Artificial Intelligence Lab Report 4

Anuj Saha
202351010
BTech CSE
IIIT, Vadodara

G. Nikhil
202351037
BTech CSE
IIIT, Vadodara

Divyanshu Ghosh
202351036
BTech CSE
IIIT, Vadodara

***Abstract*—This report presents the application of simulated annealing to two distinct optimization problems: jigsaw puzzle reconstruction and North Indian classical music melody generation. For the jigsaw puzzle problem, we formulate the reconstruction of a 512×512 grayscale image divided into 16 tiles as a state space search problem, incorporating tile transformations including rotations and flips. The simulated annealing algorithm successfully reconstructed the scrambled Lena image with a final energy of 189,274.75 after 1,200,000 iterations. For the challenge problem, we generated melodic sequences following Raag Bhairav grammar using simulated annealing, achieving a melody with energy 30.0 that incorporates the characteristic pakad phrase and adheres to the modal constraints. The results demonstrate the effectiveness of simulated annealing for large-scale combinatorial optimization problems with complex constraint structures. Complete code implementation is available at: https://github.com/AnujSaha0111/CS307-Lab-Submissions/tree/main/Submission_4.**

***Index Terms*—simulated annealing, jigsaw puzzle, combinatorial optimization, melody generation, Raag Bhairav, state space search**

I. Introduction

Simulated annealing is a probabilistic metaheuristic employed to solve global optimization problems, inspired by the physical process of annealing used in metallurgy. The nature of the algorithm is to accept moves both improving and deteriorating the current solution with a probability whose value is a function of time, thus enabling the search to escape from local optimal solutions. This property makes simulated annealing especially suitable for solving combinatorial optimization problems with large search spaces, where exhaustive search is impractical due to computational complexity.

This laboratory exercise uses simulated annealing on two separate problem domains. The main task involves solving a jigsaw puzzle by reconstructing a scrambled 512×512 pixel grayscale image of 16 tiles, each of which can have eight transformations applied to it. The challenge problem extends this technique to generative tasks by constructing melodic sequences which conform to the grammatical rules of Raag Bhairav, which is a morning raga in Hindustani classical music.

Both problems have similar properties that allow simulated annealing to solve them: large discrete solution spaces, non-convex energy landscapes with many local minima, and complicated structures of constraints that avoid analytical solution. The jigsaw puzzle has constraints based on the spatial arrangement of the pieces and edges that are compatible with one another, and the melody generation problem required modal rules and specific phrases to be included in the music.

II. Problem Formulation

A. Jigsaw Puzzle as State Space Search

The jigsaw puzzle reconstruction problem is formulated as a state space search with the following components:

States: A state S is represented as a 4×4 grid where each cell contains an integer index $v \in \{0, 1, ..., 127\}$ representing one of 128 transformed tiles. The state space size is $(16!)$ permutations $\times 8^{16}$ transformations = 2.58×10^{33} states.

Initial State: S_0 consists of tiles arranged in reading order with identity transformations applied, where tile at grid position (i, j) receives index $(4i + j) \times 8$.

Goal State: Any configuration S_g where the energy function $E(S_g)$ is minimized, ideally approaching zero for perfect reconstruction.

Actions: The action set includes four move types with associated probabilities:

- Swap positions (0.5): Exchange tiles between two grid positions
- Swap transformations (0.15): Exchange transformation types between tiles
- Change transformation (0.25): Apply a random transformation to a tile
- Rotate transformation (0.10): Incrementally rotate a tile

Transition Model: Given state S and action a, the successor state $S' = \text{APPLY}(S, a)$ modifies the grid according to the selected move type.

Energy Function: The optimization objective combines pixel difference and gradient discontinuity measures:

$$E(S) = \sum_{(i,j)\in\mathcal{E}} (D_{\text{pixel}}(i, j) + 0.5 \cdot D_{\text{grad}}(i, j)) \qquad (1)$$

where $\mathcal{E}$ denotes all adjacent tile pairs, D_{pixel} measures absolute intensity differences at boundaries with width $w = 2$, and D_{grad} captures gradient discontinuities.

For horizontal adjacency:

$$D_{\text{pixel}}(i, j) = \sum_{r=0}^{127} \sum_{c=126}^{127} |T_{i,j}[r, c] - T_{i,j+1}[r, c - 126]| \qquad (2)$$

where $T_{i,j}$ denotes the tile at grid position (i, j). Similar formulations apply for vertical boundaries and gradient terms.

B. Raag Bhairav Melody Generation

The melody generation problem is formulated with the following structure:

States: A melody M is a sequence of length 16 drawn from the note set $\mathcal{N} = \{\text{Sa}, \text{Re(b)}, \text{Ga}, \text{Ma}, \text{Pa}, \text{Dha(b)}, \text{Ni}, \text{Sa'}\}$.

Initial State: M_0 begins with the pakad phrase $[\text{Re(b)}, \text{Sa}, \text{Pa}, \text{Ma}, \text{Re(b)}]$ followed by random notes.

