ZALG cv 7

Stavový prostor a jeho procházení

Typické příklady úloh:

- hlavolamy a hry (nejen) pro jednoho hráče (sudoku, šachy, aj.)

Úloha:

- **Zadání:** je dán počáteční stav světa (např. zadání sudoku) a množina cílových stavů (přípustná řešení problému)
- **Cíl:** pomocí nějakých akcí / tahů se potřebujeme dostat ze startu (počátečního stavu) do cíle (jednoho z cílových stavů)

Stavový prostor úlohy

- speciální typ grafu:
  - vrcholy = stavy (situace)
  - hrany = přechody mezi stavy (akce, tahy)

$\rightarrow$ stavový strom (nebo obecnější graf) úlohy

Procházení stavovým prostorem (grafem)

- Metody
  - 1) Do hloubky – metoda prohledávání s návratem (backtracking)
  - 2) Do šířky – metoda vlny (pohyb po vrstevnicích)

- Algoritmy jsou hodně podobné algoritmům na grafech. U backtrackingu ovšem máme rozdíl s tím, že procházíme prostor potencionálních řešení. V případě, že v dané větvi řešení neexistuje, vracíme se zpět. U výpisu stromu (metodou inorder) jsme procházeli větev do hloubky až k listu a po jeho vypsání jsme se vraceli zpět.

Backtracking

- Motivační příklad – Vlk, koza a zelí
- Úloha:
  - Na začátku je zahradník, vlk, koza a zelí na jednom břehu
  - Cílem zahradníka je převézt všechny 3 na druhý břeh
  - Zahradník má loďku a v jednu chvíli smí převážet pouze jednu věc/zvíře
- Pravidla:
  - Loď nesmí plout bez zahradníka
  - Pokud zůstane vlk bez dozoru s kozou, vlk kozu sežere
  - Pokud zůstane koza bez dozoru se zelím, koza zelí sežere

Backtracking – jeden krok rekurze

- Jsme v aktuálním stavu u
- Pokud je to cílový stav, zaznamenáme řešení
- Jinak vyzkoušíme postupně všechny přípustné tahy z u
  - Pomocí tahu t přejdeme do stavu v (krok vpřed)
  - Zavoláme rekurzivně metodu na stav v
  - Po návratu z rekurze vrátíme tah t (krok zpět) - BACKTRACKING

Problém n dam

- Úloha:
  - Vstup: n – rozměr šachovnice/počet dam
  - Cíl: rozmístit na šachovnici typu *n x n* celkem *n* dam tak, aby se žádné dvě neohrožovaly
- Naivní řešení
  - Generuje postupně všechny stavy a každý stav testujeme
  - Časová složitost $\binom{n^2}{n}$

Problém n dam

- Lepší řešení – Backtracking
  - Postupujeme po sloupcích šachovnice zleva doprava
  - V aktuálním sloupci se pokusíme na nějaký řádek umístit dámu
    - Procházíme jednotlivé řádky shora dolu a zkoušíme umístit dámu
    - Pokud se nám to podaří, přejdeme na další sloupec
    - Pokud ne, tak se vrátíme k předchozímu sloupci

- Časová složitost shora omezená O(n!)

Backtracking – rekurze pseudokód

vyřeš(index sloupce j, bool jenJedno)
  if $j \geq n$ ... hotovo
    zaznamenej řešení (matici sachovnice)
    return jenJedno
  endif
  for $i = 1$ to n do... projdeme pozice ve sloupci
    if $[i, j]$ je bezpečná pozice
      $sachovnice[i][j] = true$ ... krok DOPŘEDU
      if vyřeš($j + 1$,jenJedno)
        return true
      endif
      $sachovnice[i][j] = false$ ... krok ZPĚT
   endif
  enddo
  return false

Procházka koněm

- Vstup: šachovnice *n x n*, počáteční pozice koně
- Cíl: Nalézt procházku koněm po šachovnici tak, aby navštívil každé pole šachovnice

Problém reprezentace šachovnice

- Máme koně a na políčku [x][y] a chceme skočit na pole [z][w], abychom ovšem na políčko mohli vstoupit musíme zjistit, jestli tah není mimo šachovnici nebo, jestli jsme dané pole již nenavštívili

- Tedy políčko je přístupné, pokud:
  - $z \geq 0 \ \&\& \ z < n \ \&\& \ w \geq 0 \ \&\& \ w < n \ \&\& \ w\_nenavštíveno$
  - $\rightarrow$ tedy v každém tahu musíme ověřit minimálně hned 5 podmínek

