Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Hybrid Genetic Algorithm for Dynamic Portfolio Optimization Problems

Sarah Ayatun Nufus[1], Sutarman[2]*, Elvina Herawati[3]
[1] Graduate Student of Mathematics, Universitas Sumatera Utara, Indonesia
[2][3] Department of Mathematics, Universitas Sumatera Utara, Indonesia
[1]sarahnufus17@gmail.com, [2]sutarman@usu.ac.id, [3]herawaty.elv@gmail.com

Submitted :Jun 5, 2025 | **Accepted** : Jun 7, 2025 | **Published** : Aug 10, 2025

Abstract: Dynamic portfolio optimization is a complex problem due to continuous changes in market conditions, demanding algorithms capable of effective adaptation. Genetic Algorithms (GA) are often used for optimization problems but may face limitations in convergence speed and solution precision. This research aims to develop and evaluate a Hybrid Genetic Algorithm (HGA) that integrates GA with the Hill Climbing local search method, and to compare its performance against standard GA in solving dynamic portfolio optimization problems with the objective of maximizing the Sharpe Ratio. A series of simulation-based experiments were conducted by varying key algorithmic and dynamic environment parameters. Simulation results indicate that HGA generally has significant potential to improve performance compared to standard GA. Consistently, HGA successfully achieved superior solution quality, both in terms of Offline Performance Solution Quality and Overall Best Fitness. Regarding robustness to dynamic changes, HGA also demonstrated a smaller impact from performance degradation and a more promising recovery capability after market environment changes. Although HGA's superiority in convergence speed is not always absolute and the implementation of Hill Climbing adds to the computational time per generation, the improvement in solution quality and robustness offered in many configurations can be considered a worthwhile trade-off, especially for complex dynamic portfolio optimization problems. These findings support the hypo that hybridizing GA with local search can provide a positive contribution, noting that careful parameter tuning is crucial for maximizing HGA's potential.

Keywords: Dynamic optimization; dynamic portfolio optimization; genetic algorithm; hill climbing; hybrid genetic algorithm; sharpe ratio

INTRODUCTION

Optimization is a critical research area directly related to everyday decision-making problems, such as planning, transportation, and logistics, aiming to either maximize or minimize outcomes. Efficient solutions allow problems to be solved by optimally utilizing resources like time and computation. However, mathematical complexity and the exponentially increasing number of possible solutions with problem size pose significant challenges (Goldberg, 1989). Dynamic optimization problems, a variation of optimization problems, involve making interconnected decisions over time, where decisions in one period affect future periods, unlike static optimization, which considers only a single timeframe (Bertsekas, 2017; Branke et al., 2005). In many real-world scenarios, such as determining annual investment amounts, problems change over time; thus, algorithms for dynamic optimization must not only find optimal solutions but also track these changes effectively (AbdAllah et al., 2018).

A significant application of dynamic optimization is in investment management, leading to the concept of dynamic portfolio optimization (Elton et al., 2014). Investors continuously adjust their asset allocations in response to evolving market expectations, personal goals, risk tolerance, and new information. This approach extends Markowitz's classic portfolio theory (H. Markowitz, 1952), which is inherently static, by introducing time and the ability for periodic portfolio rebalancing to adapt to dynamic market conditions (Elton et al., 2014). The primary challenge in dynamic portfolio optimization lies in its complexity, often involving high uncertainty, large decision spaces, and potential structural shifts in the investment environment (Branke et al., 2005). Consequently, metaheuristic approaches are often favored for their efficiency in finding solutions.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

 e-ISSN : 2541-2019
p-ISSN : 2541-044X

Genetic Algorithms (GAs), a type of evolutionary algorithm (Goldberg, 1989), are frequently used for such complex optimization problems but can suffer from slow convergence and may get trapped in local optima, especially in dynamic contexts (Sun et al., 2013). To address these limitations, Hybrid Genetic Algorithms (HGAs) combining GAs with local search methods like Hill Climbing are proposed (Neri & Cotta, 2012). Hill Climbing focuses on refining current solutions incrementally (Russell & Norvig, 2021). This hybridization aims to enhance efficiency and accuracy, allowing the algorithm to adapt better to changing conditions and improve the potential for finding optimal solutions (Sun et al., 2013; Yuan & Yang, 2013). Previous research indicates that HGA can be a robust and efficient tool, improving search speed and precision, and effectively solving dynamic optimization problems by tracking dynamic optima (Sun et al., 2013; Yuan & Yang, 2013). This study aims to develop and evaluate an HGA for dynamic portfolio optimization, specifically focusing on maximizing the Sharpe Ratio (Sharpe, 1966, 1994), and compare its efficacy against a standard GA.

RESEARCH METHOD

This research employs a quantitative approach using optimization and computational experiments to develop and implement a hybrid genetic algorithm for dynamic portfolio optimization. The problem is defined by dynamically changing market parameters and the objective of maximizing the Sharpe Ratio (Sharpe, 1994). This section outlines the specific problem formulation, the architecture of the proposed hybrid algorithm, and the metrics used for performance evaluation.

Dynamic Portfolio Optimization Problem

Dynamic portfolio optimization addresses the challenge of managing investment portfolios in environments where market conditions are constantly changing. Unlike static approaches that assume a fixed set of parameters, this paradigm acknowledges the evolving nature of financial markets, requiring strategies that can adapt over time. The core of the problem involves determining the optimal allocation of assets within a portfolio and periodically rebalancing these allocations to achieve specific investment objectives, typically maximizing returns for a given level of risk, or minimizing risk for a target return, in the face of fluctuating market parameters such as expected returns and asset correlations (Branke et al., 2008).

