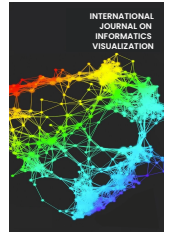




INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Understanding Search Behavior in the Simulated Kalman Filter Algorithm

Nor Hidayati Abdul Aziz ^{a,*}, Ooi Jing Hao ^b

^a Centre for Engineering Computational Intelligence, Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia

^b Faculty of Information and Science Technology, Multimedia University, Melaka, Malaysia Country

Corresponding author: *hidayati.aziz@mmu.edu.my

Abstract—Metaheuristic algorithms are crucial for solving complex and dynamic problems in computational optimization. It is essential to fully understand how an algorithm searches, as it helps to improve the algorithm and its applications in various domains. This paper provides a detailed analysis of how the Simulated Kalman Filter algorithm searches for optimal solutions. The SKF algorithm is an optimization method inspired by the Kalman filter estimation techniques. The algorithm was introduced in 2015 to address unimodal problems. Since its inception, the SKF algorithm has improved and is used to solve many optimization problems. Our study aims to bridge the gap in existing research by investigating how SKF effectively balances search space exploration and known solution exploitation. Through systematic experimentation using the Brown function as a benchmark, we explored the social dynamics and movement style of the SKF algorithm, in addition to the convergence efficiency and accuracy. When we applied the same approach as suggested in the referenced paper, we gained insights into SKF's unique strengths and limitations of SKF when compared to other algorithms. The findings illustrate SKF's unique capabilities in handling the exploration-exploitation trade-off. This study helps to set the foundation for creating more advanced algorithms and optimization strategies in the future. Future research will examine how enhancements to the SKF algorithm impact and enhance its search behavior.

Keywords—Simulated Kalman filter algorithm, metaheuristic optimization, search behavior analysis, optimization algorithms.

Manuscript received 11 Aug. 2024; revised 29 Oct. 2024; accepted 11 Dec. 2024. Date of publication 31 Jan. 2025.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Metaheuristic algorithms are vital in solving complex optimization problems across various domains. Metaheuristic algorithms are known for being flexible, robust, and excellent at finding the best solutions to a wide range of issues without relying on specific problem structures or gradients [1]. These algorithms are generally simple to implement and require fewer parameters than traditional optimization methods [2]. Metaheuristics provides a better exploration-exploitation trade-off, solution quality, and computing time [3]. They are beneficial for high-dimensional problems that conventional methods cannot resolve [4]. These characteristics make them more reliable and efficient than exact methods [3] and have led to their widespread use in engineering, finance, and computer science [5].

The list of state-of-the-art metaheuristic algorithms is long. Genetic Algorithm (GA) which was introduced by John Holland in the 1970s, Particle Swarm Optimization (PSO) introduced by James Kennedy and Russel Eberhart in the

1990s, and Ant Colony Optimization (ACO) which was introduced by Marco Dorigo are widely adopted state-of-the-art metaheuristic algorithms. Each has numerous variants and improvements. The categorization of metaheuristic algorithms differs from the way researchers look at it. Evolutionary Algorithms (EA) and Swarm Intelligence (SI) were among the earliest and most essential metaheuristic algorithms [6]. Both philosophies are rooted in natural phenomena. EAs are influenced by genetics and natural selection, whereas self-organized animals inspire SIs. GA falls under EAs. The GA imitates the natural-selection process. It selects the fittest individuals for reproduction to produce offspring for the next generation. In addition to GA, EA encompasses other categories, such as Evolutionary Strategies (ES) and Differential Evolution (DE). ES focuses on adapting strategy parameters, such as the mutation and crossover rates. Researchers primarily use ES to solve real-valued optimization problems. DE uses vector differences to perturb the population vectors. The DE algorithm is also excellent for solving continuous optimization problems. PSO

and ACO belong to the SI. The social behavior of birds and fish inspires PSO. In PSO, the individuals in the swarm move towards the best solution by following the top-performing individuals in the swarm. ACO is based on the behavior of ants seeking a path between their colony and source of food. ACO is particularly useful for discrete optimization such as routing and scheduling.

