

香港教育大學

五年全日制小學教育榮譽學士

數學與資訊科技學系

MTH4902 Honours Project II

畢業論文

學生姓名：張天佑

學生編號：

指導老師：梁景信博士

題目：數字推盤遊戲（n-puzzle）的策略研究

目錄

前言.....	3
研究問題.....	4
研究原因.....	4
研究方法.....	5
文獻探究.....	5
建議方法.....	9
完整示範.....	11
更大數字推盤上的適用性.....	13
延伸討論.....	13
總結.....	15
參考文獻.....	16

前言

本人的畢業論文的題目是「數字推盤遊戲(n-puzzle)的策略研究」。首先簡單介紹一下何謂數字推盤遊戲的構造。數字推盤(n-puzzle)是一塊有一個正方形凹槽的板子，中間放有 n 個面積相同的小方塊， $n = a^2 - 1, a$ 為自然數。

$$\begin{aligned} a^2 - n \\ &= a^2 - (a^2 - 1) \\ &= 1 \end{aligned}$$

凹槽能放 a^2 個小方塊，即是說把 n 個小方塊放進去後會剩餘一個空格。如圖(1)所示，凹槽的板子中間放有 8 個面積相同的小方塊，即是它是一個八數字推盤。小方塊上會寫有由 1 至 n 的自然數，由左至右、上至下排列。本文稱這個形態為「原型」。

1	2	3
4	5	6
7	8	

圖(1)

整個板子可以移動的方塊只有空格相鄰的方塊，玩家就是透過這個設計改變方塊位子。而遊戲的玩法就是先把數字位子調亂(圖(2))，然後還原成圖(1)的形態。

5	1	4
7	6	2
8	3	

圖(2)

研究問題

1. 解決方法

研究的最終目標就是找出一個解決數字推盤遊戲的解決方法。目前大部份相關研究都是以電腦程式來演算大量的移動方法，從而找出可能的解。而筆者希望能找出一種固定的移動法，只要依照這種移動法，就能解決這個問題。

1、可解性

本研究希望能找出一定能解決這問題的方法，然而首先要確認這個遊戲是可解的。如果是傳統玩法，即是原型→弄亂→原型，至少會存在一個解。首先每一次的單格移動都是可逆的。而弄亂的方法是透過一連串的單格移動來進行，因此可以透過逆轉移動的順序來還原，因此必然有解。但是如果是隨機分配小方塊的位置，由於不是基於原型來產生，情況就未必一樣。因為研究中有機會要透過實際操作來測試不同的解決方法或策略，為免浪費時間在不可解的情況上，需要先對其可解性進行研究。

2、避免循環

筆者在曾經在小時候玩過這個遊戲，當時最常遇到的困難就是循環。循環的意思即是說在通過若干次的移動後，方塊的位置並沒有改變。如果陷入循環之中，就即是無法把小方塊移動到正確的位置。因此要解決這個問題，首先就要找出進入循環的原因和解決方法。

2. 最優解或更優解

在找出有效的解決方法後，就可以嘗試考慮如何減少不必要的移動數量，從而找出更優解。

研究原因

研究這個問題是其本身已有值得探索的趣味。而且這也是本人一直以來有興趣的題目，所以很希望能作探討。如果能整理出一個簡單易明的方法，也可以與學生分享。

研究方法

1. 文獻探究

由於筆者之前並沒有對這個問題進行過學術上的研究，因此需要閱讀其他相關文獻，學習與這個問題相關的基礎知識和需要注意的地方，或是借助前人的發現解決問題。同時文獻探究一個非常重要的功能就是為之後的數學探究做準備，例如學習其他學者如何用數學方法去解釋數字推盤遊戲等。筆者在分析文獻的時候會依照以上四個研究問題來進行分類，方便整理。

2. 數學探究

數學探究包括研究與數字推盤遊戲相關的數學概念，從而運用數學知識去找出解決問題的有效方法。由於一些對於數字推盤遊戲的學術研究都有採用類似的方法，因此應該是可行的。運用數學探究的原因是方便進行證明。由於弄亂數字推盤的過程帶有隨機性，會出現不同的形態。如果需要證明方法可行，就要證明它可行於所有情況。如果能用數學知識去證明其方法可行，就不必嘗試所有情況。研究會先由八數字推盤開始進行探究，再逐步推廣至更大的數字推盤。掌握數字推盤的數學規律，如一方塊回到自己原來位置的最短路徑，能令推廣過程更順暢。