1. Decision Variables (H. M. Markowitz, 1968):
   The portfolio weights
   $$w(t) = [w_1(t), w_2(t), \dots, w_N(t)]^T$$
   for N assets at time t, subject to:
   $$\sum_{i=1}^{N} w_i(t) = 1$$
   $$w_i(t) \geq 0, \forall i \in \{1,2,\dots,N\}$$
2. Dynamic Market Parameters:
   a. Expected Asset Returns (Elton et al., 2014):
   $$\mu(t) = [\mu_1(t), \mu_2(t), \dots, \mu_N(t)]^T$$
   b. Asset Covariance Matrix (Elton et al., 2014; H. Markowitz, 1959):
   $$\Sigma(t)$$
   an $N \times N$ matrix where $\Sigma_{ij}(t)$ is the covariance between asset i and j at time t
3. Portfolio Performance Metrics:
   a. Expected Portfolio Return:
   $$E[R_p(t)] = w(t)^T \mu(t) = \sum_{i=1}^{N} w_i(t)\mu_i(t)$$
   b. Portfolio Variance (H. M. Markowitz, 1968):
   $$\sigma_p^2(t) = w(t)^T \sum(t)\, w(t), = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i(t) w_j(t) \sum\nolimits_{ij}(t)$$
   c. Portfolio Standard Deviation [6]:
   $$\sigma_p(t) = \sigma_p^2(t)$$
   assuming $\sigma_p(t) > 0$
4. Objective Function:
   Maximize the Sharpe Ratio $S(w,t)$ (Sharpe, 1966, 1994):
   $$(w,t) = \frac{E[R_p(t)] - R_f(t)}{\sigma_p^2(t)}$$
   where $R_f(t)$ is the risk-free rate of return at time t.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The optimization problem is to find $w^{*}t_g$ such that:

$$w^{*}t_g = \arg\max_{w} S(w,t)$$

subject to the weight constraints $\sum_{i=1}^{N} w_i(t) = 1$ and $w_i(t) \geq 0, \forall i \in \{1,2,\ldots,N\}$, where tg indicates discrete time periods (generations) and market parameters $\mu(t_g)$ and $\Sigma(t_g)$ change (AbdAllah et al., 2018; Branke et al., 2005).

Hybrid Genetic Algorithm (HGA) with Hill Climbing

This approach combines the global search capabilities of a standard Genetic Algorithm (GA) with the local search proficiency of the Hill Climbing (HC) method. The GA component explores the solution space broadly, identifying promising regions, while the HC component refines solutions within these regions to find local optima more efficiently. This synergy aims to overcome the limitations of using either algorithm in isolation, potentially leading to better quality solutions and faster adaptation in dynamic environments (Goldberg, 1989; Neri & Cotta, 2012; Russell & Norvig, 2021). The HGA operates through a sequence of evolutionary steps, including initialization, fitness evaluation, selection, genetic operations (crossover and mutation), and the integrated local search phase.

1. Initialization:
   Generate an initial population
   $$P(0) = \left\{w_1(t_0), w_2(t_0), \ldots, w_{N_{\text{pop}}}(t_0)\right\}$$
   of N_{pop} individuals (portfolios) [1]. Each $w_j(t_0)$ must satisfy the sum-to-one and non-negativity constraints. Set current generation $g = 0$, initial time $t_g = t_0$.
2. Evaluation:
   For each individual w_j in the current population, calculate its fitness using the Sharpe Ratio $S(w_j, t_g)$with current market parameters $\mu(t_g), \Sigma(t_g)$, and $R_f(t_g)$ [12].
   $$S(w_j, t_g) = \frac{w_j^T \mu(t_g) - R_f(t_g)}{\sqrt{w_j^T \Sigma(t_g) w_j}}$$
3. Selection:
   Select N_{pop} parents $P_{\text{parent}}(g)$ from the current population based on their fitness values (e.g., tournament selection) (Goldberg, 1989).
4. Crossover:
   With probability P_c, pair individuals from $P_{\text{parent}}(g)$. For a pair (w_a, w_b), apply arithmetic crossover (for real-coded representation) (Michalewicz, 1996) to produce offspring w'_a, w'_b:
   $$w'_a = \alpha w_a + (1-\alpha) w_b$$
   $$w'_b = (1-\alpha) w_a + \alpha w_b$$
   where α is a random number in [0,1]. Resulting offspring form $P_{\text{offspring\_raw}}(g)$. Normalize weights to meet constraints
5. Mutation:
   with probability P_m, apply mutation (e.g., Gaussian mutation) (Michalewicz, 1996) to each gene (weight) $w'_i{}'$ in an offspring w':
   $$w''_i = w'_i + N(o, \sigma_{\text{mut}})$$
   Resulting offspring form $P_{\text{offspring\_raw}}(g)$. Normalize weights and ensure non-negativity.
6. Local Search (Hill Climbing):
   For each individual $w'' \in P_{\text{offspring\_mutated}}(g)$ (or a subset selected with probability pHC), apply Hill Climbing for kHC iterations or until local convergence (Russell & Norvig, 2021). The Hill Climbing process $w'' = HC(w'', t_g)$ involves (Russell & Norvig, 2021):
   a. Initialize $w_{\text{current}} = w''$
   b. Repeat kHC times or until no improvement:
      i. Generate a neighboring solution $w_{\text{neigbor\_hc}}$ by a small modification of $w_{\text{curent\_hc}}$ (e.g., shifting a small weight amount between two randomly chosen assets, then re-normalizing).
      ii. Calculate $S(w_{\text{neigbor\_hc}}, t_g)$.
      iii. If $(w_{\text{neigbor\_hc}}, t_g) > S(w_{\text{curent\_hc}}, t_g)$, then $w_{\text{curent\_hc}} = w_{\text{neigbor\_hc}}$
   c. Return the improved individual $w'' = w_{\text{current\_hc}}$. This produces $P_{\text{offspring\_hc}}(g)$. The integration of local-search like Hill Climbing into GA characterizes a memetic algorithm or HGA (Neri & Cotta, 2012).
7. Re-evaluation (if necessary):

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Re-evaluate fitness of individuals in $P_{\text{offspring\_hc}}(g)$, if not implicitly done by Hill Climbing.