Interestingly, some sources provide more specific categorizations. For instance, [7] presented two more categories: physics-based, and human-based algorithms. Several other sources of inspiration have influenced the development of various algorithms. Researchers have developed many algorithms to search for the best solutions, each with its own unique traits and abilities [8], [9]. Biological, chemical, and physical phenomena inspire most metaheuristic algorithms. In addition to these natural sources of inspiration, estimation-based methods such as the Kalman filter and Finite Impulse Response (FIR) are highly inspiring [10]–[17]. These algorithms integrate empirical estimation techniques in their search strategies and demonstrate improved performance across various problem domains. Among all estimation-based optimization algorithms, the Simulated Kalman Filter (SKF) algorithm stands out for its unique approach to balancing exploration and exploitation using the Kalman filter approach.

Exploration–exploitation balance is a fundamental concept in meta-heuristic algorithms, reflecting the trade-off between exploring the search space for new solutions and exploiting known reasonable solutions to find even better solutions [1],[18]. This aspect is essential in all optimization processes. To achieve this balance, metaheuristic algorithms employ various strategies. For instance, the Adaptive Balance Optimization Algorithm (ABOA) uses a global search phase and a local search phase controlled by a fixed parameter to balance exploration and exploitation throughout the iterative process [19]. Similarly, the Memetic Chaotic Gravitational Search Algorithm incorporates a quasi-Newton method to enhance the exploitation capabilities while maintaining exploration through chaotic mechanisms [20]. Different algorithms approach this balance uniquely. The Balance Adjustment based Chaotic Gravitational Search Algorithm (BA-CGSA) introduces a sine random function and balance mechanism to improve diversity and prevent premature convergence [21]. The Spherical Search Gravitational Search Algorithm (SSGSA) combines the effective exploration of Spherical Search with the exploitation capabilities of the Gravitational Search Algorithm [22]. These diverse approaches highlight the ongoing research efforts to optimize the exploration-exploitation balance in metaheuristic algorithms.

Metaphors are of significant importance in the formation and description of metaheuristic algorithms. It helps in understanding and effectively conveying complex ideas. In the seminal work by Hayward and Engelbrecht, the authors highlighted a critical gap in the meta-heuristic research domain, specifically the under-explored area of search behavior characteristics [23]. In addressing this gap, the authors created a guide that outlines a straightforward approach for characterizing search behavior that researchers can universally apply to various types of meta-heuristics. This study aims to shed light on the dynamics of the SKF search

process through a search behavior analysis based on this method.

The importance of a detailed understanding of the search behavior in optimization algorithms cannot be overstated. This behavior ultimately determines an algorithm's efficiency, effectiveness, and applicability to various problems. The number of journal papers on metaheuristic research surged from 1994 to 2021, and the application areas were diverse [24]. Reference [25] highlighted the similarities between new algorithms (algorithms with high citation counts from 2000) and classical algorithms by identifying common features shared among different algorithms. He suggested that many new algorithms reassemble existing concepts from classical algorithms rather than introduce fundamentally new ideas. Considering this, our study aims to significantly advance the understanding of the Simulated Kalman Filter algorithm by providing a detailed analysis of its search behavior. Unlike previous works that primarily focused on performance metrics, this research delves into the mechanics of how SKF effectively balances exploration and exploitation. This novel perspective helps to identify specific strengths and potential areas for enhancement, positioning SKF as a versatile tool for various optimization problems. This study presents exciting possibilities for further algorithmic advancements and practical uses of SKF by thoroughly examining SKF's search behavior in terms of its social dynamics and movement style.