10 x 12 šachovnice

- Datová struktura používaná pro reprezentaci šachovnic 8x8 ve starých šachových motorech.
- Datová struktura je v angličtině nazývaná **mailbox**
- Šachovnice je uložena v jednorozměrným poli velikosti 10x12.
- Šachovnice kromě 64 polí obsahuje 56 zarážkových polí
- Mailbox je těmito poli ohraničen tak, že obsahuje 2 zarážkové sloupce a 4 zarážkové řádky.
- Důvod: jednoduché odhalení tahu mimo šachovnici
- Zmodifikujeme datovou strukturu pro použití na obecné n x n šachovnici

„(n+2)x(n+4) mailbox“

- Indexy

110	111	112	113	114	115	116	117	118	119
100	101	102	103	104	105	106	107	108	109
90	91	92	93	94	95	96	97	98	99
80	81	82	83	84	85	86	87	88	89
70	71	72	73	74	75	76	77	78	79
60	61	62	63	64	65	66	67	68	69
50	51	52	53	54	55	56	57	58	59
40	41	42	43	44	45	46	47	48	49
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

Zarážky

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	91	92	93	94	95	96	97	98	-1
-1	81	82	83	84	85	86	87	88	-1
-1	71	72	73	74	75	76	77	78	-1
-1	61	62	63	64	65	66	67	68	-1
-1	51	52	53	54	55	56	57	58	-1
-1	41	42	43	44	45	46	47	48	-1
-1	31	32	33	34	35	36	37	38	-1
-1	21	22	23	24	25	26	27	28	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Tahy jezdcem

- new_square = square + move

2(n+2) - 2	2(n+2) - 1	2(n+2)	2(n+2) + 1	2(n+2) + 2
1(n+2) - 2	1(n+2) - 1	1(n+2)	1(n+2) + 1	1(n+2) + 2
-2	-1	0	1	2
-1(n+2) - 2	-1(n+2) - 1	-1(n+2)	-1(n+2) + 1	-1(n+2) + 2
-2(n+2) - 2	-2(n+2) - 1	-2(n+2)	-2(n+2) + 1	-2(n+2) + 2

Tahy dámou

- new_square = square + move

2(n+2) - 2	2(n+2) - 1	2(n+2)	2(n+2) + 2	2(n+2) + 2
1(n+2) - 2	1(n+2) - 1	1(n+2)	1(n+2) + 1	1(n+2) + 2
-2	-1	0	1	2
-1(n+2) - 2	-1(n+2) - 1	-1(n+2)	-1(n+2) + 1	-1(n+2) + 2
-2(n+2) - 2	-2(n+2) - 1	-2(n+2)	-2(n+2) + 1	-2(n+2) + 2

Přepočty mezi 2D a 1D souřadnicemi

- Z 2D [x][y]

$$\rightarrow square = 2 \cdot (n+2) + 1 + x \cdot (n+2) + y = (2+x)(n+2) + y + 1$$

