# Room Layout Optimization: A Comparative Study of Simulated Annealing and Exact Methods

Zelin Li
Software Innovation Class 3B 511172176
Department of Computer Science and Information Engineering
Fu Jen Catholic University

Abstract—In limited indoor spaces, automated room layout optimization can assist designers in quickly finding high-quality layout solutions. This study compares the performance of Simulated Annealing (SA) and Exact Solver methods in the room layout optimization problem. We selected four different scale room cases (3×3m, 5×4m, 2.5×2m, 6×5m) for experiments, analyzing their runtime and energy (quality) performance. The results show that: Exact Solver can quickly obtain the optimal solution in small-scale problems, and SA can also achieve solutions of the same quality but takes longer; in medium to large-scale problems, Exact Solver fails to find a solution (N/A), while SA, although time-consuming, can obtain feasible solutions. Additionally, we provide integrated energy and time comparison graphs to clearly present the advantages and disadvantages of both methods.

Keywords—Simulated Annealing, Exact Solver, Room Layout, Space Optimization, Automated Design

## I. Introduction

In interior design, the rational arrangement of furniture is crucial for space utilization and comfort. Traditional design relies on experience, making it difficult to ensure global optimality. By transforming the problem into a combinatorial optimization and applying advanced algorithms, we can find high-quality solutions within controllable time frames.

This study compares the applicability of Simulated Annealing (SA) and Exact Solver in room layout optimization problems. SA can find approximate solutions in large-scale problems [1], while Exact Solver can guarantee optimal solutions for small problems but is unsuitable for medium to large-scale problems [2].

## II. Research Background and Methods

### A. Problem Definition and Energy Mechanism

Given a room $R(w,h)$ and a set of furniture $F$. Each furniture has fixed dimensions, possible placement positions, and rotation directions (0° or 90°). To measure the layout quality, the energy function is defined as:

$$\min E = \sum_{i=1}^{n} \sum_{j=i+1}^{n} O_{ij} + \sum_{i=1}^{n} B_i, \tag{1}$$

$$\begin{aligned} O_{ij} = {} & \max\left(0, \min(x_i + w_i, x_j + w_j) - \max(x_i, x_j)\right) \\ & \times \max\left(0, \min(y_i + h_i, y_j + h_j) - \max(y_i, y_j)\right), \end{aligned} \tag{2}$$

$$B_i = \max\left(0, x_i + w_i - R_w\right) + \max\left(0, y_i + h_i - R_h\right). \tag{3}$$

Where:

- $O_{ij}$ is the overlapping area between furniture $i$ and furniture $j$. When $O_{ij} > 0$, it indicates that the two furniture pieces overlap.
- $B_i$ is the boundary penalty for furniture $i$, ensuring that the furniture does not exceed the room boundaries. The lower the energy, the better the layout quality.

### B. Simulated Annealing (SA) Pseudocode

**Algorithm 1** Simulated Annealing (SA) Pseudocode

Generate initial solution $S$
$T = T_0$ (initial temperature)
while $T > T_{\min}$ do
  for $k = 1$ to $L$ do
    $S' = \text{GenerateNeighbor}(S)$
    $\Delta E = E(S') - E(S)$
    if $\Delta E < 0$ or $e^{-\frac{\Delta E}{T}} > \text{rand}(0,1)$ then
      $S \leftarrow S'$
    end if
  end for
  $T \leftarrow \alpha \cdot T$
end while
return $S$

SA allows worse solutions to be accepted in high-temperature stages to escape local optima and gradually converges to the optimal solution as the temperature decreases.

### C. Mathematical Description of Simulated Annealing (SA)

Simulated Annealing is a heuristic search algorithm used to find approximate global optimal solutions in large-scale search spaces.

Probability of Accepting a New State:

$$P = \begin{cases} 1 & \Delta E \leq 0, \\ e^{-\frac{\Delta E}{T}} & \Delta E > 0. \end{cases}$$

Where:

- $\Delta E = E_{\text{new}} - E_{\text{current}}$ is the energy change of the new layout.
- $T$ is the current temperature, gradually decreasing with iterations.

Cooling Schedule:

$$T_{\text{new}} = \alpha \cdot T_{\text{current}}$$

Where:

- $\alpha$ is the cooling rate, set to 0.85 in the program.
- $T_{\text{initial}}$ is the initial temperature, set to 5000 in the program.

### D. Exact Solver Pseudocode (Branch and Bound)

**Algorithm 2** Exact Solver (Branch and Bound) Pseudocode

```
Initialize search tree node (empty layout), BestE = ∞
while Queue not empty do
    Node = ExtractMin(Queue)
    if Node lower bound > BestE then
        Prune this node
    else
        Expand node: place next furniture
        for each feasible placement do
            Calculate E or lower bound
            if All furniture placed then
                if E < BestE then
                    BestE = E, update solution
                end if
            else
                Insert child node
            end if
        end for
    end if
end while
return BestSolution
```

Exact Solver can quickly find the optimal solution for small problems, but in medium to large-scale problems, the computational complexity grows exponentially, making it unsolvable (N/A). The complete pseudocode is placed at the end of the document.