數學探究首先是希望能用數學方便探討方塊的移動路徑和規律，因為每次單一方塊的移動都有很大機會會影響到其他方塊的移動，如果能夠歸納出一些規律和關係，可能有助設計解決方法。

然後是兩格同時處理的可能性，在實際操作中會發現有機會兩個相鄰的方塊在弄亂後依然相鄰。因此筆者想如果可以利用這些優勢，可能會有幫助解決這個問題。

最後，如果能設計到新方法，希望可以附上一個完整的證明。證明此方法適用於所有的數字推盤遊戲。

文獻探究

1. 名詞釋義

原型

數字推盤中的小方塊按小至大的順序由左至右、上至下排列的形態。一個 n 數字推盤會有 n 個小方塊，小方塊上的編號為 1 至 n 的自然數。

目標位置

小方塊在原型中位置，就是目標位置。

目標

把大目標(GOAL)轉化為不同的次目標(SUBGOAL)可以幫助解題。設 $G(n)$ 為把一個任意的 n 數字推盤化為原型。 $g(p)$ 就是把編號 p 的方塊移回目標位置。數字推盤是為原型，當且僅當所有方塊移回目標位置，即是 G
 $G(n) = \{g(1), g(2), \dots, g(n)\}$ 。

配置

小方塊的排列方法。

循環

設一個任意的配置 A，若經過若干步的移動後的配置 B 與配置 A 中所有小方塊得的位置都是相同的，就視為發生了循環。循環產生的其中一個情形是 g(i) 已達成，但當想達成另一目標 g(j) 時，會把方塊(i)移離目標位置。然後在想重新達成 g(i) 時，又會把方塊(j)移離目標位置。這個情形在只考慮最短距離時有可能產生。

逆序數

在一個排列中，如果一對數的前後位置與順序相反，那麼它們就稱為一個逆序。一個排列中逆序的總數就稱為這個排列的逆序數。

曼哈頓距離

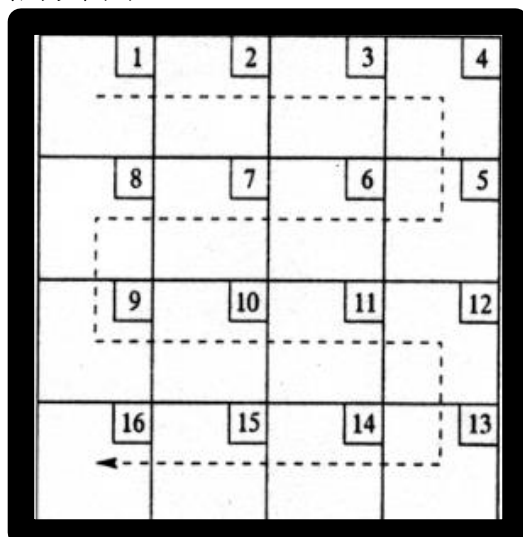
曼哈頓距離的公式為 $d(i, j) = |x_i - x_j| + |y_i - y_j|$ 。x, y 取值於 i, j 兩點的平面坐標。由於數字推盤中的小方塊只可以水平或垂直移動，所以曼哈頓距離比起直線距離更適用於數字推盤的數學探究。

2. 可解性

在數字推盤遊戲中，可以運用方塊移動，使其還原到原型的配置方式，就是可解的。根據數字推盤的定義，一個 n 數字推盤中會有 n 塊小方塊和 1 個空白的位置，即是說共有 (n + 1) 個位置可供重新配置，因此 (n + 1) 個位置會有 (n + 1)! 個配置方法。例如一個八數字推盤就會有 $(8 + 1)! = 9! = 362880$ 個配置方法(Bhasin & Singla, 2012)。但只有一半的配置方法是可以解的。判斷方法是檢視逆序數的奇偶性。

以下是一個簡單的證明：

根據下圖：



想像方塊按此形式排列(Archer, 1999)

使方塊依以上軌跡排列，即 $I = \{1,2,3,4,8,7,6,5,9,10,11,12,15,14,13\}$ 。首先，可以想像在空白的位置有一塊「幽靈方塊」存在，而每次移動則視為「幽靈方塊」與其他方塊交換位置。「幽靈方塊」可以通過以上軌跡移動而不改變實際存在的方塊的排列順序，即是說不會改變逆序數(Archer, 1999)。

