Magic Squares with Heuristic Methods

Kan-Anek Tantitayapong

Mahidol University International College

30 Nov 2023

Outline

Preliminaries

Definition (Semi-magic square)

A semi-magic square of order n is an $n \times n$ matrix M with entries $a_{ij} \in \{1, 2, \ldots, n^2\}$ such that

$$\sum_{j=1} a_{ij} = \sum_{i=1} a_{ij} = \frac{n(n^2+1)}{2}$$

Preliminaries

Definition (Magic square)

A magic square of order n is an $n \times n$ matrix M with entries $a_{ij} \in \{1, 2, \ldots, n^2\}$ such that

$$\sum_{j=1} a_{ij} = \sum_{i=1} a_{ij} = \sum_{i=1} a_{ii} = \sum_{j=1} a_{i,\,n-i+1} = \frac{n(n^2+1)}{2}$$

Theorem

For every natural number $n \geq 3$, there exists a magic square of order n.

Example

An example of a 3×3 magic square.

4	9	2
3	5	7
8	1	6

The rows, columns, and diagonals add up to $\frac{3(3^2+1)}{2} = 15$.

Deriving the sum

To find out what each row, column and diagonal should add up to, we first consider the sum of the entire square. That is, the sum of the numbers 1 to n^2, which is given by

$$S_{\text{total}} = \frac{n^2(n^2+1)}{2}$$

Now, we look at this as the sum of n rows. With that perspective, we divide by n and we have that the sum of each row is

$$S_{\text{row}} = \frac{n(n^2+1)}{2}$$

Why Heuristic Methods?

- The search space for the magic square problem of order n is $(n^2)!$.
- Exhaustive search can be done for (very) small n, but becomes impractical for larger n. To give you an idea, the search space for a 5×5 magic square is 1.55×10^{25}, which is 100 times the number of stars in the observable universe.
- Heuristic algorithms allow us to cleverly traverse the search space to find (feasible) solutions.

Outline

Procedure

Steps (Local Search Family)

1. Find a semi-magic square of order n using the cost function

$$\text{cost}_1(H) = \sum_{i=1}^{n} |S - \sum_{j=1}^{n} a_{ij}| + \sum_{j=1}^{n} |S - \sum_{i=1}^{n} a_{ij}|$$

2. Take the solution and only make moves that preserve row and column sums using the cost function

$$\text{cost}_2(H) = |S - \sum_{i=1}^{n} a_{ii}| + |S - \sum_{i=1}^{n} a_{i,n-i+1}|$$

Framework

Solution Representation

1-d array of length n^2 representing the entries read row by row.

Perturbation Methods

1. Random Swap: pick two elements at random and swap them.
2. Adjacent Swap: pick two adjacent elements and swap them.

Testing

- (for algorithms in the local search family) 100 trials, 10,000 iterations each step 1. 10,000 more iterations for step 2.
- 10,000 generations for Genetic Algorithm.
- Increase the order until we can't find solutions.

How to Preserve Row and Column Sums?

Swap pairs of elements based on the following rules:

1. Elements a_{ij} and a_{il} are swapped with a_{kj} and a_{kl} respectively if
$$a_{ij} + a_{il} = a_{kj} + a_{kl}$$
2. Elements a_{ij} and a_{kj} are swapped with a_{il} and a_{kl} respectively if
$$a_{ij} + a_{kj} = a_{il} + a_{kl}$$

Example

Consider the following 4×4 semi-magic square

$$H = \begin{array}{|c|c|c|c|} \hline 13 & 4 & 11 & 6 \\ \hline 12 & 1 & 5 & 16 \\ \hline 7 & 14 & 10 & 3 \\ \hline 2 & 15 & 8 & 6 \\ \hline \end{array}$$

We swap 1 with 5 and 14 with 10 since $14 + 1 = 10 + 5 = 15$.