### E. Mathematical Description of Exact Solver

Brute-force search is an algorithm that enumerates all possible layouts to find the global optimal solution.

$$S = \prod_{i=1}^{N} P_i$$

Where:

- $N$ is the total number of furniture.
- $P_i$ is the number of possible placement and rotation combinations for furniture $i$.

Description:

- The number of permutations for each furniture is determined by the step length (e.g., 0.25 m) and room size.
- By traversing $S$, the layout with the minimum energy is found. When $E = 0$, it is the optimal solution.

## III. Experimental Design

This study tests four cases:

- CASE1: 3×3m (small-scale)
- CASE2: 5×4m (medium-scale)
- CASE3: 2.5×2m (small-scale)
- CASE4: 6×5m (large-scale)

Comparing the final energy and runtime of SA and Exact Solver.

## IV. Results and Analysis

The optimal layout diagrams for each case are presented (one for Exact Solver and one for SA, some cases only have SA diagrams), followed by integrated energy and time comparison graphs, and finally summarized in a table.

### A. CASE1 (3×3m)

Small-scale. Exact Solver can instantly achieve energy 0, and SA can also achieve 0 but takes longer.

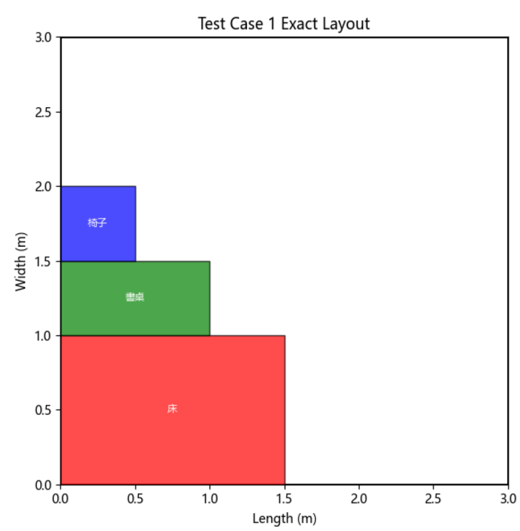


Fig. 1. CASE1 Optimal Layout using Exact Solver

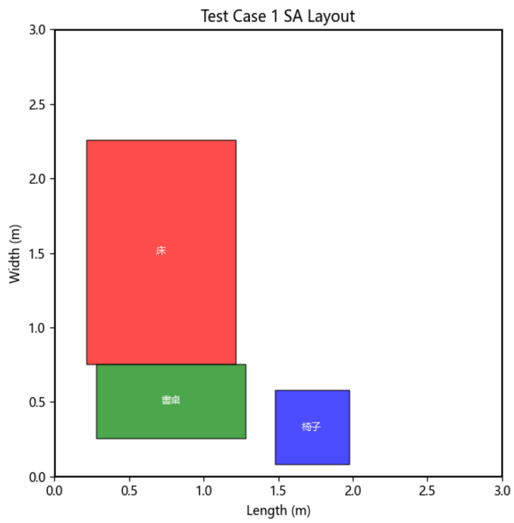


Fig. 2. CASE1 Optimal Layout using SA

B. CASE2 (5×4m)

Medium-scale. Exact Solver fails to find a solution (inf), while SA obtains an energy 0 solution in approximately 8.7276 seconds.

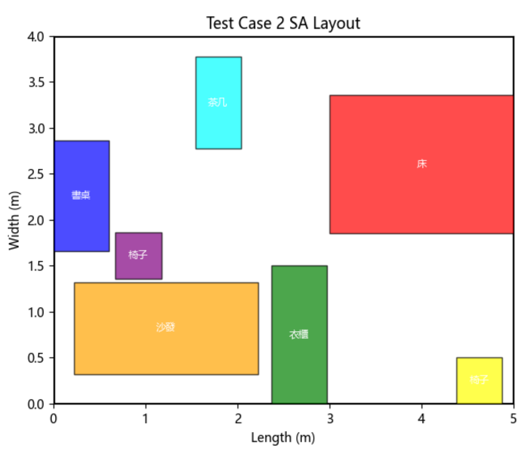


Fig. 3. CASE2 Optimal Layout using SA

C. CASE3 (2.5×2m)

Small-scale similar to CASE1. Exact Solver quickly achieves energy 0, and SA can as well but takes longer.