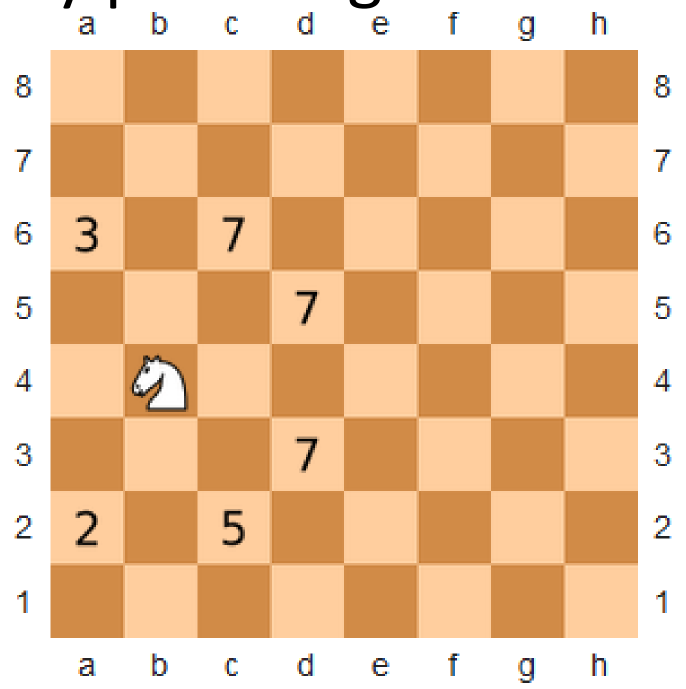
- Z 1D sq

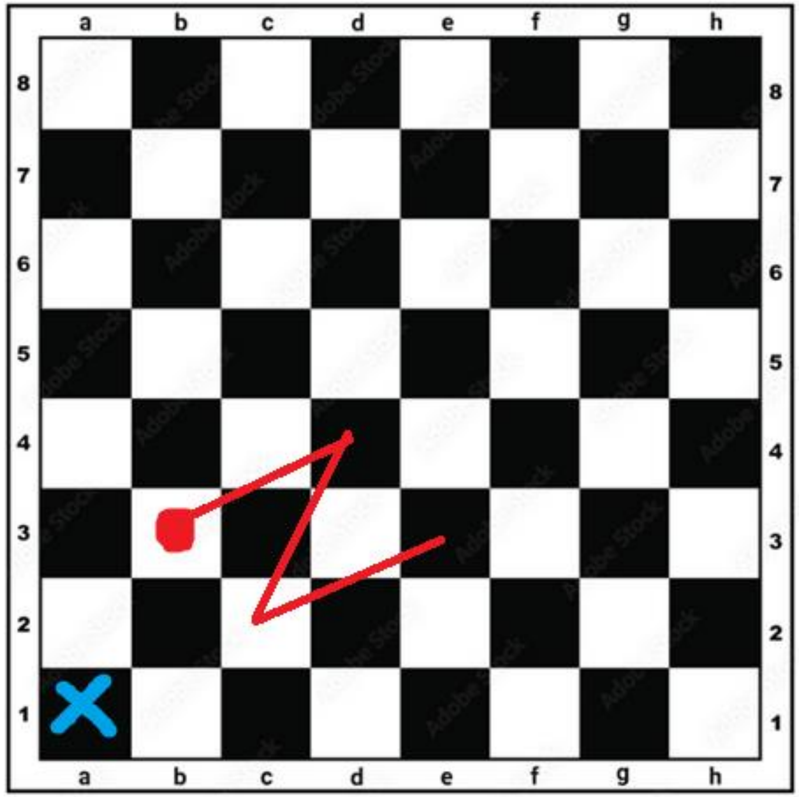
$$x = \left\lfloor \frac{sq}{n+2} \right\rfloor = \frac{sq}{n+2}$$

$$y = sq \ \% \ (n+2) - 1$$

Warnsdorffovo pravidlo

- Heuristické pravidlo, které se snaží zabránit odříznutí nějakého políčka
- Pravidlo: vždy volíme tah na pole, s nejmenším stupněm. (Pole, z kterého máme nejmenší možný počet legálních tahů)





Procházka koněm metodou Rozděl a panuj

- 1) ROZDĚL – rozděl šachovnici na 4 podšachovnice
- 2) VYŘEŠ – Najdi v každé podšachovnici Hamiltonův cyklus
- 3) SPOJ – spoj podšachovnice

- Hamiltonův cyklus – cyklus, který navštíví každé pole (vrchol grafu) končí v místě kde začíná

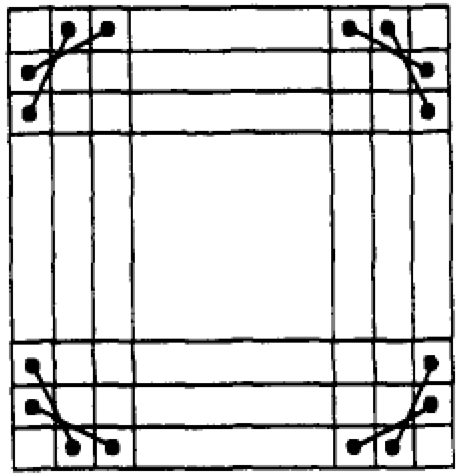


Fig. 1. Required moves for a structured knight's tour.