8. Population Replacement:
   Form the new population $P(g)$ for the next generation by selecting Npop individuals from a combination of $P(g-1)$ and $P_{\text{offspring\_hc}}(g)$. (e.g., using elitism) (Goldberg, 1989).
9. Environment Update & Iteration:
   If a predefined number of generations (change frequency) has passed, update market parameters: $(t_g) = \mu_{new}, \Sigma(tg) = \Sigma_{\ new}$ (Branke et al., 2008). Increment $g = g + 1$. Repeat from Step 2 until termination criteria (e.g., max generations $G_{\max}$) are met.

Performance Metrics

To evaluate and compare the effectiveness of the proposed HGA against a standard GA in solving dynamic portfolio optimization problems, a set of well-defined performance metrics is employed. These metrics are designed to capture various aspects of algorithmic performance, including the speed at which an algorithm adapts to changes, the quality of the solutions it finds over time, and its stability or robustness when faced with shifts in the problem environment. The selection of these metrics is guided by established practices in the field of evolutionary optimization in dynamic environments (Branke et al., 2005).

1. Convergence Speed: Generations to reach a certain threshold (e.g., 90% of best-known fitness) after a market change (Branke et al., 2005).
   $$\frac{1}{K}\sum_{i=1}^{K} \text{Generations}_i$$
   where K is the number of market changes and Generations_i is generations to converge after i-th change.
2. Solution Quality (Offline Performance): Average of the best fitness found in each generation (Branke et al., 2008).
   $$P_{\text{offline}} = \frac{1}{G_{\text{total}}}\sum_{g=1}^{G_{\text{total}}} F_{\text{best}}(g)$$
   where G_{total} is total generations, $F_{\text{best}}(g)$ is best fitness at generation g.
3. Robustness (Fitness Drop after Change): Average absolute drop in best fitness immediately after an environmental change (Branke et al., 2008).
   $$\text{Drop}_k = F_{\text{best\_before\_change}} - F_{\text{best\_immediately\_after\_change}}$$

RESULT

This section reports the empirical outcomes of the Hybrid Genetic Algorithm (HGA) and the baseline GA on dynamic portfolio optimization benchmarks. We present the experimental settings, the time-varying market environment, and the recorded metrics—best fitness per generation, offline performance, recovery time to 90% of peak after shifts, immediate fitness drop, and computation time—without interpretation. The goal is to provide a transparent account of what was measured and observed. The following subsections provide the figures (Figures 1–8) and summary tables (Tables 1–2).

Experimental Parameters

Simulations were conducted with varying parameters to assess algorithm performance across different conditions. Careful parameter selection is crucial for algorithm performance. Key parameters included:

1. Number of Assets (NUM_ASSETS_OPTIONS): [5, 10].
2. Maximum Generations (MAX_GENERATIONS_OPTIONS): [100, 200].
3. Population Size (POPULATION_SIZE_OPTIONS): [30, 60].
4. Mutation Rate (MUTATION_RATE_OPTIONS): [0.05, 0.15].
5. Crossover Rate (CROSSOVER_RATE_OPTIONS): [0.7, 0.9].
6. Hill Climbing Parameters (HGA specific):
   a. HC Iterations (HC_ITERATIONS_OPTIONS): [5, 10].
   b. HC Probability (HC_PROBABILITY_OPTIONS): [0.2, 0.5].
7. Dynamic Environment Parameters (Baseline):
   a. Market Change Frequency (BASE_CHANGE_FREQUENCY): Every 20 or 25 generations.
   b. Annual Risk-Free Rate (BASE_RISK_FREE_RATE_ANNUAL): 0.02 (2%).

The dynamic market environment simulated changes in expected asset returns $(\mu(t))$ and the covariance matrix $(\Sigma(t))$ periodically.

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Visualization of Simulation Outcomes

To further illustrate the performance characteristics of the GA and HGA algorithms, this section refers to the graphical and tabular data presented below. These visualizations provide a more detailed view of the evolutionary process and the comparative performance across different experimental setups.

The evolutionary trajectory of the best fitness (Sharpe Ratio) per generation for various parameter combinations of the standard Genetic Algorithm is depicted in Fig. 1 through Fig. 8 is:

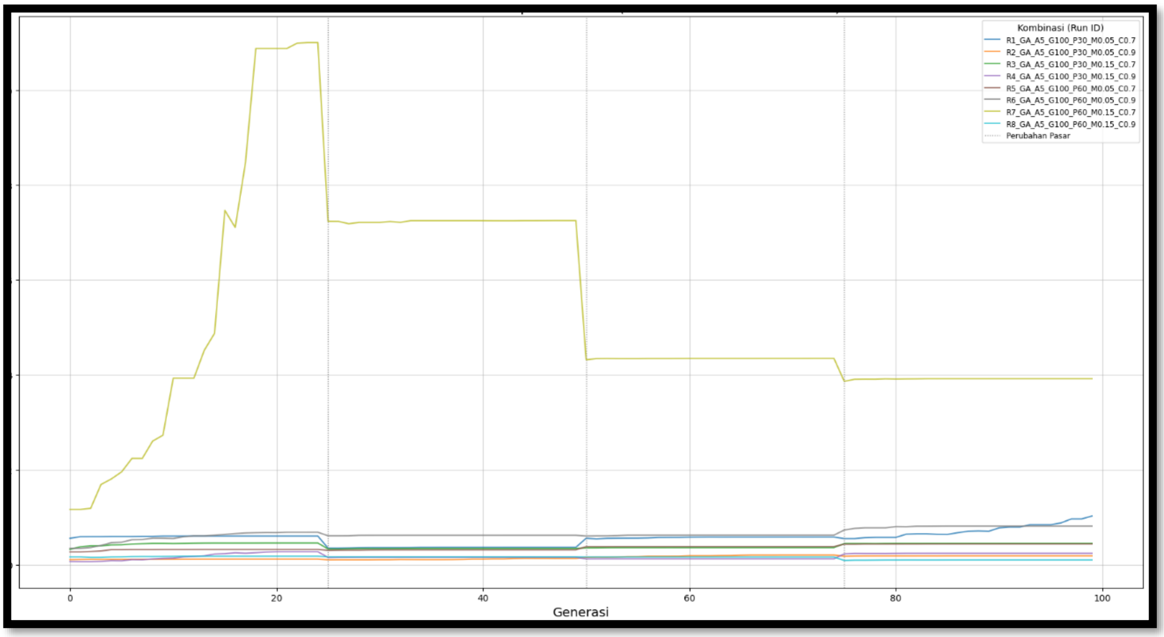

Fig. 1 The GA fitness evolution for 5 assets and a maximum of 100 generations

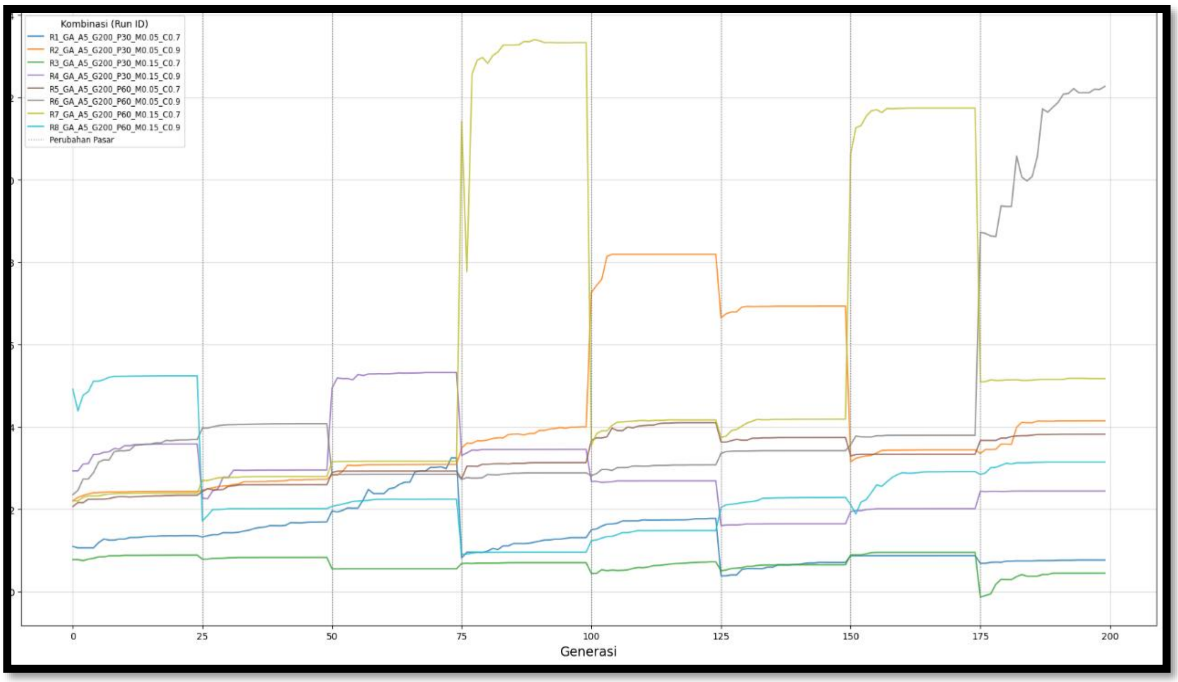

Fig. 2 The GA fitness evolution for 5 assets and a maximum of 200 generations

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

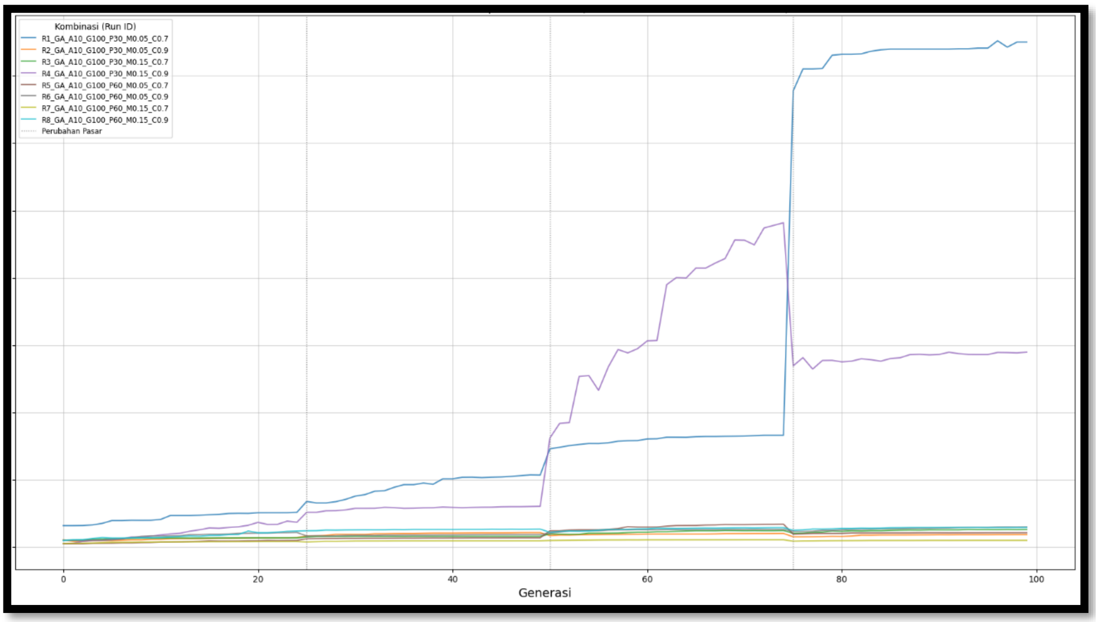

Fig. 3 The GA fitness evolution for 10 assets and a maximum of 100 generations

