Comparative Analysis of Hill Climbing Variants for Multimodal Fitness Landscapes: First Improvement vs Best Improvement

Cazacu Tudor
NIM-IAO
UAIC-Faculty of Computer Sciences
Iasi, Romania
19tudor99@gmail.com

***Abstract*—This paper investigates how binary representation transforms the fitness landscape of a polynomial function and analyzes the behavior of two Hill Climbing variants on the resulting multimodal space. The problem consists of maximizing $f(x) = x^3 - 60x^2 + 900x + 100$ over the integer domain [0, 31] using a 5-bit binary encoding with Hamming distance 1 neighborhood. Through exhaustive analysis, we identify four local maxima and characterize their basins of attraction for both First Improvement (FI) and Best Improvement (BI) strategies. Results demonstrate that BI significantly outperforms FI, with 62.5% vs 43.75% probability of converging to the global optimum. The study reveals fundamental differences in how these strategies navigate multimodal landscapes and provides insights into basin structure, convergence patterns, and the trade-offs between computational efficiency and solution quality.**

***Index Terms*—Hill Climbing, First Improvement, Best Improvement, fitness landscape, basins of attraction, multimodal optimization, binary representation**

I. Introduction

The performance of local search algorithms depends critically on the topology of the fitness landscape [1]. Hill Climbing (HC), one of the simplest local search algorithms, iteratively moves to neighboring solutions with better fitness values until no further improvement is possible. However, its effectiveness is strongly influenced by both the structure of the search space and the specific strategy employed for neighbor selection.

The "representation problem" [3] in metaheuristics stipulates that solution encoding can dramatically alter the fitness landscape structure. Binary representation, while computationally convenient, can transform a simple unimodal function into a complex multimodal landscape with multiple local optima, potentially creating "traps" that prevent algorithms from finding the global optimum.

This paper studies this phenomenon on the polynomial function $f(x) = x^3 - 60x^2 + 900x + 100$, which on the continuous interval [0, 31] is unimodal with a single maximum at $x = 10$. When discretized and represented using 5-bit binary encoding with Hamming distance 1 neighborhood, this function exhibits surprising behavior: the landscape becomes multimodal with four distinct local maxima, creating different basins of attraction that profoundly affect algorithm performance.

Our contribution is twofold: (1) we provide a comprehensive characterization of the induced multimodal landscape, identifying all local maxima and their basins of attraction, and (2) we conduct a detailed comparative analysis of First Improvement and Best Improvement Hill Climbing strategies, revealing significant differences in their convergence behavior, efficiency, and success rates.

II. Methodology

A. Problem Definition

The objective is to maximize the fitness function:

$$f(x) = x^3 - 60x^2 + 900x + 100 \tag{1}$$

over the discrete domain $x \in \{0, 1, ..., 31\}$. Each candidate solution x is represented by a 5-bit binary string, allowing representation of exactly 32 distinct values ($2^5 = 32$).

B. Neighborhood Structure

The neighborhood of a solution x, denoted $N(x)$, consists of all solutions at Hamming distance 1 from x. This means that a neighbor is obtained by flipping exactly one bit in the binary representation. Consequently, each solution has exactly 5 neighbors.

For example, $x = 10$ (binary: 01010) has neighbors:

- Flip bit 0: 01011 = 11
- Flip bit 1: 01000 = 8
- Flip bit 2: 01110 = 14
- Flip bit 3: 00010 = 2
- Flip bit 4: 11010 = 26

C. Local Maximum Definition

A solution x is a *local maximum* if and only if:

$$f(x) \geq f(x') \quad \forall x' \in N(x) \tag{2}$$

This is a strict definition based on the Hamming distance 1 neighborhood structure.

D. Hill Climbing Algorithm Variants

1) First Improvement (FI): The FI variant explores neighbors in a predefined order and immediately moves to the first neighbor that improves the current fitness value.