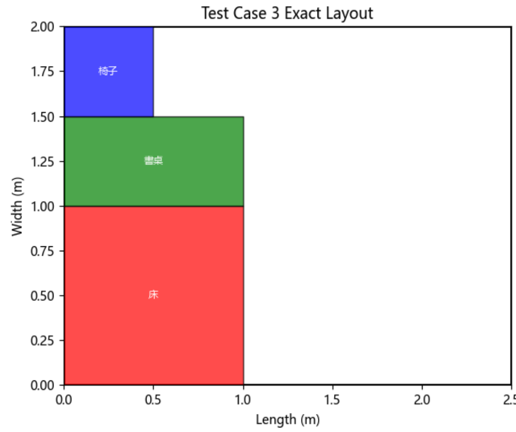


Fig. 4. CASE3 Optimal Layout using Exact Solver

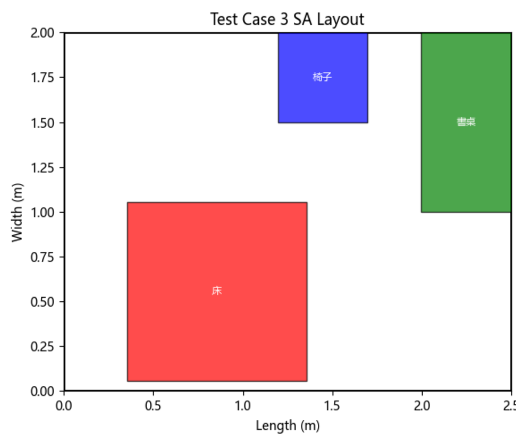


Fig. 5. CASE3 Optimal Layout using SA

D. CASE4 (6×5m)

Large-scale. Exact Solver cannot find a solution (N/A), while SA obtains an energy 0.39 solution in approximately 16.3204 seconds.

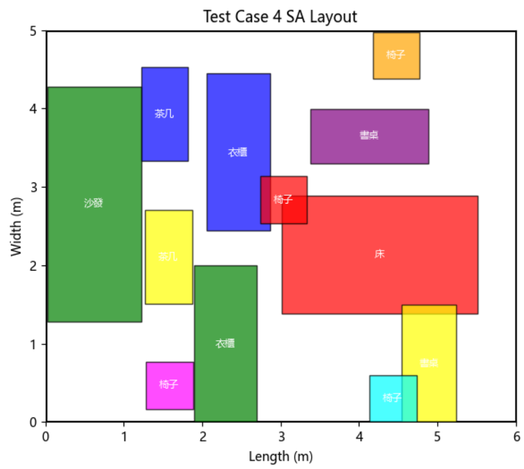


Fig. 6. CASE4 Optimal Layout using SA

## E. Overall Energy Comparison

The following is the energy comparison graph between SA and Exact Solver:

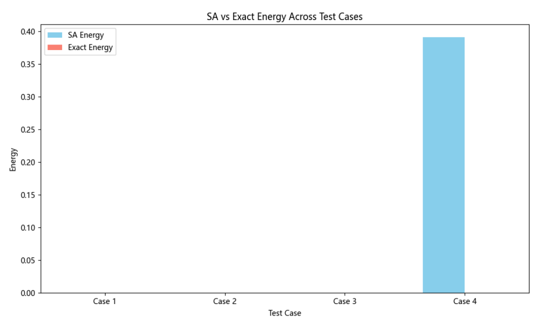


Fig. 7. Overall Energy Comparison: Small-scale Exact Solver and SA both achieve 0; Medium to large-scale Exact Solver N/A, SA has feasible solutions

## F. Overall Time Comparison

The following is the time comparison graph between SA and Exact Solver:

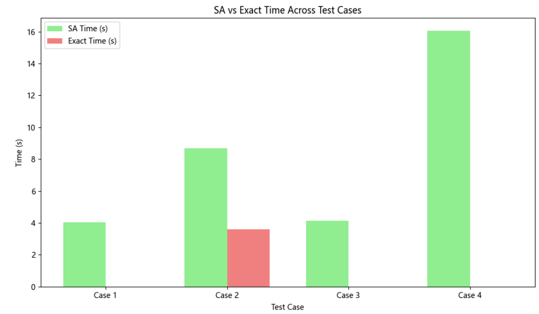


Fig. 8. Overall Time Comparison: Exact Solver is extremely fast for small problems, SA takes longer; Exact Solver is N/A for medium to large-scale problems, SA takes longer but has solutions

## G. Summary of Results in Table

Table I summarizes the results of the four cases.

TABLE I
Summary of Results for Four Cases

| Test Case | Room Size | SA Energy | SA Time(s) | Exact Energy | Exact Time(s) |
|---|---|---|---|---|---|
| 1 | 3x3m | 0.00 | 4.0218s | 0.00 | 0.0050s |
| 2 | 5x4m | 0.00 | 8.6686s | inf | 3.5944s |
| 3 | 2.5x2m | 0.00 | 4.1367s | 0.00 | 0.0029s |
| 4 | 6x5m | 0.39 | 16.0464s | N/A | N/A |

# V. Conclusion and Future Work

This study compares the performance of SA and Exact Solver in room layout optimization problems. The results show:

- Small-scale problems: Exact Solver can quickly obtain the optimal solution with energy 0, and SA can also achieve 0 but takes longer.
- Medium to large-scale problems: Exact Solver fails to solve (N/A), while SA, although time-consuming, can obtain feasible solutions.

Future work may consider hybrid strategies (such as using Exact Solver for local solutions first, then using SA for global optimization) or adding practical constraints to make the results more aligned with real design requirements.

# References

[1] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

[2] Woeginger, G. J. (2003). Exact algorithms for NP-hard problems: A survey. In Jünger, M., Reinelt, G., & Rinaldi, G. (Eds.), Combinatorial Optimization—Eureka, You Shrink! (Lecture Notes in Computer Science, vol. 2570). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36478-1_17