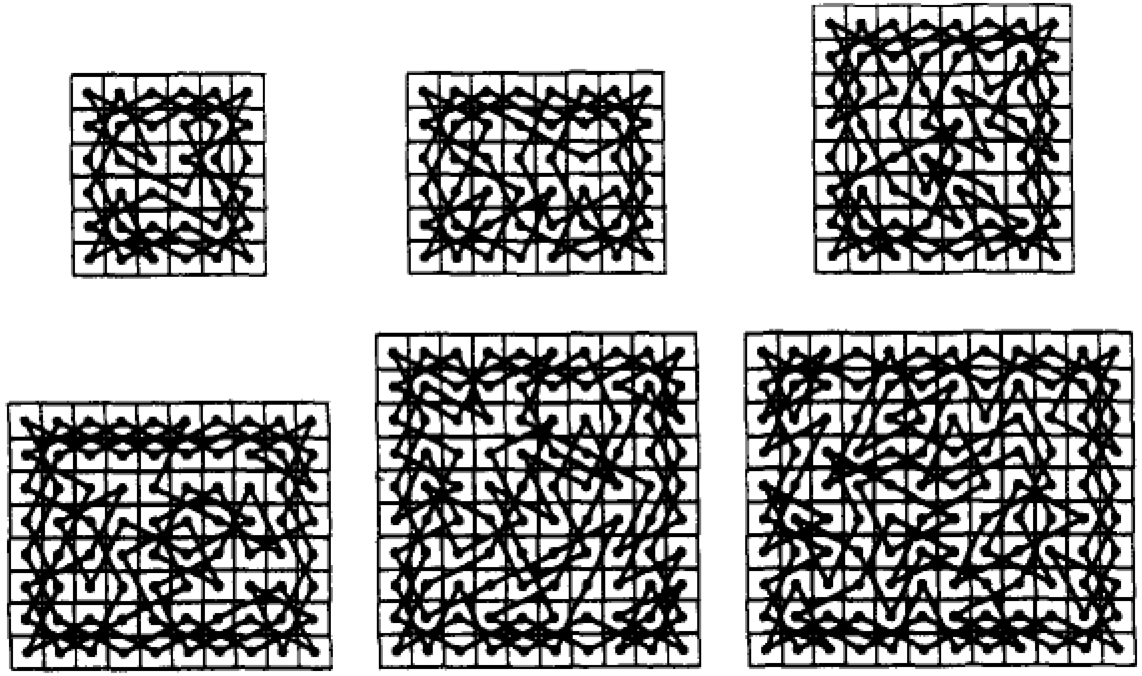


Fig. 2. Structured knight's tours for (in row-major order) 6×6, 6×8, 8×8, 8×10, 10×10, and 10×12 boards.

