



Instituto Politécnico Nacional.

Unidad Profesional Interdisciplinaria de Ingeniería
campus Zacatecas.

Ingeniería en Sistemas Computacionales

Análisis de algoritmos.

Profesor: Roberto Oswaldo Cruz Leija.

Grupo: 3CM1

“PROBLEMA DEL CABALLO”

ALUMNA;

Alejandra Monserrath Esparza Ríos.

Zacatecas, Zac., a 28 de octubre del 2019.

INTRODUCCION.

El problema del caballo es un antiguo problema matemático en el que se pide que, teniendo una cuadrícula de $n \times n$ casillas y un caballo de ajedrez colocado en una posición cualquiera (x, y) , el caballo pase por todas las casillas y una sola vez. Lo que resulta en $n^2 - 1$ movimientos.

Muchos matemáticos han buscado una solución matemática a este problema, entre ellos Leonhard Euler.

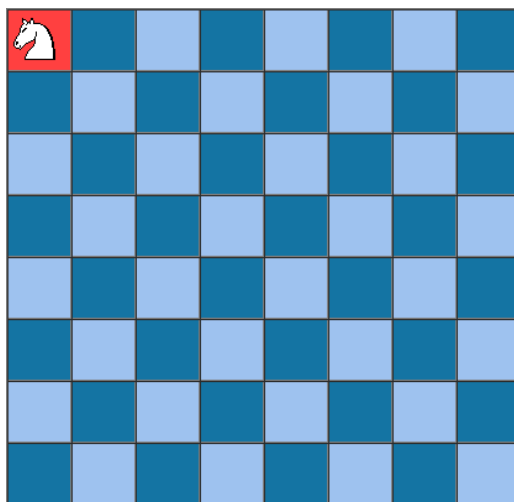
Se han encontrado muchas soluciones a este problema y de hecho no se sabe con seguridad de cuántas maneras diferentes es posible solucionarlo.

Algunas variaciones de este problema han sido estudiadas por los matemáticos, tales como:

- Buscar soluciones cíclicas, en la cual se debe llegar a la misma casilla de la cual se partió.
- Tableros de diferente número de columnas o diferente número de filas.
- Juegos de dos jugadores basados en la idea.
- Problemas usando ligeras variaciones en la forma de moverse el caballo.

El problema del caballo es una forma del problema más general problema de la ruta Hamiltoniana en la teoría de grafos.

Los saltos del caballo consisten en que el caballo de ajedrez colocado en una posición cualquiera sea capaz de recorrer todas las casillas sin repetirlas y sin dejar ninguna sin recorrer.



CARACTERISTICAS DEL PROBLEMA.

- ✓ Recorrer las casillas del tablero de un ajedrez.
- ✓ No repetir las casillas ya recorridas.
- ✓ Pasar por todas las casillas del tablero.

- ✓ Completar los 63 saltos de las 64 casillas.
- ✓ Metodología ensayo y error.
- ✓ Calcular los movimientos del caballo

METODOLOGIA DE SOLUCION.

Paso 1.	Escoja una casilla inicial, si $n \times n$ (el tamaño del tablero) es impar escoja una casilla cuyo color sea blanca, inclúyala al tour y siga al paso 2.
Paso 2.	Pase a la siguiente casilla (la casilla $m+1$) tal que: <ul style="list-style-type: none"> • sea alcanzable legalmente desde la m-ésima casilla • aún no haya sido visitada • es la casilla con el menor número de posibles jugadas Si no existe ninguna casilla que cumpla con las condiciones anteriores, el tour ha terminado. De lo contrario, si existe exactamente una casilla que cumpla las tres condiciones, inclúyala al tour y repita el paso 2 (desde la nueva casilla). De lo contrario, si existe más de una casilla que cumple las condiciones vaya al paso 3.
Paso 3.	De las casillas candidatas pase a la casilla más cercana a cualquiera de las esquinas del tablero. Si existe un empate entre varias casillas pase al paso 4. De lo contrario, incluya la casilla ganadora al tour y regrese al paso 2.
Paso 4.	De las casillas empatadas con menor distancia pase a la casilla más cercana a una de las paredes del tablero. Si existe un empate entre varias casillas pase al paso 5. De lo contrario, incluya la casilla ganadora al tour y regrese al paso 2.
Paso 5.	De las casillas aun empatadas pase a la casilla siguiente dependiendo del orden de los movimientos del caballo en sentido horario y teniendo como punto de inicio el cuadrante correspondiente a la casilla inicial del tour. Solo una casilla puede cumplir esta condición, por tanto, inclúyala al tour y regrese al paso 2.

RESULTADOS.

Posición 1,1

run:

```

1 60 39 34 31 18 9 64
38 35 32 61 10 63 30 17
59 2 37 40 33 28 19 8
36 49 42 27 62 11 16 29
43 58 3 50 41 24 7 20
48 51 46 55 26 21 12 15
57 44 53 4 23 14 25 6
52 47 56 45 54 5 22 13
BUILD SUCCESSFUL (total time: 7 seconds)

```

Posición 1,8

```
run:
60 45 26 39 62 9 24 1
27 38 61 44 25 2 63 10
46 59 42 21 40 11 8 23
37 28 47 58 43 22 3 64
48 57 36 41 20 5 12 7
35 54 29 50 31 14 17 4
56 49 52 33 16 19 6 13
53 34 55 30 51 32 15 18
BUILD SUCCESSFUL (total time: 11 seconds)
```

Posición 8,8

```
run:
47 34 21 30 5 14 19 64
36 31 48 33 20 63 4 13
49 46 35 22 29 6 15 18
60 37 32 51 62 17 12 3
45 50 61 38 23 28 7 16
56 59 42 25 52 9 2 11
41 44 57 54 39 24 27 8
58 55 40 43 26 53 10 1
BUILD SUCCESSFUL (total time: 43 seconds)
```

Posición 8,1

```
run:
50 43 30 61 14 63 28 7
31 60 51 42 29 8 13 64
44 49 58 25 62 15 6 27
59 32 45 52 41 26 9 12
48 53 40 57 24 11 16 5
39 56 33 46 35 18 21 10
54 47 2 37 20 23 4 17
1 38 55 34 3 36 19 22
BUILD SUCCESSFUL (total time: 9 seconds)
```

Posición 8,5

```
run:
46 61 28 51 12 63 26 5
29 50 45 62 27 6 11 64
60 47 54 23 52 13 4 25
49 30 59 44 55 24 7 10
58 43 48 53 22 9 14 3
37 40 31 56 33 16 19 8
42 57 38 35 18 21 2 15
39 36 41 32 1 34 17 20
BUILD SUCCESSFUL (total time: 10 seconds)
```

CONCLUSIONES

Al realizar este problema uno se podría dar cuenta que este sería un caso de camino y circuito hamiltoneano para un grado. Ya que se busca un camino ideal a la vez convirtiéndose en un circuito el cual debe ser recorrido por el caballo.

El algoritmo empleado para este caso muestra todas las posiciones en las que debe recorrer el caballo dependiendo de su primer posición (x0, y0), en el cual se pudo apreciar que al momento de comenzar en las posiciones en las esquinas y en una posición de las orillas es menos tardado en dar la solución que cuando la posición de inicio está en una casilla de en medio, este es más tardado ya que se ocupan más soluciones.

Url de github: <https://github.com/ale-esparza/Problema-del-caballo>