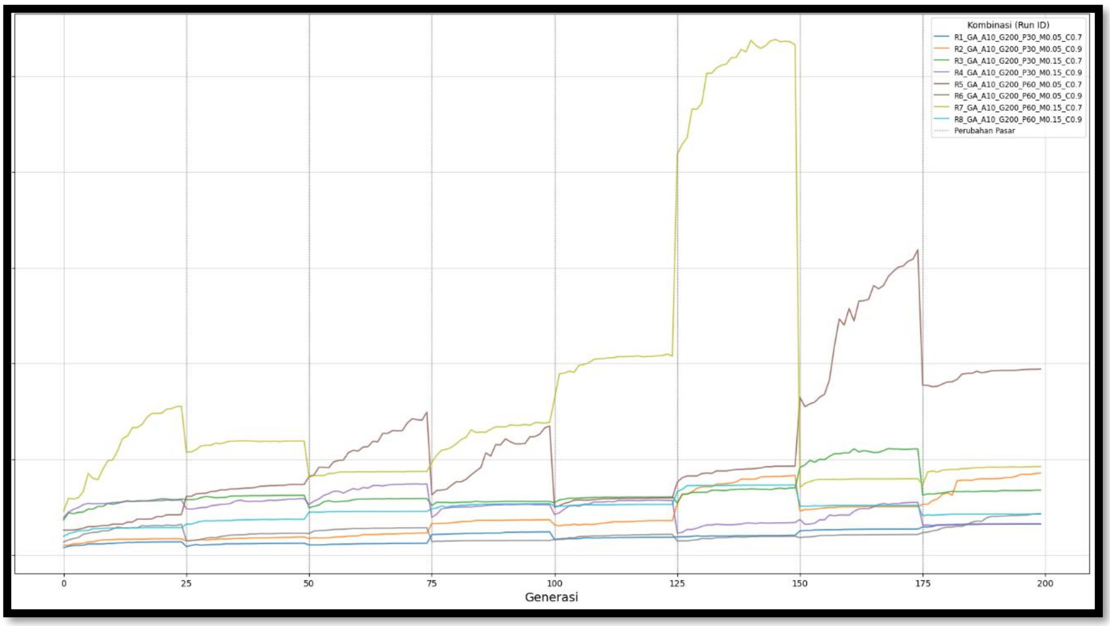

Fig. 4 The GA fitness evolution for 10 assets and a maximum of 200 generations

These figures generally show an upward trend in fitness, with noticeable drops and recoveries corresponding to the periodic changes in the market environment, demonstrating the GA's adaptive attempts. Similarly, Fig. 5 through Fig. 8 of the illustrate the fitness evolution for the Hybrid Genetic Algorithm:

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

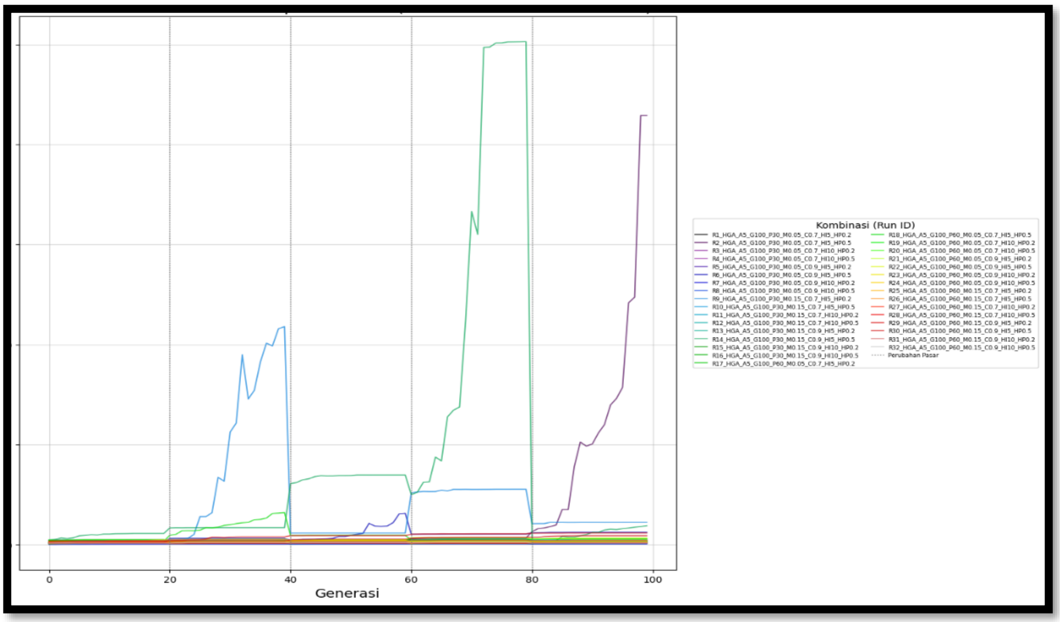

Fig. 5 The HGA fitness evolution for 5 assets and a maximum of 100 generations

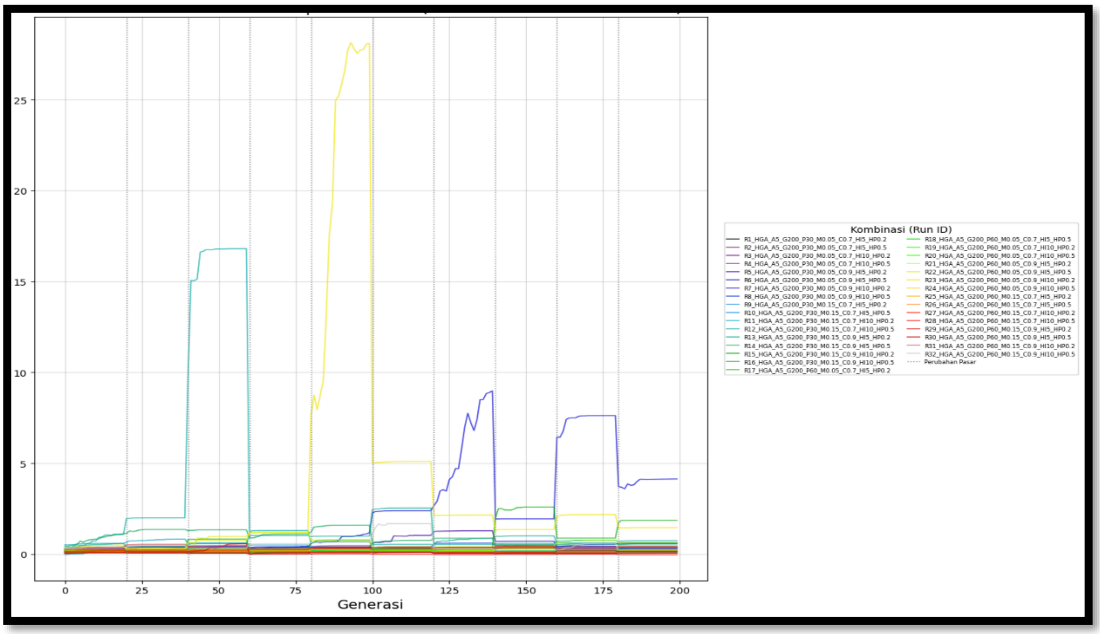

Fig. 6 The HGA fitness evolution for 5 assets and a maximum of 200 generations

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

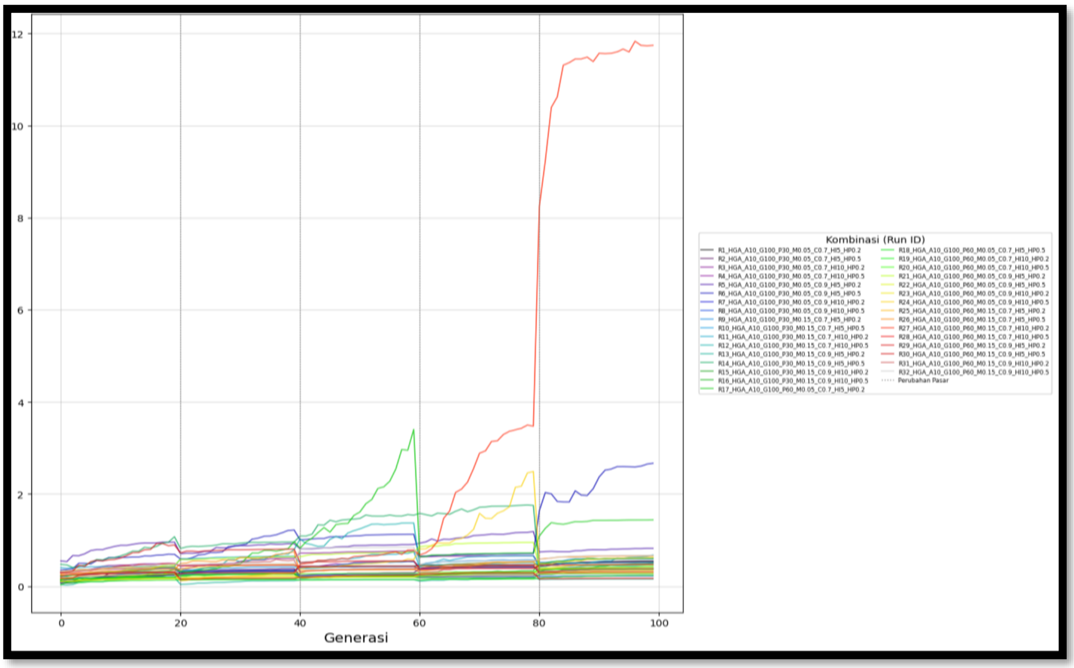

Fig. 7 The HGA fitness evolution for 10 assets and a maximum of 100 generations

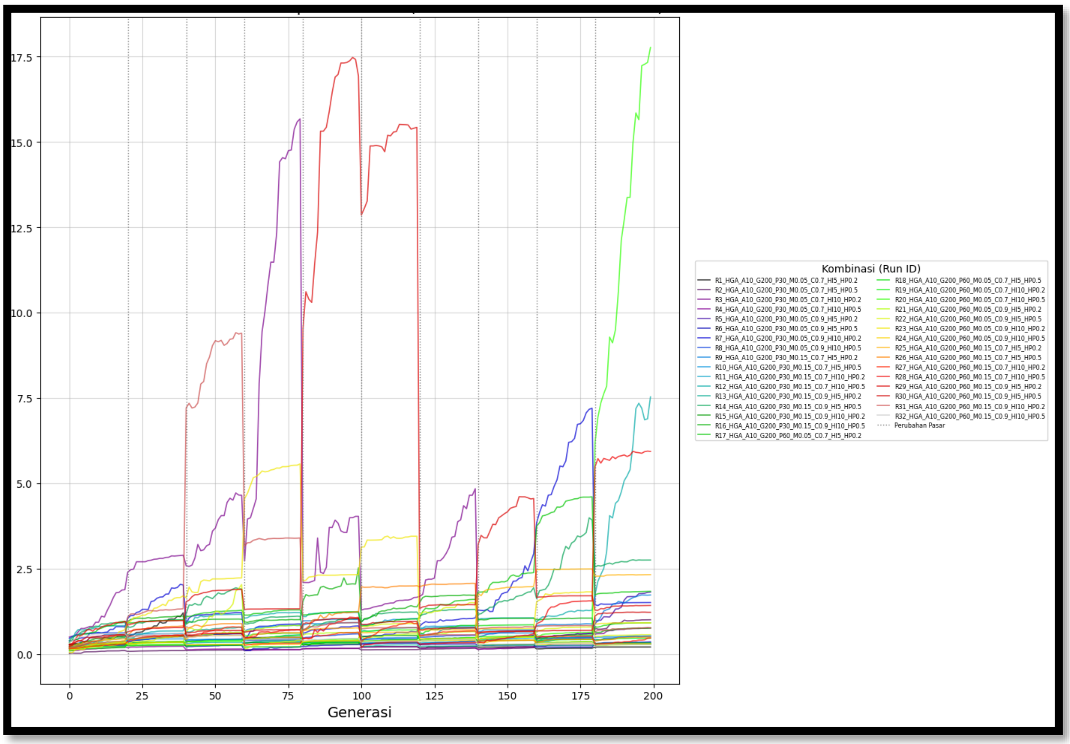

Fig. 8 The HGA fitness evolution for 10 assets and a maximum of 200 generations

Observation of these HGA figures, when compared to their GA counterparts, often reveals steeper initial climbs in fitness and potentially higher plateaus, suggesting the positive impact of the Hill Climbing local search. This allows for a direct comparison of HGA's performance characteristics against GA under similar experimental conditions.