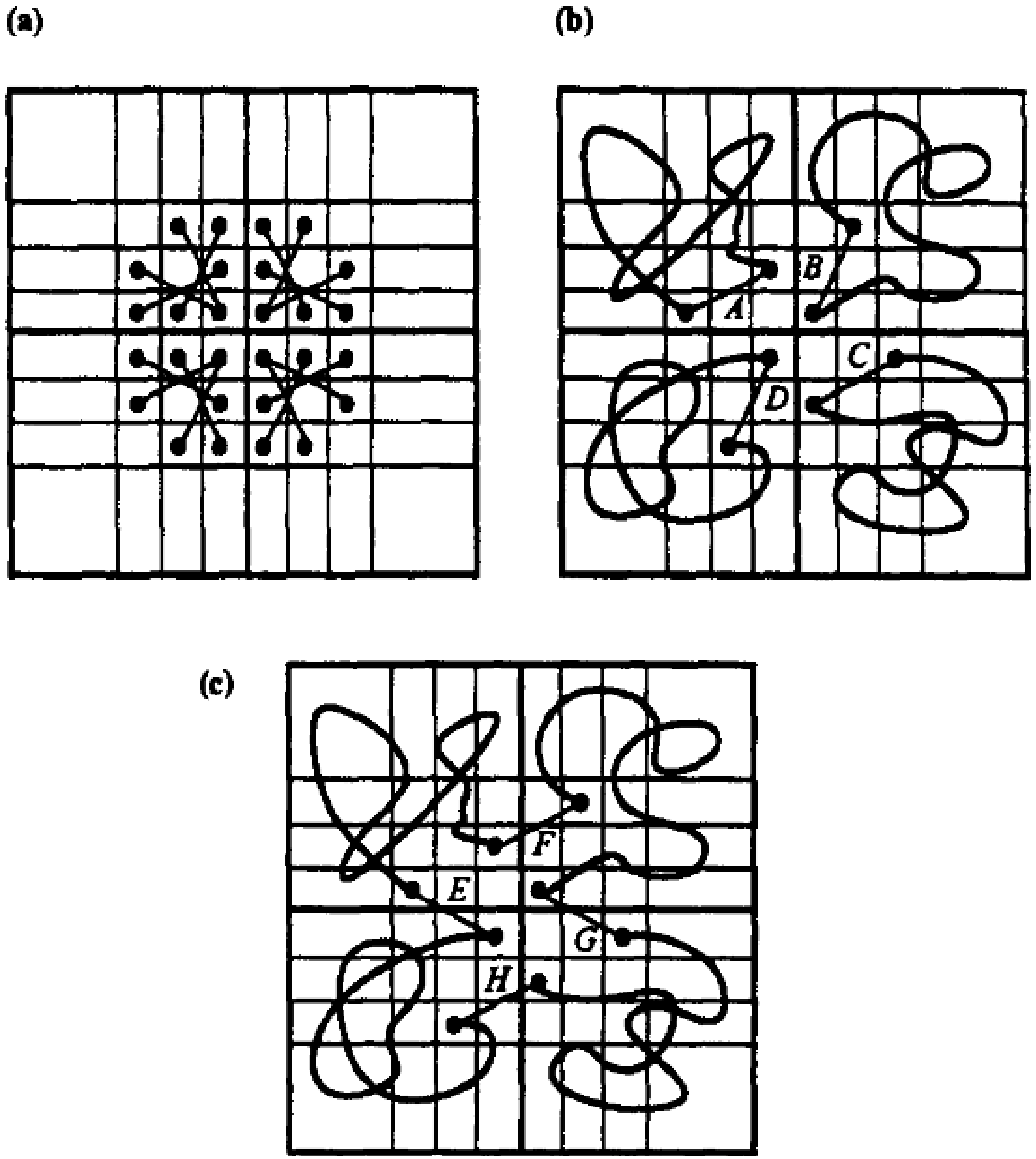


Fig. 3. How to combine four structured knight's tours into one: (a) the moves at the inside corners, (b) the edges A, B, C, D to be deleted, and (c) the replacement edges E, F, G, H.