當「幽靈方塊」與左右的相鄰方塊交換位置時，不會改變實際存在的方塊的排列順序，即是說不會改變逆序數。

在「幽靈方塊」上下的相鄰方塊交換位置時，由於兩者右方和左方的方塊數是相等的，即是兩者在排列上的差距是偶數。該次交換可視為將一實質方塊前移或後移偶數格，即是在排列中置換了偶數次。

偶數次置換後的數列的逆序數奇偶性和置換前的會是一樣的，因此可解的配置方式的逆序數的奇偶性和原型的逆序數的奇偶性會是一樣，即是偶數。

應用：

4	1	5
2	6	7
	3	8

圖(3)

以圖(3)的八數字推盤為例，各方塊的逆序數 I 是：

$$I(5) = 0$$

$$I(1) = 1$$

$$I(4) = 0$$

$$I(7) = 0$$

$$I(6) = 2$$

$$I(2) = 2$$

$$I(8) = 0$$

$$I(3) = 5$$

而整個八數字推盤的逆序數就是：

$$\sum_{n=1}^8 I(n) = 0 + 1 + 0 + 0 + 2 + 2 + 0 + 5 = 10 \text{ 為偶數，即可解的。}$$

3. 避免循環與解決方法

對於數字推盤的解決方法的研究大多數都是採用 A* 搜尋演算法。A* 搜尋演算法是一種用作搜尋最短路徑的演算法，透過電腦的數據處理能力搜尋出能最快達成目標的移動路徑，如果某幾步會構成循環，就嘗試其他可能 (Markov, Russell, Neller, & Zlatareva, 2006)。簡單來說就是透過巨量的試誤，找出更多或更好的移動法。

也有研究者提出過以電腦記下之前走過的步數，去除重複動作太浪費容量，無法處理更高階的數字推盤，研究者參考地理學模擬水流的做法，提出

了第二個方法，固定一些已經達成目標的方塊(Drogoul, & Dubreuil, 1993)。由於已經達成目標的方塊不能移動，而且程式有找出最短路徑的能力，所以循環基本不會發生。

這個方法相當全面，由於不用大量搜尋，所以可以解決更大的數字推盤問題(Drogoul, & Dubreuil, 1993)。但是缺點就是沒有考慮弄亂後的方塊與目標位置的距離，以致會浪費一定的步數。而且某些方塊固定了後會妨礙到其他方塊回歸自己位置的路徑，所以要依照特定的順序來處理。其中一個取態就是先處理其中一個目標位置為角落的方塊，因為角落擋在其他方塊的移動路徑的機會比較少。試想像在一個八數字推盤中，先達成 $g(5)$ 並固定了方塊 (5) 在目標位置上，其他的方塊要回到目標位置就會很困難。

4. 最優解

尋找數字推盤遊戲的最優解(最少移動步數)一個 NP 困難問題(Goldreich, 2011)。由於無法證明數字推盤遊戲實際上有多少解，因此只能透過搜尋最短路徑去推測最優解的範圍。首先計算限制下計算每格方塊與目標位置的曼哈頓距離，設 $m(p)$ 為編號 p 的方塊與目標位置的曼哈頓距離。由於數字推盤遊戲只可以上下左右移動，因此曼哈頓距離就會是最短距離。

達成 $g(p)$ 最少需要移動 $m(p)$ 步

$$\because G(n) = \{g(1), g(2), \dots, g(n)\}$$

$\Rightarrow$ 達成 $G(n)$ 最少需要移動 $m(1) + m(2) + \dots + m(n)$ 步

$\Rightarrow G(n)$ 的最優解所用的步數 $S(n)$ 大過或等於 $M(n) = \sum_{p=1}^n m(p)$

$$\Rightarrow S(n) \geq M(n)$$

設 $K(n)$ 為一個實際找出的解中所用最少的步數。

$S(n)$ 為最優解所用的步數

$$\Rightarrow K(n) \text{ 不會少於 } S(n)$$

$$\Rightarrow K(n) \geq S(n)$$

$$\therefore K(n) \geq S(n) \geq M(n)$$

以上展示了最優解的可能範圍，以下會簡單介紹尋找最優解的方法。

1. A*搜尋演算法。透過大量搜尋使 $KK(nn)$ 逼近 $MM(pn)$ ，從而收窄 $SS(nn)$ 的可能範圍。缺點同上述，由於運算量極大所以有其技術上的局限性。
2. 遺傳算法。遺傳算法是借鑒於生物遺傳演化所生的算法。在數字推盤遊戲的應用上，就是把若干解法視作「生物」，然後讓不同的「生物」互相「繁殖」，「繁殖」之後出現的「後代」會混合兩者的移動法，然後讓更優解的「後代」再「繁殖」直到找到更優解，如此類推(Bhasin, & Singla, 2012)。