Table 1 Offers a comparison of GA and HGA performance when their simulations might have been run with slight variations or in separate batches

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Algorithm	NUM_ASSETS	MAX_GENERATIONS	Offline Performance	Best Fitness	Average Convergence Generation Speed (to 90% Peak)	Average Fitness Drop After Change	Average Total Computation Time (s)
GA	5	100	0.96587229	1.800505439	44.25	0.316361335	0.3848793
HGA	5	100	0.75086166	4.11974373	41.46875	1.031548385	0.5572990
GA	5	200	0.31247655	0.659331909	74.75	0.034224306	0.8339613
HGA	5	200	0.55580750	2.289204986	86.3125	0.613192864	0.1359154
GA	10	100	1.35844355	3.464127639	60.625	0.574536125	0.4580272
HGA	10	100	0.53052448	1.175173819	55.90625	0.528767847	0.0495863
GA	10	200	0.65377674	1.597291944	143.5	0.132286163	0.7326053
HGA	10	200	0.96404073	3.736880699	114.625	1.021119908	0.1982857

Table 2 Offers a comparison of GA and HGA performance when their simulations might have been run with GA is treated as a baseline

Algorithm	NUM_ASSETS	MAX_GENERATIONS	Offline Performance	Best Fitness	Average Convergence Generation Speed (to 90% Peak)	Average Fitness Drop After Change	Average Total Computation Time (s)
GA	5	100	0.84096945	3.077320344	32.125	0.57466977	0.341304
HGA	5	100	0.50564597	1.5585778	39.5625	0.271796688	1.466975
GA	5	200	0.45889809	1.31369912	93.875	0.109716711	3.331089
HGA	5	200	1.15749333	7.707800728	87.25	0.083615831	2.680616
GA	10	100	1.44652514	10.1328908	62.625	0.258711791	2.979706
HGA	10	100	0.79590402	3.601414211	60.65625	0.154694895	1.904345
GA	10	200	0.66246646	1.595677054	92.42857143	0.112217264	1.830245
HGA	10	200	0.80280807	3.039164638	102.5	0.208250309	2.938683

Collectively, these figures and tables from above provide the empirical basis for the analysis and conclusions drawn regarding the efficacy of HGA compared to GA for the dynamic portfolio optimization problem.

DISCUSSIONS

This section interprets the empirical findings and articulates their implications for dynamic portfolio management. We discuss differences in solution quality and tracking ability, post-shift convergence behavior, robustness to environmental change, and computational trade-offs, and translate these patterns into practical recommendations. We also outline limitations and directions for future work. For coherence, the discussion follows the same metrics and order used in the result section.

The performance of GA and HGA was evaluated based on convergence speed, solution quality (offline performance and overall best fitness), and robustness to dynamic changes (average fitness drop after change) and total computation time, drawing comparisons similar to studies by (Sun et al., 2013; Yuan & Yang, 2013).

Solution Quality and Tracking Ability

We assess solution quality along two complementary dimensions: offline performance (the average of the best fitness attained at each generation) and overall best fitness (the single highest Sharpe Ratio achieved during a run).

For offline performance, HGA consistently maintained higher average quality over time than GA, indicating that it not only reaches good portfolios but also keeps them more steadily across environmental changes. For example, with 10 assets over 200 generations, HGA attained an offline performance of 0.964 (Table 1), whereas GA configurations were around 0.654 (Table 1). In like-for-like comparative runs, GA and HGA recorded 0.662 and 0.803, respectively (Table 2). This pattern suggests stronger tracking of the moving optimum and fewer prolonged dips after regime shifts, which is desirable in dynamic portfolio settings where interim performance matters as much as end-of-run peaks.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 3, July 2025
DOI : https://doi.org/10.33395/sinkron.v9i3.14868
e-ISSN : 2541-2019
p-ISSN : 2541-044X

For overall best fitness, HGA also reached higher peaks in most scenarios, reflecting the benefit of combining global search with local refinement. With 5 assets and 100 generations, for instance, HGA achieved an average best fitness of 4.12 compared with 1.80 for GA (Table 1). The superiority in peak Sharpe Ratio aligns with expectations from hybrid (memetic) designs: local search polishes promising weight vectors discovered by GA, raising the attainable ceiling. While the magnitude of the gap can vary with parameterization and problem scale, the majority of tested settings favor HGA on this criterion as well.

Convergence Behavior

We measure convergence speed as the average number of generations needed to regain 90% of the run's peak fitness after each market change. Lower values mean faster recovery. Using this metric, HGA generally showed a tendency to recover more quickly than GA, but the advantage was not universal across all parameter settings and problem sizes. For instance, with 5 assets and 100 generations, HGA recovered in 41.47 generations on average, compared with 44.25 for GA (Table 1), indicating a modest but consistent edge for HGA. In the 10-asset, 200-generation setting, the Table 1 averages likewise favor HGA (114.625) over GA (143.5), suggesting that the local-search component helped HGA re-stabilize sooner after shocks. However, when we look at a matched baseline in Table 2, a GA run reached 92.43 generations—faster than that HGA average—showing that with certain parameterizations and landscapes GA can still converge more quickly.

Overall, these results imply that HGA's local refinement usually accelerates post-shift recovery, but sensitivity to parameters (population size, mutation/crossover rates, and hill-climbing intensity) and instance scale can flip the outcome. Practically, this argues for tuning or adapting the local-search intensity over time (e.g., more aggressive right after a shift, lighter once stability returns) and for reporting dispersion statistics (e.g., medians and interquartile ranges across seeds and shifts) alongside means to capture variability in recovery behavior.