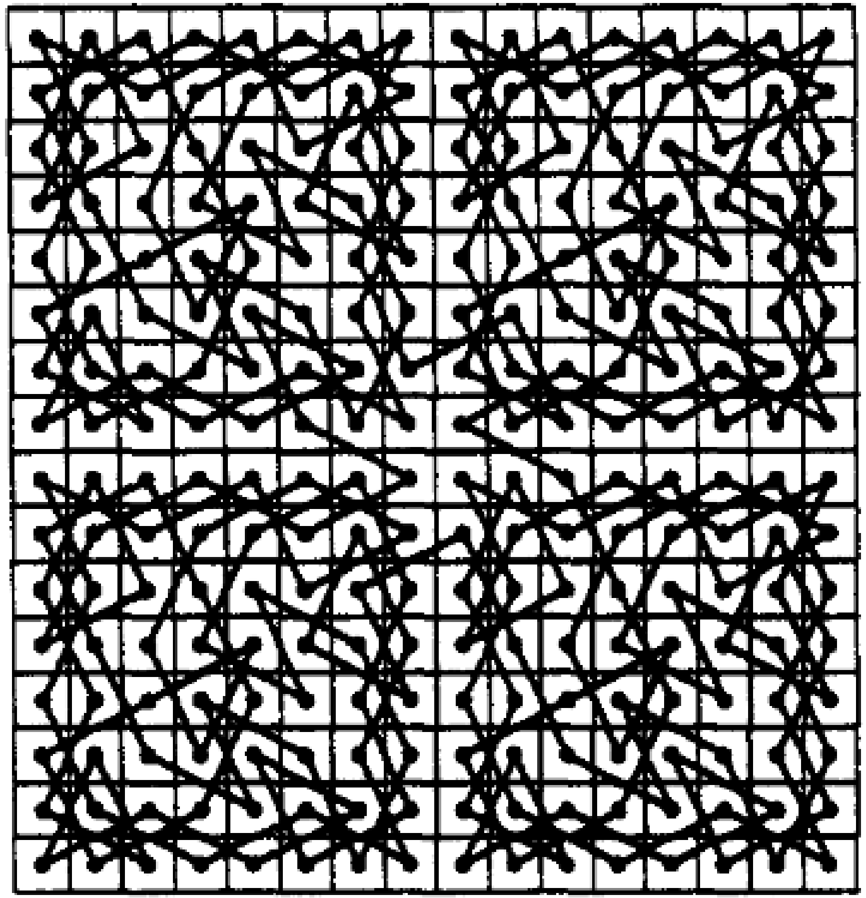
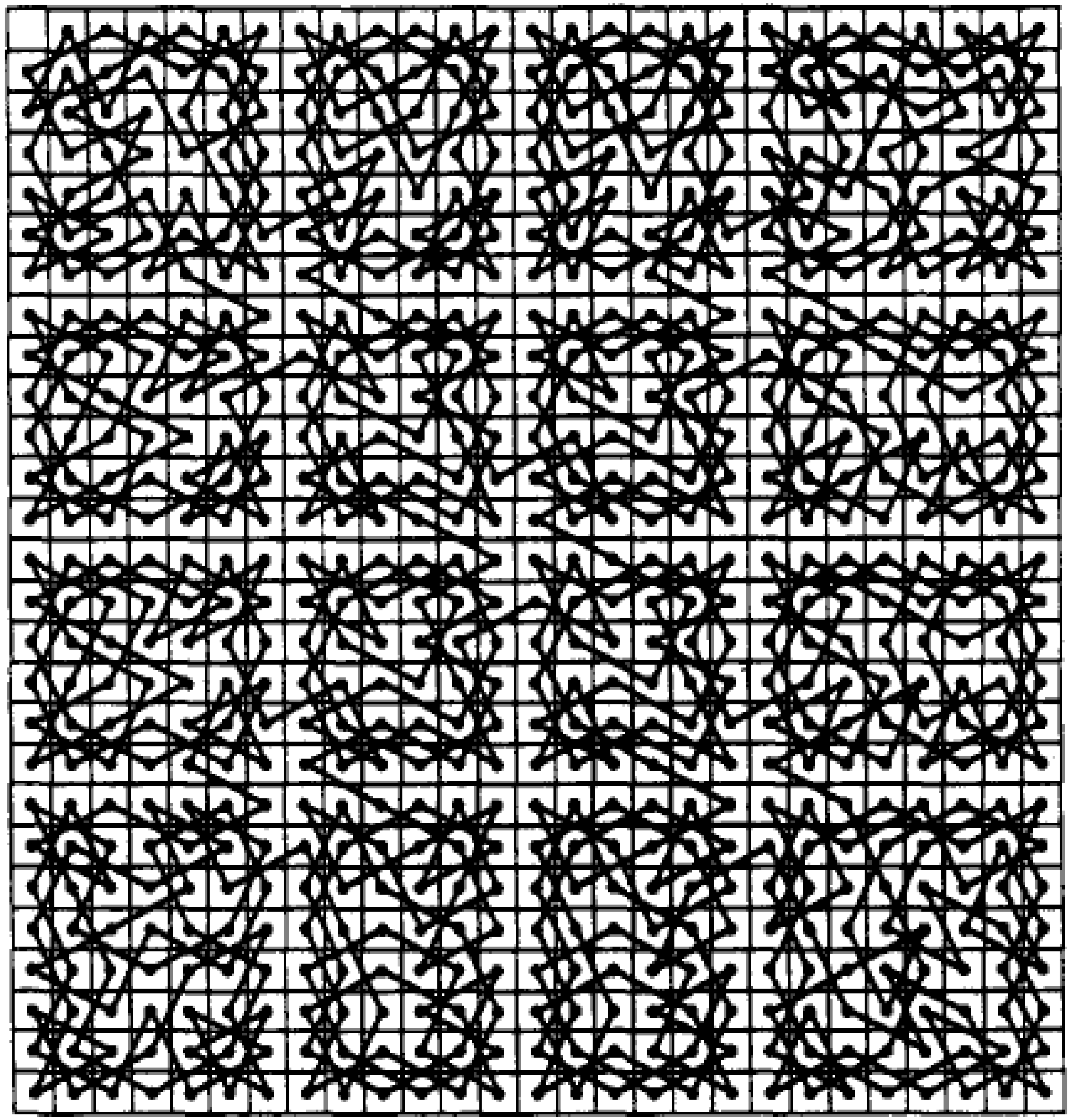


Fig. 4. A 16×16 knight's tour constructed from the 8×8 knight's tour in Fig. 2 using the technique of Theorem 2.1.



Dobrovolný domácí úkol

- Nejkratší cesta jezdcem
- Nalezněte nejkratší cestu jezdcem z pole [x][y] do [z][w] a cestu vypište.

- Metoda – procházení do šířky (metoda vlny). (Vhodné použít frontu)