II. MATERIALS AND METHOD

The SKF algorithm is an optimization approach inspired by the estimation abilities of the Kalman Filter [26]. Its purpose is to address global optimization problems by treating the state estimation problem as an optimization challenge. SKF believes that estimating the state of a dynamic system is similar to finding the best solution in a search space for optimization problems. SKF views the optimal solution as a time-independent estimate. The term “state” in SKF represents an agent's location in the search space of the optimization problem. Similar to many metaheuristic algorithms, SKF employs a population of agents that continually adjust their positions in the search domain to approach the best possible solution. Each agent functions as its Kalman filter to estimate the optimal solution based on a simulated measurement process.

A. Predict-Measure-Estimate Cycle

The SKF algorithm employs three well-known steps in the Kalman filter operation for optimization. This process allows each agent to update its best estimate of the optimal solution based on simulated measurement. Fig. 1 illustrates the simplified principle of the SKF algorithm, where N refers to the total agents used.

The SKF algorithm shown in Fig. 2 is crucial for visually understanding the flow and structure of the SKF algorithm. It provides a clear overview of how Kalman equations are adapted to the SKF algorithm. The equations are divided into three sets: the first one comprises the prediction. In the original SKF algorithm, the predicted value was the previously estimated value. The second one is the simulated measurement, which explores the search space. The simulated measurement mimics the measurement update in a standard Kalman filter using the best-so-far solution (X_{true}) as the

guide. The last one is for estimation. The estimate equations ensure that all agents are improving towards the optimal solution. This iterative process continued until the stopping condition was satisfied.

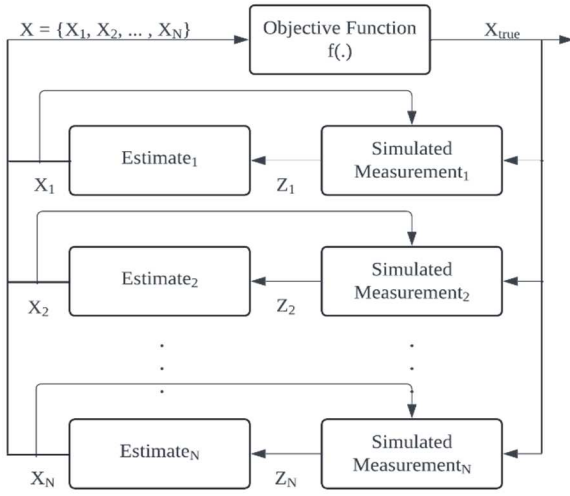


Fig. 1 Simplified principle of Simulated Kalman Filter algorithm

Fig. 2 shows the suggested SKF parameters. The results of the experiments in [27]–[29] show that altering the P, Q, and R parameters had no impact on SKF algorithm's effectiveness. However, this study employed the original SKF algorithm for the analysis.

