天時無法提供給需要的人使用，即使提升初始傘的數量，在各個時間點無傘可用的人還是很多，必須要達到將近人口數的 280% 傘數量，才能使得每一個人在每個時間點都有傘可以用。

7. CONCLUSIONS

我們設計出一套模擬系統，可以藉由改變不同的參數輸入，來模擬各個時間點的人流、以及傘在建築物的移動狀況。藉由輸入參數的情境不同，探討不同種類問題，這裡我們以初始愛心傘的分配問題來做為例子。

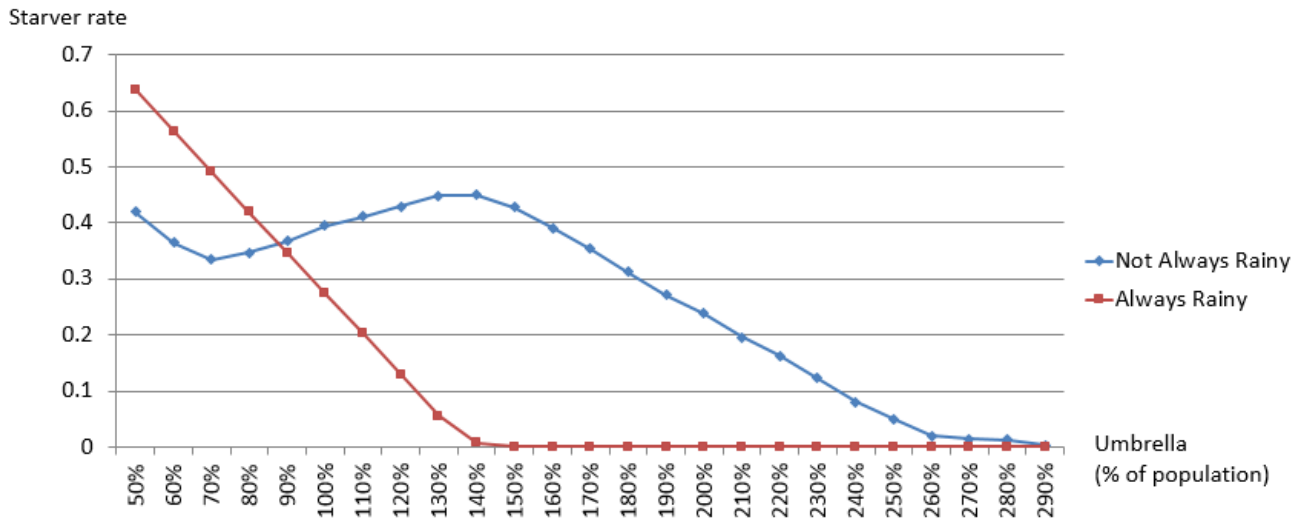


Figure 6. 愛心傘分配模擬結果

並透過不同平行化的方法，來提升整體模擬的效率。從平行化的實驗結果可得知，相比於在大數量的建築物，反而在大數量人口的情況下，平行化後的速度提升較為顯著。推測是因為雖然平行化的步驟是以建築物做為單位，但在模擬系統中，主要的移動計算單位還是人，所以在人口數大的時候，平行化的效果會比較顯著。

在 Case Study 的部分，我們以交大做為環境例子，探討愛心傘的分配，根據結果觀察到非常有趣的現象。當處於一直下雨的狀態時，只需要約略大於人口總署 50% 的傘就可以滿足所有人在所有時間點都有傘可以使用。但當我加入晴天的狀況時，傘的需求量卻會大幅度的升高，需要人口總數的 280% 傘量才足夠，也就是一個人平均要用到 2.8 把愛心傘。其原因是因為在在下雨的時候，使用者從 A 處拿傘，到 B 處，但到了下一個移動時間點可能是晴天，使用者離開時並不會帶走傘，也就是將傘丟在原本的建築物，而造成傘在建築物的分布與人在建築物的分布不一，造成我們必須在初始時就放置大量的雨傘，才能改善這個問題。這個模擬結果告訴我們，如果每個人拿了愛心傘都沒有歸還到原處，用完就隨處放置，其實會造成相當大的資源浪費，因此在這呼籲大家養成歸還愛心傘的好習慣！

8. REFERENCES

- [1] "OpenMP " <https://zh.wikipedia.org/wiki/OpenMP>
- [2] "POSIX 執行緒" <https://zh.wikipedia.org/wiki/POSIX> 线程
- [3] 應用於大規模人群移動之分析、模擬與視覺化之研究. 王昱舜。
- [4] 公共自行車系統營運特性大數據分析—以台北 YouBike 為例. 黃晏珊、鍾智林
- [5] 都會區公共自行車租借系統之設計與營運方式研究，成功大學工業與資訊管理學系研究所碩士論文. 張立蓁
- [6] Real-Time Crowd Simulation: A Review. Richard Leggett
- [7] 具溝通機制及客製化事件的人群模擬 A Crowd Simulation Platform with Communication Mechanism and Customized Events. 趙偉銘, 李蔡彥
- [8] 應用系統模擬於記憶體測試人力配置問題 Applying Computer Simulation to Solve DRAM Testing Manpower Allocation Problem. 吳佩珊
- [9] 人群疏散虚拟现实模拟系统 Guard. 王兆其、毛天露、蔣浩、夏时洪
- [10] Establishing a General Pricing Information System by Using Monte Carlo Simulation Methods and its Empirical Studies on Exotic Call Options, Warrants and Strutured Instrument . WAN, Jin-Cai
- [11] Constructing a model of fire loss for Factory-type buildings based on statistic analysis and computer simulation. Jia-Woei Chen
- [12] 電腦模擬應用於改善家具製程之研究. 陳湘琴
- [13] 利用超級電腦做視覺化模擬的分散式處理環境之研究. 李晴
- [14] A study of simulation model for pedestrian movement with evacuation and queuing. Shigeyuki Okazaki and Satoshi Matsushita
- [15] Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance. Jan Dijkstra, Harry Timmermans