Algorithm 1: First Improvement

```
FI_HC(x_start):
  x = x_start
  while True:
    improved = False
    for x' in N(x):
      if f(x') > f(x):
        x = x'; improved = True; break
    if not improved: return x
```

2) Best Improvement (BI): The BI variant evaluates all neighbors and selects the one with the highest fitness value.

Algorithm 2: Best Improvement

```
BI_HC(x_start):
  x = x_start
  while True:
    x_best = x
    for x' in N(x):
      if f(x') > f(x_best): x_best = x'
    if x_best == x: return x
    x = x_best
```

E. Basin of Attraction

The basin of attraction for a local maximum x^* is defined as the set of all starting points that converge to x^*:

$$B(x^*) = \{x \in \mathcal{X} : \mathrm{HC}(x) = x^*\} \tag{3}$$

where $\mathcal{X} = \{0, 1, ..., 31\}$ is the search space and $\mathrm{HC}(x)$ denotes the local maximum reached when starting from x.

III. Results and Discussion

A. Fitness Landscape Characterization

Exhaustive analysis of all 32 points in the search space reveals that the binary representation induces a multimodal landscape with four distinct local maxima (Table I).

Table I
LOCAL MAXIMA IN THE DISCRETE SEARCH SPACE

x	Binary	f(x)	Quality
10	**01010**	**4100.00**	**Global Optimum**
12	01100	3988.00	Local optimum
7	00111	3803.00	Local optimum
16	10000	3236.00	Local optimum

The global optimum at $x = 10$ with $f(10) = 4100$ is surrounded by inferior neighbors, as demonstrated in Table II.

Table II
NEIGHBOR ANALYSIS FOR GLOBAL OPTIMUM (X=10)

Bit	Binary	x'	f(x')	Δf
0	01011	11	4059	-41
1	01000	8	2276	-1824
2	01110	14	3880	-220
3	00010	2	1668	-2432
4	11010	26	1156	-2944

All neighbors of $x = 10$ have strictly lower fitness values, confirming its status as a local maximum. The fitness differences range from -41 (neighbor at $x = 11$) to -2944 (neighbor at $x = 26$), demonstrating the non-linear nature of the landscape induced by binary representation.

B. Basins of Attraction Analysis

1) First Improvement Strategy: Table III presents the basins of attraction for the FI strategy. The global optimum at $x = 10$ attracts only 14 out of 32 starting points (43.75%), while 25% of the search space converges to the weakest local optimum at $x = 16$.

Table III
BASINS OF ATTRACTION - FIRST IMPROVEMENT

Optimum	f(x)	Basin Size	Percentage
x=10	**4100**	**14**	**43.75%**
x=16	3236	8	25.00%
x=12	3988	6	18.75%
x=7	3803	4	12.50%

The basin for $x = 10$ (FI) includes: {5, 6, 9, 10, 11, 13, 14, 21, 22, 25, 26, 27, 29, 30}.

Notably, starting from $x = 0$ leads to convergence at the weak local optimum $x = 16$, which is problematic since $x = 0$ is a common initialization choice.

2) Best Improvement Strategy: The BI strategy demonstrates significantly different basin structure (Table IV). The global optimum's basin expands dramatically to 20 points (62.5%), representing a 42.9% increase over FI.

Table IV
BASINS OF ATTRACTION - BEST IMPROVEMENT

Optimum	f(x)	Basin Size	Percentage
x=10	**4100**	**20**	**62.50%**
x=16	3236	5	15.63%
x=7	3803	4	12.50%
x=12	3988	3	9.37%

The basin for $x = 10$ (BI) includes: {0, 1, 2, 3, 5, 8, 9, 10, 11, 13, 14, 15, 21, 24, 25, 26, 27, 29, 30, 31}.

Critically, BI successfully converges from $x = 0$ to the global optimum, addressing the major weakness of FI. The basin for the weakest local optimum ($x = 16$) shrinks by 37.5%.

C. Comparative Analysis

1) Convergence Differences: Table V highlights the 11 starting points where FI and BI converge to different local maxima.