$$H^* = \begin{array}{|c|c|c|c|} \hline 13 & 4 & 11 & 6 \\ \hline 12 & 5 & 1 & 16 \\ \hline 7 & 10 & 14 & 3 \\ \hline 2 & 15 & 8 & 6 \\ \hline \end{array}$$

Outline

Algorithm Details

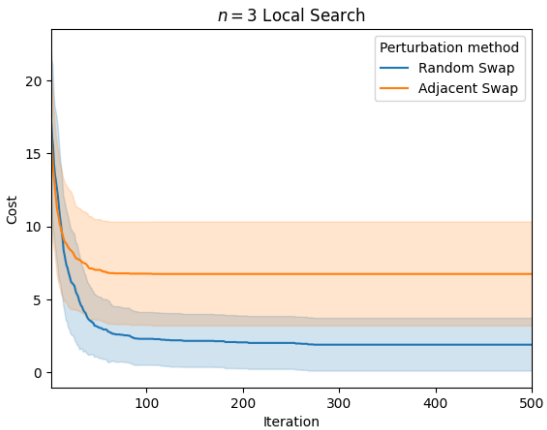
Simulated Annealing

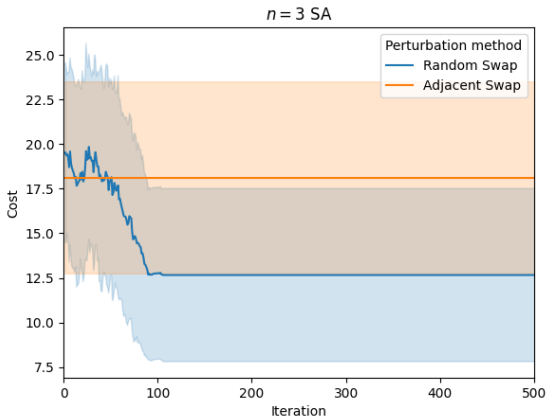
- $T_0 = 100$ for $n = 3$, $T_0 = 500$ for $n = 4$, $T_0 = 1000$ for $n = 5$ and $n = 6$.
- Initial temperatures found by experimenting with various values and looking at the acceptance rate.
- Fixed cooling rate $\alpha = 0.95$

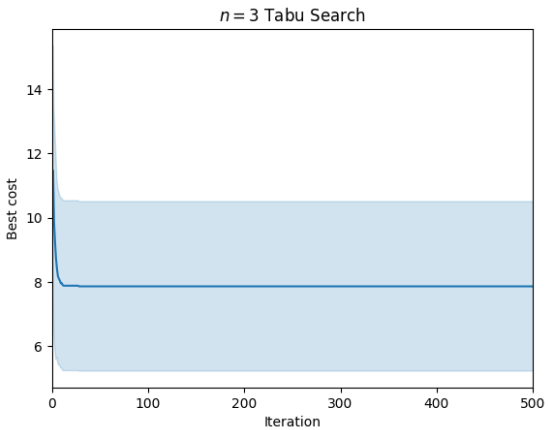
Tabu Search

- $|V^*| = 10$
- $k = 4$
- Attribute stored: pair of swapped elements.
- Punish worse solutions based on frequency every 30 iterations