綜合來說，尋找最優解的方法主要都是以數量取勝，反而較少直接從移動方法上著手。

建議方法

基本移動

為方便解說，下文會以 (k, h) 來表示不同方塊， k 為方塊的號數而 h 為方塊的位置。當 $k=h$ ，即是方塊回到自己的位置。

設要移動的方塊為 $(4, 3)$ 。試想像在目標位置畫一條垂直線，原本位置畫一條水平線，兩線成一交點。根據曼哈頓距離的定義，由原本位置到交點，再到目標位置，就是最短的路徑。兩者的線可調轉，不在此處詳解。但是要移動還需要借助空白方塊來進行，因此實際操作不一定能做到。

2		4
3	6	1
8	7	5

移動位置順序

為了防止循環的情況發生，筆者參考 Drogoul and Dubreuil (1993)的方法，把已達成目標的方塊固定好。由於固定方塊會阻擋其他方塊的移動路徑，因此應該由相鄰路線最少的位置作開始。以一個八數字推盤為例，角落位置 1，3，7 的相鄰路線為 2；邊界位置 2，4 的相鄰路線為 3；中心位置 5 的相鄰路線為 4。因此應先處理角落位置，即 1，3，7 其中一個。由於最後空格會在右下角，所以先處理左上角，即完成 $g(1)$ ，如果按照剛才的結果，現在應該處理 3 或 7。但是 1 號方塊已經不能移動，即是有兩條路線會消失，所以現在 2，4 的相鄰路線為 2。在這個情況下，2，3，4，7 的相鄰路線都為 2，理論上處理任何一點都是可以。現在假設完成 $g(2)$ ，可以看到 3 的相鄰路線只有 1，在這個情況下會變得有點困難。相鄰路線只有 1 的時候會有兩情況，以下會逐一解釋：

假設已完成的方塊不可移動

情況一：空格方塊不在目標方塊

如果需要的方塊不在目標位置，就要把它移動到目標位置旁邊。由於相鄰路線只有 1，無法從另一方向讓出空格，需要的方塊無法進入目標位置。

情況二：空格方塊在目標方塊

如果需要的方塊在目標位置旁邊，自然可以馬上解決，但整體而言可能性不高。但空格方塊只有一個，因此要進行移動，就要把另一方塊暫時塞入目標位置。結果與情況一同理。

如果根據 Drogoul and Dubreuil (1993)提出的方法，玩家可以先把先邊線上其他方塊暫時移走。這時，目標位置的相鄰路線就會收就會變回 2，然後把目標方塊移入目標位置，最後再還原邊線上其他方塊。這個是可行的方法，但是會

破壞「固定好目標已達成的方塊」的原則，和帶來陷入循環的風險。因此筆者提出一次處理兩個方塊的方法，來避免陷入路線只有 1 的困境。

畫圖法

但是，有一種方式是一定能成功移動的。以方塊為點，畫一個閉合圖形，點不一定要包含全部方塊，但一定要有要移動的方塊，空白方塊和在目標位置的方塊。然後以與空白方塊相鄰的兩點為起點和終點，逐一移動。由於閉合圖形有必定可以一筆畫完的特性，即是說空白方塊可以移到圖形中任何一點。而空白方塊可以讓其他方塊移動，所以移動的方塊必然可以到目標位置。而且這個方法有一個特性，就是方塊會順著次序移動。這個特性可以讓玩家在移動前對結果可以有一定的預判，可以用作處理某些較多障礙的位置。不過要注意一定要是簡單圖形，因為一個空白方塊無法同時讓交叉點的兩方同時通過。

相連畫圖法

這個方法是由前文描述的「畫圖法」所發展出來的，適用於兩個目標位置相連，而其中一方完成都會令對方的相鄰路線剩下 1 的情況。在使用時，同樣需要畫一個閉合圖形，但是有條件限制。以下會逐一說明：