Table V
STARTING POINTS WITH DIFFERENT CONVERGENCE

Start	Binary	FI→	BI→	Winner
0	00000	16	**10**	BI
1	00001	16	**10**	BI
2	00010	16	**10**	BI
3	00011	16	**10**	BI
6	00110	**10**	7	FI
8	01000	12	**10**	BI
15	01111	7	**10**	BI
20	10100	12	16	Neither
22	10110	**10**	7	FI
24	11000	12	**10**	BI
31	11111	7	**10**	BI

BI finds the global optimum in 8 out of these 11 cases, while FI succeeds in only 2 cases. This demonstrates BI's superior navigation capabilities in this multimodal landscape.

2) Search Path Examples: Table VI illustrates representative search trajectories for both strategies.

Table VI
EXAMPLE SEARCH TRAJECTORIES

Start	Strategy and Path	Steps
0	FI: 0 → 16 BI: 0 → 8 → 10	2 3
15	FI: 15 → 7 BI: 15 → 11 → 10	2 3
5	FI: 5 → 13 → 9 → 11 → 10 BI: 5 → 13 → 9 → 11 → 10	5 5
25	FI: 25 → 9 → 11 → 10 BI: 25 → 9 → 11 → 10	4 4

Interestingly, for some starting points (e.g., 5, 25), both strategies follow identical paths. However, critical differences emerge for problematic initializations (e.g., 0, 15), where BI's comprehensive evaluation strategy leads to superior outcomes.

3) Efficiency Metrics: Table VII summarizes key performance metrics.

Table VII
PERFORMANCE COMPARISON METRICS

Metric	FI	BI
Success rate (→ x=10)	43.75%	**62.50%**
Weak optimum (→ x=16)	25.00%	**15.63%**
Average path length	2.28	2.34
Function evaluations/step	**1-5**	5
Computational cost	**Lower**	Higher

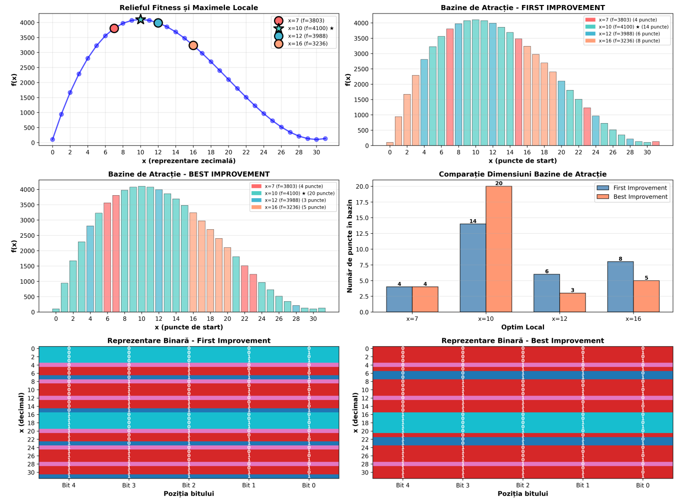

Figure 1. Comprehensive visualization: (top left) fitness landscape with four local maxima; (top right, middle left) basins of attraction for FI and BI; (middle right) basin size comparison; (bottom) binary representation patterns for both strategies.

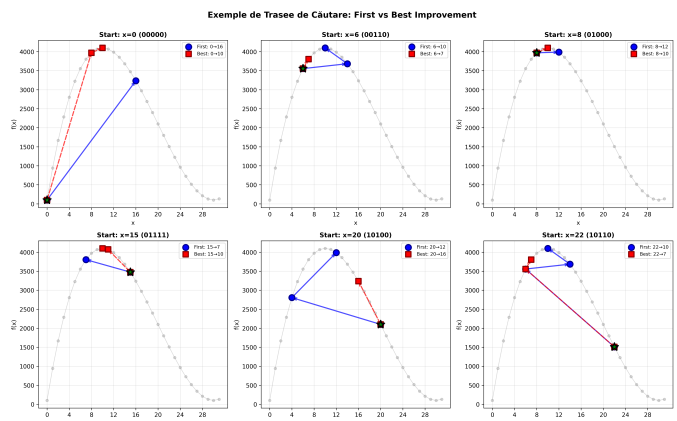

Figure 2. Example search trajectories for six different starting points. Blue circles indicate FI paths, red squares indicate BI paths, and green stars mark starting positions. Arrows show the direction of search.