$n = 3$







Convergence percentages: 44%, 0%, 0%

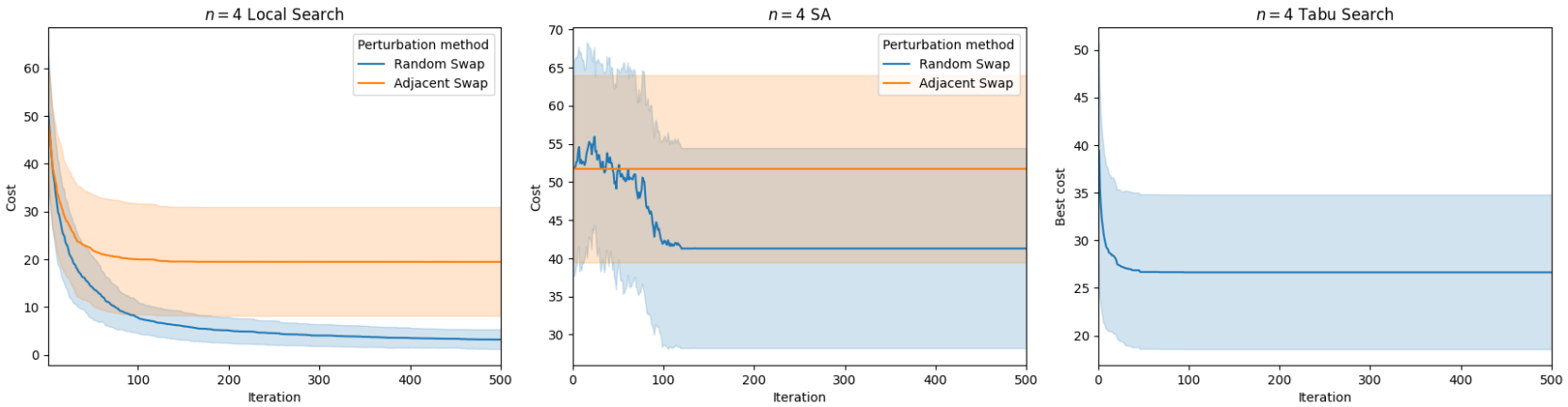
$n = 3$

Solution found (Local Search):

2	9	4
7	5	3
6	1	8

$S = 15$

$n = 4$



Convergence percentages: 13%, 0%, 0%

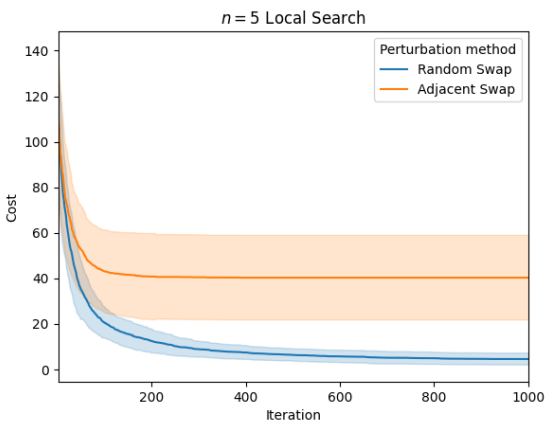
$n = 4$

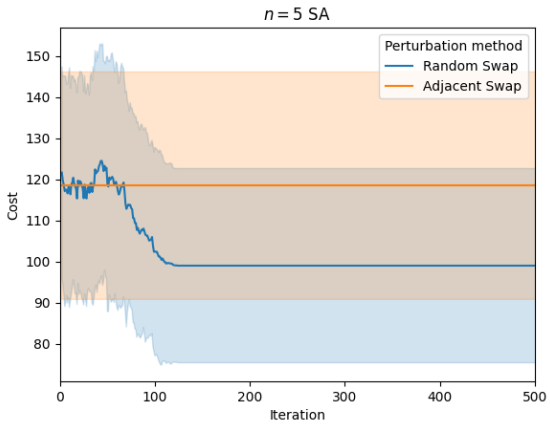
Solution found (Local Search):