1．圖形要包含兩個目標方塊，兩個目標位置的方塊和空白方塊

2．兩個目標方塊必須相連

由於兩個目標位置相連，如果目標方塊中間隔了其他方塊，便無法進入適當的位置。

3．目標位置的順序和目標方塊的順序要一致

以圖形的任何一點開始以順（逆）時針拉排次序。設目標位置 A，B 相連，A 為先。由於兩者相連，因此

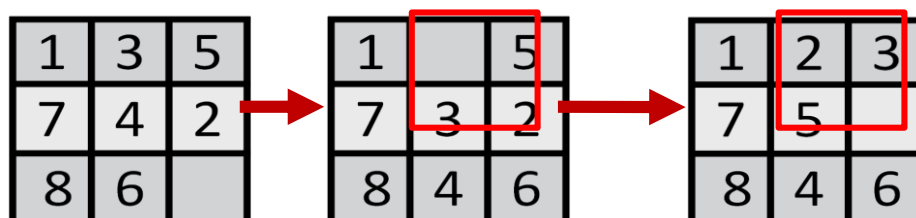
$$\text{序數}(A) - \text{序數}(B) = 1$$

同時目標方塊(A, h1)和(B, h2)在圖形中的序數，必需有以下的關係

$$\text{序數}(A, h1) - \text{序數}(B, h2) = 1$$

就是順序一致。以圖像來解釋，就是如果目標位置的順序是先 A 後 B 的，目標方塊的順序也要是先 A 後 B。

雖然這個方法是針對路線只有 1 的情況而設計出來，但是如果條件許可的話，也不妨在其他位置使用。以下會附以圖像說明實際操作方法。



當上方及左方的位置完成後，由逆序數理論可知剩下的三個方塊必定可還原，而且只需簡單位移即可(Archer, 1999)。因此可以證明，以上方法必定可以解決 8 數字推盤問題。

完整示範

5	1	4
7	6	2
8	3	

以下會示範一次如何以上文所述的方法完成這一個拼圖。根據指定的順序，首先要完成 **g(1)**。即是要將(1,2)轉成(1,1)。為方便解釋圖中會展示其相應的圖形。

5	1	4
7	6	2
8	3	

→

1		4
5	6	2
7	8	3

圖形包括要移動的方塊，空白方塊和在目標位置的方塊，可以進行移動及完成 **g(1)**。接下來可以進行 **g(2,3)**和 **g(4,7)**，由於兩者皆是兩個目標位置相連，而其中一方完成都會令對方的相鄰路線剩下 1 的情況，因此要用相連畫圖法

。現在先完成 **g(2,3)**。

1		4
5	6	2
7	8	3

→

1	2	3
5	4	
7	6	8

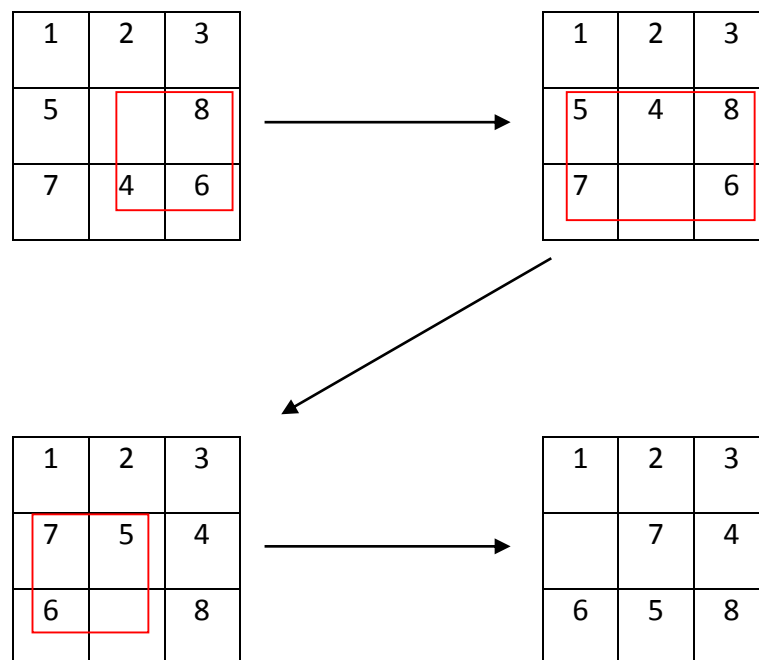
從右圖可見圖形已滿足所需的條件，(2,6)和(3,0)位置相連且與目標位置對應，可以順著圖形來完成 **g(2,3)**。接下來會到 **g(4,7)**，但是現在未能滿足到相連畫圖法的使用條件，因此要做一些準備。