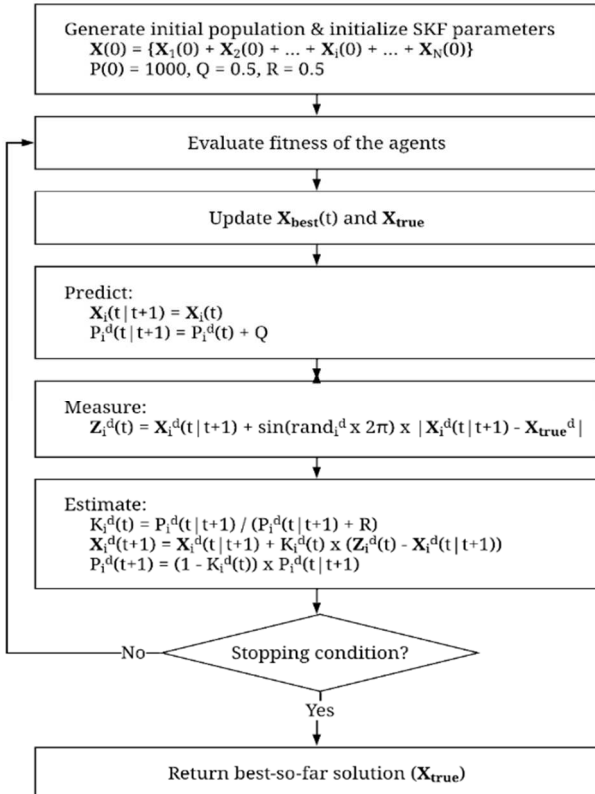


Fig. 2 Flowchart of Simulated Kalman Filter algorithm

III. RESULTS AND DISCUSSION

In minimization problems, the global minimum is the best solution, whereas the local minimum is a lower point but not

the absolute lowest point. When analyzing metaheuristics, researchers typically examine the quality of solutions produced. This evaluation involves assessing the solutions' proximity to the optimal solution or global minimum.

A. Benchmark Objective Function

Several studies have discussed the importance of developing comprehensive and challenging benchmarks for multi-objective optimization in the context of benchmark functions. These include the proposal of new benchmark functions based on zigzag functions [30], the development of a generative benchmarking approach for many-objective problems [31], and the introduction of mixed-integer benchmark problem suites [31]. These benchmarks aim to provide a diverse set of test problems with various features and difficulties in effectively evaluating the performance of the optimization algorithms.

This study focuses on optimizing the Brown function [32]. The Brown function is a benchmark function used to evaluate global optimization algorithms. Because of its unique mathematical characteristics, it is especially valuable for testing an algorithm's performance in steep and flat regions. In the experimental setup, we set the dimensions to 20 and used an SKF population size of 30. By using the same problem and experimental setup, researchers can compare the SKF search behavior with other algorithms studied in [23].

The Brown function is highly curved, continuous, differentiable, and scalable. It is also non-separable. This means that the terms in the function are interdependent, involving the products of different input variables. This key feature makes it challenging for optimization algorithms to determine the global minimum efficiently. Equation 1 defines the Brown function for x_i between -1 and 4.

$$f(x) = \sum_{i=1}^{n-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)} \quad (1)$$

Fig. 3 shows the surface plot of the Brown function in 2-dimension. The function is smooth, and small changes in the input result in small changes in the output.

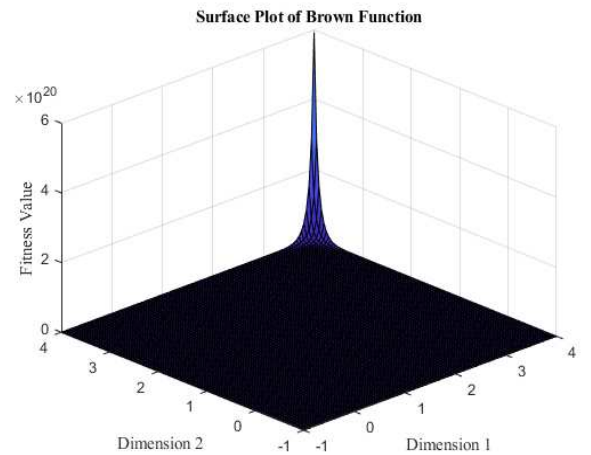


Fig. 3 Surface plot of Brown function

The Brown function has a single global optimum (i.e., no local minimum). However, steep gradients and complex interactions between variables can make it appear locally multimodal. The function reaches its lowest value of 0 when

all input variables are 0, as shown in Fig. 4, marked by the red dot.

The Brown function presents a more complex landscape than the Sphere function. Compared to the Rosenbrock function, the Brown function presents greater challenges owing to its strict penalty for deviations from the optimum path. By using the Brown function, we can determine how well a metaheuristic algorithm converges, the accuracy of its solutions, and the overall efficiency of the algorithm.