D. Binary Representation Patterns

Analysis of the binary patterns reveals interesting structural properties:

Basin for x=7 (00111): All members have the pattern ***111 (last three bits set to 1), suggesting strong influence of least significant bits.

Basin for x=12 (01100): In FI, members follow pattern **1*0 (bit 2 = 1, bit 0 = 0), indicating middle-bit dominance.

Basin for x=16 (10000): Members cluster around 10000-10100, with the most significant bit (bit 4) consistently set to 1.

Hamming distances between maxima:

- d(10, 12) = 2: 01010 ↔ 01100 (closest pair)
- d(10, 7) = 3: 01010 ↔ 00111
- d(10, 16) = 4: 01010 ↔ 10000 (most distant)

The relatively small Hamming distances between local maxima create a complex navigation challenge, as the search can easily be trapped in suboptimal regions.

IV. Discussion

A. Why Best Improvement Outperforms

The superior performance of BI can be attributed to several factors:

1. Global perspective: By evaluating all neighbors, BI makes informed decisions based on complete local information, avoiding premature commitment to suboptimal directions.

2. Steepest ascent: BI consistently chooses the steepest upward path, which is particularly beneficial in rugged landscapes where small improvements can lead to dead ends.

3. Robustness to initialization: BI's comprehensive evaluation makes it less sensitive to unfortunate starting points, as evidenced by its success from x=0.

B. When First Improvement Succeeds

Despite its overall inferior performance, FI exhibits advantages in specific scenarios:

1. Computational efficiency: FI requires fewer function evaluations per iteration (average 2-3 vs. always 5 for BI), making it faster when computational resources are limited.

2. Lucky ordering: For certain starting points (e.g., x=6, x=22), the predetermined bit-flipping order happens to discover improving neighbors that lead to the global optimum, while BI's comprehensive evaluation leads to local optima.

C. Implications for Algorithm Design

This study reveals several important insights for metaheuristic design:

1. Representation matters: Binary encoding transformed a simple unimodal function into a complex multimodal landscape. The choice of representation is not merely a convenience but fundamentally shapes the optimization problem.

2. Strategy selection depends on context: BI is preferable when solution quality is paramount, while FI may be chosen when computational efficiency is critical and multiple restarts are feasible.

3. Multi-start strategies are essential: Even with BI's 62.5% success rate, 37.5% of initializations still fail to find the global optimum, underscoring the need for restart mechanisms.

4. Hybrid approaches: Combining FI's speed with BI's thoroughness through adaptive strategies could leverage the strengths of both approaches.

D. Limitations and Future Work

This study focuses on a single function with a small search space (32 points). Future research should investigate:

- Scalability to larger bit-lengths and search spaces
- Generalization to other function classes (multimodal, deceptive)
- Impact of different neighborhood structures (Hamming distance > 1)
- Hybrid strategies combining FI and BI
- Integration with other metaheuristics (simulated annealing, tabu search)

V. Conclusions

This paper provides a comprehensive analysis of how binary representation affects fitness landscape topology and how different Hill Climbing strategies navigate the resulting multimodal space. The key findings are:

1. Multimodality emergence: Binary representation with Hamming distance 1 neighborhood transformed a continuous unimodal function into a discrete landscape with four local maxima, demonstrating the profound impact of encoding on problem structure.

2. Basin structure matters: The distribution of basins of attraction varies significantly between strategies, with Best Improvement capturing 62.5% of the search space for the global optimum versus only 43.75% for First Improvement.