Goal State: Any melody M_g satisfying Raag Bhairav constraints with minimal energy.

Actions: Single-note substitution at a random position.

Energy Function: The objective balances melodic smoothness, phrase diversity, and grammatical correctness:

$$E(M) = \sum_{i=1}^{15} |\text{idx}(M_i) - \text{idx}(M_{i+1})|^2 + 20 \cdot N_{\text{rep}} + 10000 \cdot \mathbb{I}_{\text{pakad}} \tag{3}$$

where N_{rep} counts consecutive triple repetitions and $\mathbb{I}_{\text{pakad}}$ is an indicator function taking value 1 if the pakad phrase is absent.

III. Methodology

A. Simulated Annealing Algorithm

The core optimization procedure follows the Metropolis-Hastings acceptance criterion with geometric cooling:

Algorithm 1 Simulated Annealing

Require: Initial state S_0, initial temperature T_0, cooling rate α, minimum temperature $T_{\min}$

Ensure: Best state S_{best}

```
S_current ← S_0, T ← T_0
E_current ← E(S_current)
S_best ← S_current, E_best ← E_current
while T > T_min and iterations remain do
    S_new ← NEIGHBOR(S_current)
    E_new ← E(S_new)
    ΔE ← E_new − E_current
    if ΔE ≤ 0 or RANDOM() < e^(−ΔE/T) then
        S_current ← S_new
        E_current ← E_new
        if E_new < E_best then
            S_best ← S_new
            E_best ← E_new
        end if
    end if
    T ← α · T
end while
return S_best =0
```

B. Parameter Configuration

Jigsaw Puzzle Parameters:

- Initial temperature: $T_0 = 5000.0$
- Cooling rate: $\alpha = 0.9995$
- Minimum temperature: $T_{\min} = 10^{-6}$
- Maximum iterations: 1,200,000
- Edge comparison width: 2 pixels
- Random seed: 42

Melody Generation Parameters:

- Initial temperature: $T_0 = 1000.0$
- Cooling rate: $\alpha = 0.99$
- Minimum temperature: $T_{\min} = 0.1$
- Maximum iterations: 10,000
- Melody length: 16 notes

The jigsaw puzzle uses a slower cooling schedule to thoroughly explore the vast state space, while melody generation employs faster cooling given the smaller search space.

C. Implementation Details

Tile Transformations: Eight transformations are precomputed using NumPy operations:

- Identity, 90° rotation, 180° rotation, 270° rotation
- Vertical flip, horizontal flip
- 90° rotation + vertical flip, 90° rotation + horizontal flip

Gradient Precomputation: Tile gradients $\nabla T = |\frac{\partial T}{\partial x}| + |\frac{\partial T}{\partial y}|$ are calculated once and cached to accelerate energy evaluations.

Move Type Weighting: Position swaps receive highest probability (0.5) as they enable exploration of tile arrangement space, while transformation operations provide fine-tuning capabilities.

Undo Mechanism: Rejected moves are efficiently reversed by storing minimal state information rather than full grid copies.

IV. Results

A. Jigsaw Puzzle Reconstruction

The simulated annealing algorithm successfully reconstructed the scrambled Lena image from 16 tiles with 8 transformations each. Key performance metrics include:

- Final energy: 189,274.75
- Total iterations: 1,200,000
- Convergence: Achieved stable configuration
- Visual quality: Reconstructed image shows correct spatial arrangement with recognizable features

The energy trajectory exhibited characteristic simulated annealing behavior with rapid initial descent followed by slower refinement. The final energy indicates residual boundary mismatches due to:

- Inherent ambiguity in texture-rich regions
- Discrete tile boundaries preventing perfect alignment
- Multiple local minima with similar energy values

The algorithm successfully navigated the enormous search space of 2.58×10^{33} states to find a visually coherent reconstruction. The acceptance of uphill moves during early high-temperature phases enabled escape from poor local configurations.

B. Raag Bhairav Melody Generation

The melody generation system produced a 16-note sequence adhering to Raag Bhairav grammar:

Generated Melody:

```
Re(b) - Sa - Pa - Ma - Re(b) - Ga - Ma -
Pa - Dha(b) - Pa - Pa - Dha(b) - Dha(b) -
Ni - Ni - Dha(b)
```

Performance Metrics:

- Final energy: 30.0
- Convergence iterations: 917 (out of 10,000 maximum)
- Pakad presence: Confirmed at positions 0-4
- Consecutive repetitions: Two instances (Pa-Pa and Dha(b)-Dha(b))

Musical Analysis:

- *Modal adherence*: All notes belong to Bhairav thaat (Sa Re(b) Ga Ma Pa Dha(b) Ni Sa')
- *Pakad incorporation*: Opening phrase Re(b)-Sa-Pa-Ma-Re(b) correctly represents the characteristic Bhairav pakad
- *Melodic contour*: Predominantly ascending-descending patterns with emphasis on Pa (dominant) and Sa (tonic)
- *Phrase structure*: Natural grouping into 4-note segments with Pa and Dha(b) serving as resting points

The low final energy of 30.0 indicates smooth melodic motion with minimal large interval jumps, while the presence of some repetition adds structural coherence without excessive monotony.