1	2	3
5	4	
7	6	8

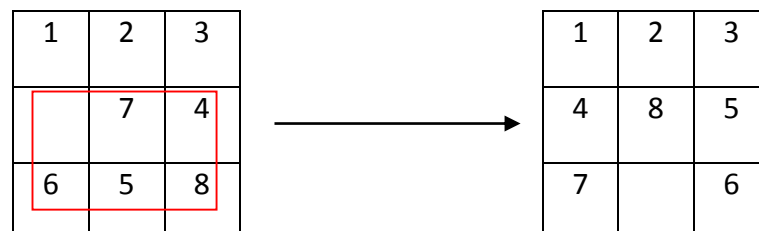
→

1	2	3
5		8
7	4	6

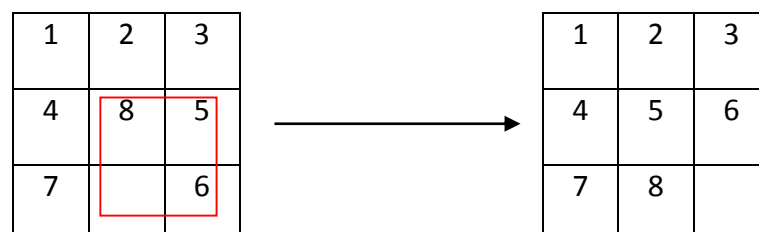
現在滿足了相連的條件，但是若檢示順序，可發現與目標前後位置不對應。這種情況可依以下步驟解決。



首先要確保前後順序，之後只把夾在中間的其他方塊移走即可。現在已滿足所需的條件，完成 $g(4,7)$ 。



剩下的三個方塊沒有難點，按著圖形範圍按任意方向移動，三步內必得解。



數字推盤回到原型，問題解決。

更大數字推盤上的適用性

以下會以數學歸納法證明以上方法適用於任何大小的正方形數字推盤。設 a 為數字推盤的單一邊界格數， Q 為「使用上文所描述的方法」， $G(n)$ 為把一個任意的 n 數字推盤化為原型， $T(n)$ 為把一個任意的 n 數字推盤的上方及左方邊界還原。 $n = a^2 - 1, a$ 為自然數

當 $a = 3, Q \Rightarrow G(8)$ 是正確的 (已證明)

假設當 $a = k \geq 3, Q \Rightarrow G(k^2 - 1)$ 是正確的。

當 $a = k + 1$ ，先無視 $T((k + 1)^2 - 1)$ 以外的目標，邊界的方塊只有三種可能。

可能性一

相鄰路線數目為 2 或以上，可以直接移動。

可能性二

相鄰路線數目為 1，以 Q 中相連畫圖法連同邊界上相連位置解決。

可能性三

相鄰路線數目為 0，只可能出現在左上角，由於 Q 會優先處理，故無需考慮。

綜合以上三個可能性，當 $a = k + 1, Q \Rightarrow T((k + 1)^2 - 1)$ 是正確的。根據定義， $T((k + 1)^2 - 1)$ 以外的部份為一個 $a = k$ 的數字推盤。因此：

$$a = k \geq 3, Q$$

$$Q \Rightarrow T((k + 1)^2 - 1) \text{ 和 } G(k^2 - 1) \text{ is true}$$

$$\Rightarrow G((k + 1)^2 - 1) \text{ is true}$$

根據數學歸納法的定理，上文所描述的方法可以解決任何大小的正方形數字推盤問題。

延伸討論

最優解或更優解

如果就移動策略上來看，筆者提出的方法省去了讓出角落空位的步驟，應該會比較快。但由於有大量不同的情況，如果要證明兩者在實際的效率上存在差異，就需要進行統計學中的假設檢定(Stuart, Ord, & Arnold, 1999)。不過要進行假設檢定，就需要相應數量的樣本才能有準確的結果。樣本可以透過電腦程式模擬取得，但由於資源所限，現時未能進行。有見及此，本文暫時未能完全回應到這個問題。