3. Strategy superiority is context-dependent: While BI demonstrates superior overall performance (42.9% improvement in success rate), FI maintains advantages in computational efficiency and occasionally benefits from fortuitous exploration ordering.

4. No guarantees: Neither strategy guarantees global optimum discovery, with 37.5% (BI) and 56.25% (FI) of initializations converging to suboptimal solutions, emphasizing the need for multi-start approaches.

5. The representation problem is real: This study empirically confirms the theoretical concerns about representation-induced landscape transformation, showing that encoding choices can create optimization challenges absent in the original problem formulation.

These findings have practical implications for algorithm selection in real-world optimization problems and underscore the importance of understanding landscape structure when designing search strategies.

Appendix A
Complete Basin Mapping

A. First Improvement - Detailed Basin Composition

Basin for x=10 (f=4100): {5, 6, 9, 10, 11, 13, 14, 21, 22, 25, 26, 27, 29, 30}

Binary: {00101, 00110, 01001, 01010, 01011, 01101, 01110, 10101, 10110, 11001, 11010, 11011, 11101, 11110}

Basin for x=16 (f=3236): {0, 1, 2, 3, 16, 17, 18, 19}

Binary: {00000, 00001, 00010, 00011, 10000, 10001, 10010, 10011}

Basin for x=12 (f=3988): {4, 8, 12, 20, 24, 28}

Binary: {00100, 01000, 01100, 10100, 11000, 11100}

Basin for x=7 (f=3803): {7, 15, 23, 31}

Binary: {00111, 01111, 10111, 11111}

B. Best Improvement - Detailed Basin Composition

Basin for x=10 (f=4100): {0, 1, 2, 3, 5, 8, 9, 10, 11, 13, 14, 15, 21, 24, 25, 26, 27, 29, 30, 31}

Binary: {00000, 00001, 00010, 00011, 00101, 01000, 01001, 01010, 01011, 01101, 01110, 01111, 10101, 11000, 11001, 11010, 11011, 11101, 11110, 11111}

Basin for x=16 (f=3236): {16, 17, 18, 19, 20}

Binary: {10000, 10001, 10010, 10011, 10100}

Basin for x=7 (f=3803): $\{6, 7, 22, 23\}$
Binary: $\{00110, 00111, 10110, 10111\}$
Basin for x=12 (f=3988): $\{4, 12, 28\}$
Binary: $\{00100, 01100, 11100\}$

APPENDIX B
IMPLEMENTATION CODE

A. Core Functions

```
def f(x):
    return x**3 - 60*x**2 + 900*x + 100

def get_neighbors(x):
    binary = format(x, '05b')
    neighbors = []
    for i in range(5):
        bit_flip = list(binary)
        bit_flip[i] = '1' if binary[i]=='0' else '0'
        neighbors.append(int(''.join(bit_flip), 2))
    return neighbors
```

B. Hill Climbing Implementations

```
def HC_first(start):
    current, path = start, [start]
    while True:
        improved = False
        for n in get_neighbors(current):
            if f(n) > f(current):
                current = n
                path.append(current)
                improved = True
                break
        if not improved: break
    return current, path

def HC_best(start):
    current, path = start, [start]
    while True:
        best = max(get_neighbors(current),
                      key=f, default=current)
        if f(best) <= f(current): break
        current = best
        path.append(current)
    return current, path
```

REFERENCES

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.
[2] P. Moscato, ”On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms,” Caltech Concurrent Computation Program, C3P Report 826, 1989.
[3] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.
[4] T. Jones and S. Forrest, ”Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms,” in *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995, pp. 184-192.
[5] E. Pitzer and M. Affenzeller, ”A Comprehensive Survey on Fitness Landscape Analysis,” in *Recent Advances in Intelligent Engineering Systems*, 2012, pp. 161-191.
[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
[7] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, ”Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
[8] P. Merz, ”Advanced Fitness Landscape Analysis and the Performance of Memetic Algorithms,” *Evolutionary Computation*, vol. 12, no. 3, pp. 303-325, 2004.