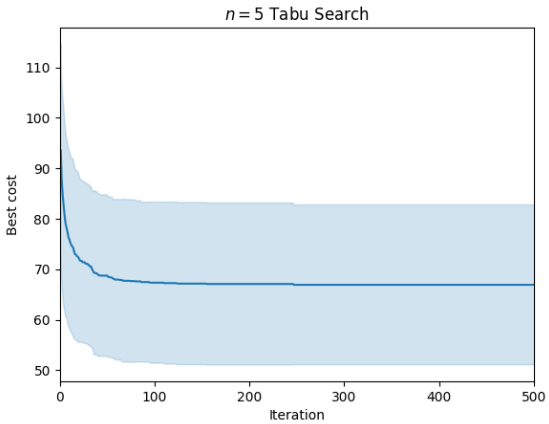
15	5	4	10
2	12	13	7
14	18	1	11
3	9	16	6

$S = 34$

$n = 5$







Convergence percentages: 5%, 0%, 0%

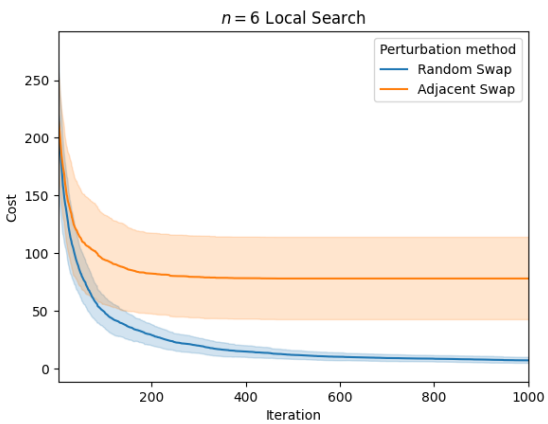
$n = 5$

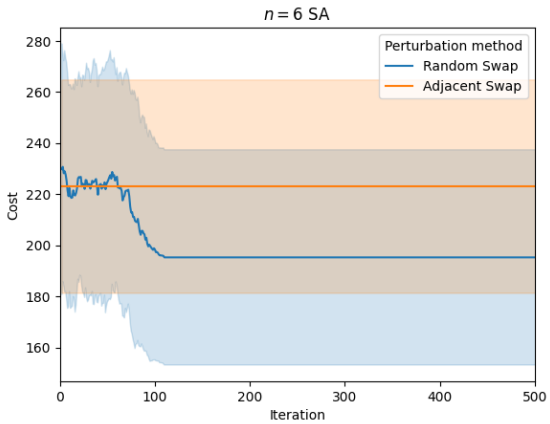
Solution found (Local Search):

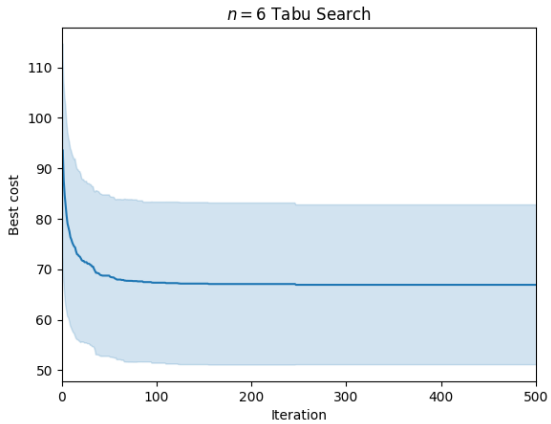
18	25	2	9	11
4	6	13	20	22
15	17	24	1	8
21	3	10	12	19
7	14	16	23	5

$S = 65$

$n = 6$







Convergence percentages: 1%, 0%, 0%

$n = 6$

Solution found (Local Search):

1	2	3	34	35	36
4	30	27	20	11	19
10	25	29	6	26	15
31	13	8	28	9	22
32	17	23	7	18	14
33	24	21	16	12	5

$S = 111$

Observations

- SA and TS didn't manage to find any semi-magic squares and hence it didn't get past the first phase.
- Local Search managed to find semi magic squares up until $n = 9$. All of the squares presented above were found by chance at the first phase.
- 10,000 more iterations on the second phase didn't improve the solutions at all.
    - Need a smarter way to find pairs to swap.