Robustness to Dynamic Changes

We assess robustness as the average immediate fitness drop after a market change—the gap between the best fitness just before a shift and the best fitness in the first generation after it. Smaller drops indicate better carry-over of solution quality across regimes. In runs with 10 assets and 100 generations, HGA showed a mean drop of 0.529 (Table 1) versus 0.575 for GA (Table 1), an absolute reduction of 0.046 (about 8% relative to GA). The advantage is clearest when GA and HGA use closely matched parameters, consistent with the idea that local refinement helps stabilize promising portfolios. That said, this metric is sensitive to parameterization and random seeds; reporting dispersion (e.g., interquartile ranges) alongside means provides a fuller picture. In practice, preserving sufficient population diversity and using modest, adaptive local search tend to soften the post-change cliff without sacrificing long-run performance.

Computation Time

As expected, HGA incurs higher runtime per generation because of the embedded hill-climbing step. In the 5-asset/100-generation setting, GA completed a run in about 0.38 s (Table 1), while HGA ranged roughly 0.56–1.59 s depending on the hill-climbing probability and number of iterations (Table 1; see Table 2 for matched comparisons). The magnitude of this overhead scales with local-search intensity, population size, and problem size. In practice, the extra cost often buys higher solution quality and better robustness; moreover, if HGA shortens the number of generations needed to recover after shifts or to reach a target fitness, the total time to a satisfactory solution can be comparable to GA. When latency is critical, dialing down the hill-climbing intensity (or applying it adaptively after detected changes) helps balance speed against quality.

Overall, the results suggest that HGA, by incorporating Hill Climbing, enhances the exploration and exploitation balance, leading to improved solution quality and adaptability in dynamic portfolio optimization, albeit with increased computational effort. This is consistent with findings in other domains where HGA has been applied.

CONCLUSION

Based on the comprehensive analysis of simulation results for dynamic portfolio optimization, this research draws several key conclusions regarding the comparative performance of the standard Genetic Algorithm (GA) and the Hybrid Genetic Algorithm (HGA) with Hill Climbing.

The HGA demonstrated significant potential for performance improvement over the standard GA within the dynamic portfolio optimization context. This superiority was particularly evident in achieving higher quality solutions and exhibiting better adaptability to changing market conditions. HGA consistently attained superior average 'Offline Performance Solution Quality' and 'Overall Best Fitness' across most tested parameter configurations, indicating that the integration of Hill Climbing local search effectively refines portfolio solutions generated by the GA's evolutionary process.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Regarding convergence speed, while HGA tended to reach target fitness levels in fewer generations in several scenarios, its advantage was not absolute and varied with specific parameter interactions. However, in terms of robustness to dynamic changes, HGA generally showed a smaller decrease in performance and a more promising recovery capability after market environment shifts, especially under identical core algorithmic parameters for GA and HGA. The implementation of Hill Climbing inherently increased the computation time per generation for HGA. Nevertheless, the enhancements in solution quality and robustness offered by HGA across many configurations suggest this trade-off can be justifiable, particularly for complex dynamic portfolio optimization problems where investment decision quality is paramount.

In essence, these findings support the hypothesis that hybridizing Genetic Algorithms with local search mechanisms like Hill Climbing positively contributes to solving dynamic portfolio optimization problems. HGA proved capable of generating portfolios with better risk-return profiles and demonstrated superior adaptability, despite considerations of computational efficiency. Careful parameter tuning for both the GA components and the Hill Climbing mechanism is crucial for maximizing the full potential of the HGA approach.

REFERENCES

AbdAllah, A. F. M., Essam, D. L., & Sarker, R. A. (2018). Genetic Algorithms-Based Techniques for Solving Dynamic Optimization Problems with Unknown Active Variables and Boundaries. In *Innovative Computing, Optimization and Its Applications* (pp. 151–166). https://doi.org/10.1007/978-3-319-66984-7_9

Bertsekas, D. P. (2017). *Dynamic Programming and Optimal Control* (4th ed., Vol. 1).

Branke, J., Deb, K., Miettinen, K., & Slowiński, R. (2008). *Multiobjective optimization: Interactive and evolutionary approaches* (Vol. 5252). Springer Science & Business Media.

Branke, J., Salihoğlu, E., & Uyar, Ş. (2005). Towards an analysis of dynamic environments. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, 1433–1440. https://doi.org/10.1145/1068009.1068237

Elton, E. J., Gruber, M. J., Brown, S. J., & Goetzmann, W. N. (2014). *Modern Portfolio Theory and Investment Analysis* (9th ed.). John Wiley & Sons.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, *7*(1), 77. https://doi.org/10.2307/2975974

Markowitz, H. (1959). *Portfolio selection: Efficient diversification of investments–John Wiley & Sons, Inc*. New York. Chapman & Hall, Limited, London.

Markowitz, H. M. (1968). *Portfolio selection*. Yale university press.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs* (3rd ed.). Springer-Verlag Berlin Heidelberg.

Neri, F., & Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, *2*, 1–14. https://doi.org/10.1016/j.swevo.2011.11.003

Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson Series in Artificial Intelligence.

Sharpe, W. F. (1966). Mutual Fund Performance. In *The Journal of Business* (Vol. 39, Issue 1, pp. 119–138). University of Chicago Press. http://www.jstor.org/stable/2351741

Sharpe, W. F. (1994). The Sharpe Ratio. In P. L. Bernstein & F. J. Fabozzi (Eds.), *Streetwise: The Best of The Journal of Portfolio Management* (pp. 169–178). Princeton University Press. https://doi.org/doi:10.1515/9781400829408-022

Sun, F., Du, W., Qi, R., Qian, F., & Zhong, W. (2013). A Hybrid Improved Genetic Algorithm and Its Application in Dynamic Optimization Problems of Chemical Processes. *Chinese Journal of Chemical Engineering*, *21*(2), 144–154. https://doi.org/10.1016/S1004-9541(13)60452-8

Yuan, Q., & Yang, Z. (2013). On the performance of a hybrid genetic algorithm in dynamic environments. *Applied Mathematics and Computation*, *219*(24), 11408–11413. https://doi.org/10.1016/j.amc.2013.06.006

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.