以下是其中一個可行的方案，參考自 Drogoul and Dubreuil (1993) 的方法：

在準備工作方面，首先搜尋一個能隨機建構出隨機可解方塊的程式。這部份的程式碼可以在不少程式共享平台上找到。然後根據於不同移動方式的相應的解題程式，當中包括按照內文的所述方法所運作的程式。

設按照內文的所述方法所運作的程式為 P1，按照另一個移動方式運作的程式為 P2

第一步:產生一個 8 數字推盤

第二步:在 8 數字推盤上運行 P1，然後計算步數

第三步:在 8 數字推盤上運行 P2，然後計算步數

第四步:重覆第一步及第二步直至有足夠有效的數據

接下來所用的檢定方法，就要視第四步所的數據而定。如果兩者的數據都服從常態分佈

實際應用

雖然這只是一個簡單的遊戲，但是在實際生活上也有應用的潛力。以下所描述的方案為筆者個人的構想，希望有看到的人有興趣實踐。

貨物儲存

A1	A2	A3
A4	A5	A6
A7	A8	o

設上圖為一個倉庫的平面圖，每個寫有 A 的方塊都是一個房間，而 o 則是一個放入及提取貨物的空間，對外的通道就是紅色線的位置。跟遊戲一樣，指定房間可以透過縱橫移到 o，不過要交由機器系統進行。例如要提取 A4 的貨品，首先要輸入指令，使 A4 移到 o 的位置，然後打開門(紅色)，提取貨品，關上門，最後系統自行還原位置。

在運用這個系統的時候，使用者只需要停在門的位置即可，不需走入去尋找，因此，可以節省了原本用作通道的空間作儲存用。而且可以把物件清晰地分門別類放好，易於管理。

其他形狀

另一個可以繼續研究的問題是這個遊戲可否拓展至正方形以外的平面圖形，以及已有方法是否可行。

筆者把形狀的轉換分為兩種，一種為與原本的正方形同倫的形狀，另一種為不同倫的形狀。同倫 (Homotopy) 是屬於拓樸學的一個概念，單來說就是若兩個物件同時可由同一較大物件「擠壓」而成，則稱之為同倫。整體而言，由於限制較大，這種做法的形狀不會有太大變化不大。

首先，新的形狀不應該有單一方塊突出的情況，因為單一突出的方塊沒有存在價值。原因是相鄰路線只有 1 的時候，需要的方塊無法進出該位置。這一點在前文已經解釋過，不再詳述。即是說，就算有單一突出的方塊，它也不會對整個遊戲的解題過程造成任何影響。如以下例子：

1	2	3
4	5	6
7	8	9

另一方面，由於遊戲的規則所限，不一定所有形狀都可以做到。如果依照原有設計，方塊為正方形且只可上下移動，只可以組成由直角組成的形狀。

筆者經初步研究，發現之前的方法應該也可以套用這些圖形之上。原因是即使轉了形狀，但是基本規則和難點並沒有重大變化。不過可能用不同倫的形狀，例如中間開洞，或是立體化，之前的方法就未必能完全套用了。

總結

本文透過把問題分拆成不同部份，成功找出了解決數字推盤遊戲（n-puzzle）的方法和策略。而針對方法的可行性，已在文中附上了完整的證明。但是在尋找最優解方面，由於研究規模所限，仍然需要繼續探討。

參考文獻

- Archer, A. F. (1999). A modern treatment of the 15 puzzle. *The American mathematical monthly*, 106(9), 793-799.
- Bhasin, H., & Singla, N. (2012). Genetic based algorithm for N-puzzle problem. *International Journal of Computer Applications*, 51(22).
- Drogoul, A., & Dubreuil, C. (1993, May). A distributed approach to n-puzzle solving. In *Proceedings of the Distributed Artificial Intelligence Workshop*.
- Goldreich, O. (2011). *Finding the shortest move-sequence in the graph-generalized 15-puzzle is NP-hard* (pp. 1-5). Springer Berlin Heidelberg
- Markov, Z., Russell, I., Neller, T., & Zlatareva, N. (2006, October). Pedagogical possibilities for the N-puzzle problem. In *Frontiers in Education Conference, 36th Annual* (pp. 1-6). IEEE.
- Stuart A., Ord K., Arnold S. (1999), *Kendall's Advanced Theory of Statistics: Volume 2A—Classical Inference & the Linear Model* (Edward Arnold) §20.2.