C. Comparative Analysis

Table I compares the two applications across key dimensions:

TABLE I
PROBLEM CHARACTERISTICS COMPARISON

Characteristic	Jigsaw	Melody
State space size	2.58×10^{33}	$8^{16} \approx 2.81 \times 10^{14}$
Branching factor	4 move types	7 notes
Energy landscape	Non-convex	Non-convex
Constraint type	Spatial	Grammatical
Iterations to solve	1,200,000	917
Cooling rate	0.9995	0.99

The jigsaw puzzle required significantly more iterations due to its larger state space and complex spatial constraints, while melody generation converged rapidly given its grammatical structure and smaller search space.

V. DISCUSSION

A. Algorithm Effectiveness

The simulated annealing approach proved effective for both problems despite their different characteristics. Several factors contributed to success:

Temperature Schedule Design: The geometric cooling schedule balanced exploration and exploitation. High initial temperatures enabled broad state space sampling, while gradual cooling allowed convergence to quality solutions.

Energy Function Design: Both energy functions successfully encoded problem constraints. The jigsaw puzzle energy combined pixel and gradient terms to capture edge compatibility, while the melody energy balanced smoothness, diversity, and grammatical correctness.

Move Operator Design: Weighted random move selection provided appropriate exploration strategies. The jigsaw puzzle benefited from diverse move types addressing different aspects of configuration quality, while simple note substitution sufficed for melody generation.

B. Limitations and Challenges

Several limitations emerged during implementation:

Computational Cost: The jigsaw puzzle required 1.2 million iterations consuming significant computation time. Energy function evaluation dominated runtime due to repeated pixel comparisons across boundaries.

Parameter Sensitivity: Performance depended critically on cooling rate selection. Faster cooling risked premature convergence, while slower cooling increased computational cost without proportional quality improvement.

Global Optimum Uncertainty: The stochastic nature of simulated annealing provides no guarantee of finding global optima. Multiple runs with different random seeds may yield solutions of varying quality.

Musical Evaluation Subjectivity: While the generated melody satisfies grammatical constraints, aesthetic quality assessment requires human expert evaluation beyond automated metrics.

C. Comparison with Alternative Approaches

Alternative optimization strategies could address some limitations:

Genetic Algorithms: Population-based search might better explore diverse regions of the solution space, potentially finding better solutions through crossover operations that combine good partial solutions.

Constraint Satisfaction: The jigsaw puzzle could be formulated as a CSP with arc consistency algorithms for early infeasibility detection, though handling transformations would complicate constraint propagation.

Deep Learning: Neural networks trained on solved puzzles could predict tile placements or evaluate configuration quality, though requiring substantial training data.

D. Practical Applications

The demonstrated techniques extend to various domains:

Image Processing: Fragment reassembly for archaeological artifact reconstruction or document recovery from shredded materials.

Music Information Retrieval: Automatic composition systems, style transfer, or improvisation assistants for musicians.

Scheduling Problems: Resource allocation, timetabling, and logistics optimization with complex constraints.

VI. Conclusion

The laboratory experiment successfully displayed simulated annealing to solve two different optimization problems. The jigsaw puzzle reconstruction created visually consistent output from a scrambled image of size 512×512 that was divided into 16 segments, each with 8 separate transformations, concluding with an energy level of 189,274.75 after 1.2 million iterations. The melody generation factory created a 16-note Raag Bhairav sequence with an energy level of 30.0 that included the pakad phrase and maintained the required modal requirements.

The main findings were that geometric cooling schedules effectively balance exploration and exploitation, the design of the energy function should be appropriate to the domain, and weighted random move operators are useful in practice. The differences in the two problems illustrated the flexibility of simulated annealing across differing optimization landscapes.

Future research may investigate adaptive cooling schedules that are based on monitored acceptance rates, explore hybrid algorithms that combine simulated annealing with a local search refinement (which has shown promise in many variants of generative applications), as well as coherent extension to more complex musical forms employing hierarchical phrase structure and temporal dynamics. Furthermore, employing perceptual quality metrics via human evaluation studies would provide more holistic measure of generative outcomes.

The implementation establishes a foundation for applying probabilistic metaheuristics to complex combinatorial problems with non-convex energy landscapes and intricate constraint structures.

Acknowledgment

The authors would like to thank Dr. Pratik Shah, faculty of IIIT Vadodara, for guidance and support in conducting this analysis.

References

[1] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," *Handbook of Metaheuristics*, vol. 272, Springer, 2019.

[2] D. Khemani, *A First Course in Artificial Intelligence*, Chapter 4. New Delhi, India: McGraw Hill Education, 2013.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Chapters 3–4. Hoboken, NJ: Pearson, 2020.