Outline

Algorithm Details

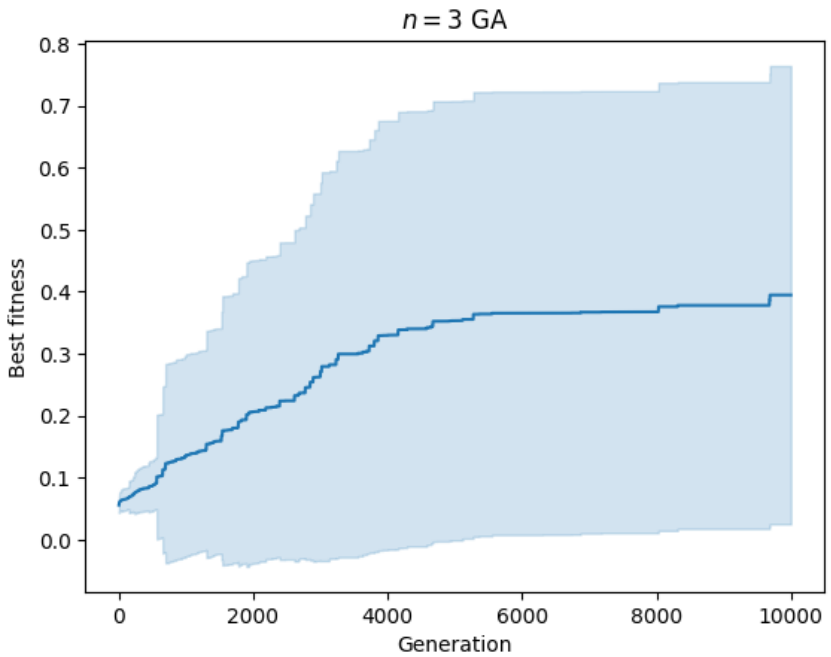
Genetic Algorithm

- Representation: string of n^2 numbers (e.g. '782456137')
- Population size $= 10$
- PMX for crossover
- Random Swap for mutation
- $p_c = 1$ and $p_\mu = 0.03$
- Fitness function:

$$f(H) = \frac{1}{1 + \text{cost}(H)}$$

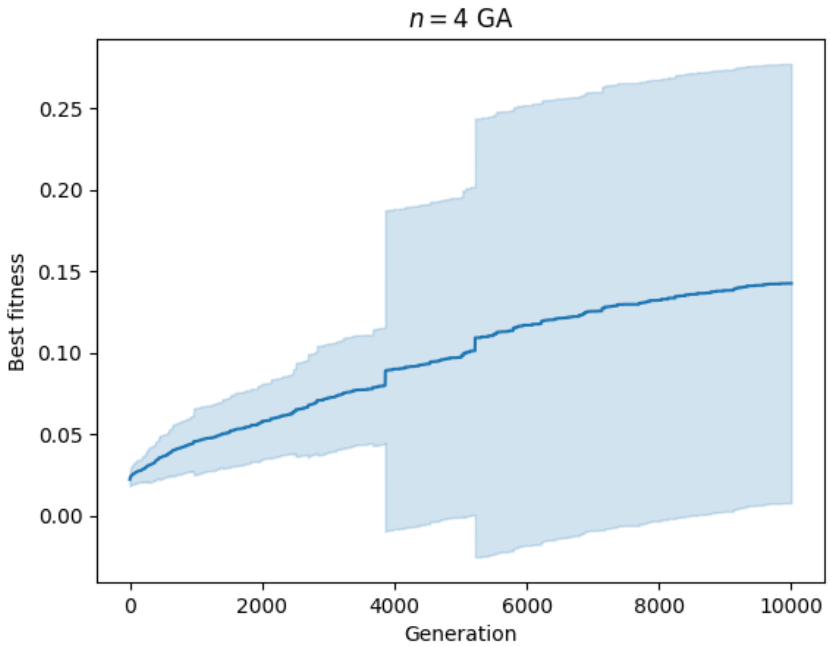
where $\text{cost}(H) = \text{cost}_1(H) + \text{cost}_2(H)$

$n = 3$



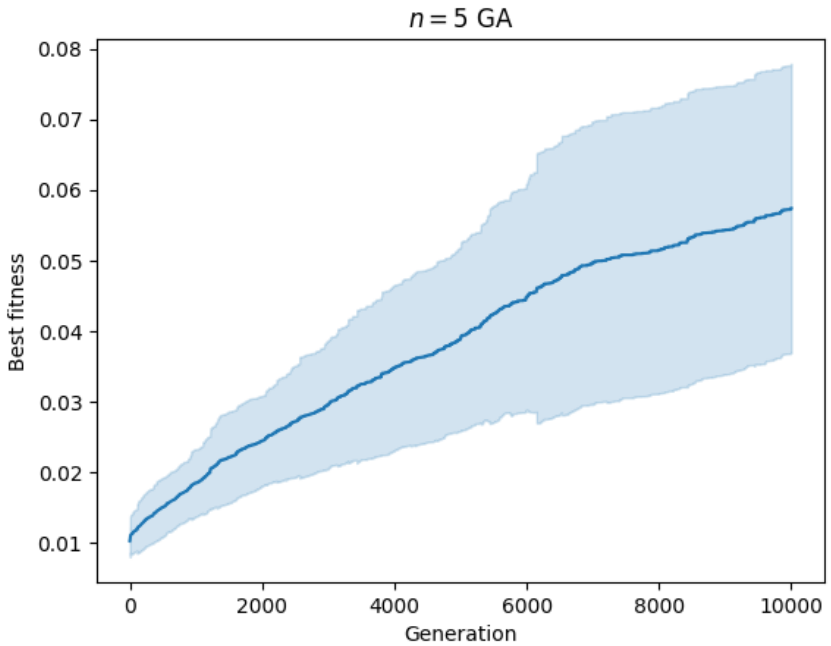
Convergence percentage: 26%

$n = 4$



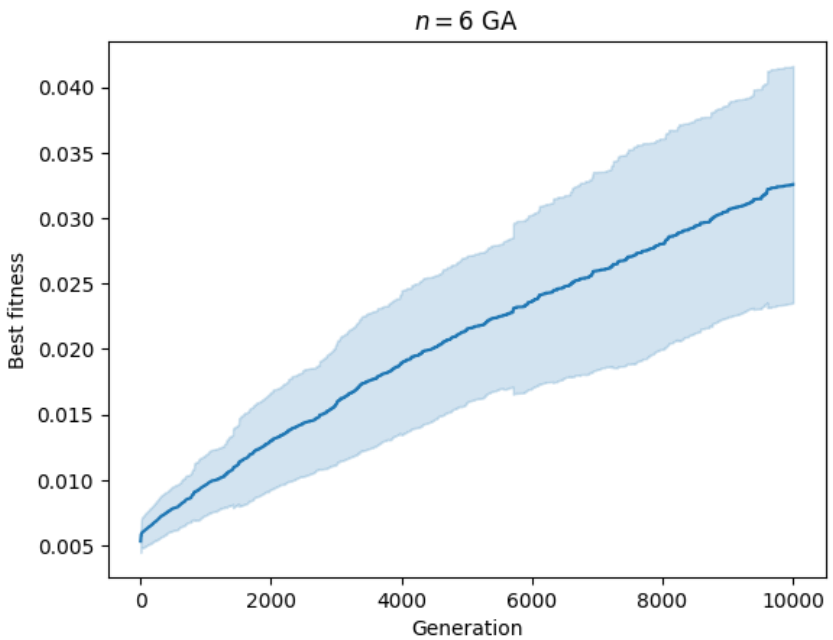
Convergence percentage: 2%

$n = 5$



Convergence percentage: 0%

$n = 6$



Convergence percentage: 0%

Outline

How did we do?

- Managed to find magic squares up to $n = 6$. Not too bad, but could be a lot better.
- SA and TS didn't work as expected (may need more iterations and/or tuning of parameters)
- We probably need a smarter way to decide which elements to swap, not just randomly choosing. For example, choosing two elements from the same row so the column sum is not changed.

References

1. E. Cruz and E. Grandchamp. Heuristic method to find magic squares. pages 119–123, 12 2012.
2. O. Cain. Gaussian integers, rings, finite fields, and the magic square of squares, 2019.