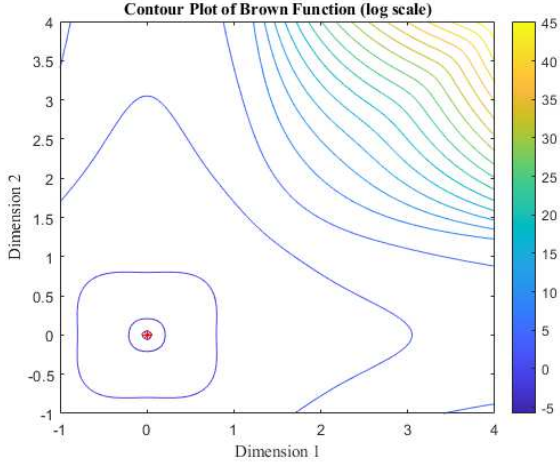


Fig. 4 Contour plot of Brown function in log scale

To better visualize the Brown function, we displayed its surface and contour on a graph. We specifically focused on the range $[-1,1]$, as shown in Fig. 5. The 3D surface plot shows the smooth surface of the Brown function with a valley and peaks, indicating regions of low and high fitness values. Closer contour lines at the sides indicate steeper gradients, whereas widely spaced lines represent flatter regions at the center of the plot.

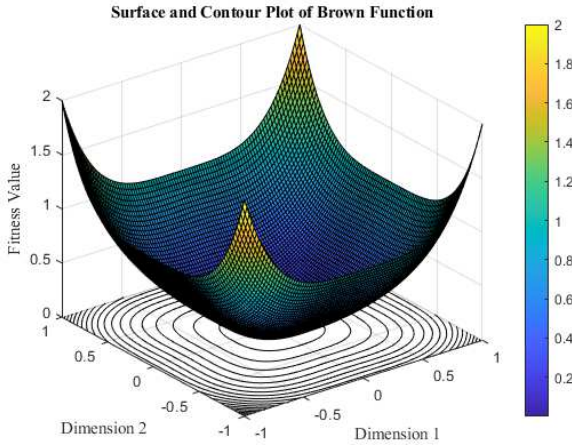


Fig. 5 Surface and contour plot of Brown function in $[-1,1]$

B. Search Behavior in SKF

Exploration and exploitation are two well-known search behaviors in the field of optimization. Achieving the right balance is critical for the performance of metaheuristic algorithms. This balance affects the capability of the algorithm to find global optima, steer clear from local optima, and converge efficiently. Many approaches attempt to switch from exploration to exploitation as the quest progresses [33].

Metaheuristic algorithms depend on exploration behavior to discover promising regions within a search space. Exploration typically occurs at the beginning of the search process. Exploration is necessary; however, excessive exploration can be inefficient. Algorithms may overlook promising solutions if they require too much time to explore. Exploitation behavior allows metaheuristic algorithms to intensify their search for the best solutions. This behavior helps the algorithms converge on the optimal solution. However, excessive exploitation can lead to the algorithm to quickly becoming trapped at a sub-optimal location.

In the SKF, the search space is initially populated randomly. This injects a high level of exploration at the beginning. The SKF algorithm continues with the current solution during the prediction step. This allows for further exploration of promising solutions. The measurement equation includes a sine function, helping the SKF agents explore, guided by their current best solution. Adding a random term to the sine function renders the measurement process more stochastic and can influence the search. The Kalman estimation equations play a vital role in exploitation. Agents in SKF progressively refine their estimates of the optimum solution based on their simulated measurement values and previous estimates. Initially, the Kalman gain was set to favor exploration, giving more weight to new information from the simulated measurements. As SKF agents gather more confidence in their estimates, the search steers towards a more precise solution. These two factors cause a gradual shift from exploration to exploitation, as the search continues. Reference [34] offers a tutorial on the population-based SKF algorithm. This includes a numerical example, which makes it easier to understand.

Various studies have demonstrated that incorporating adaptive mechanisms, hybrid approaches, and novel search strategies can significantly enhance the exploration-exploitation balance, improving performance across various optimization problems [35],[36]. Researchers have made efforts to enhance the performance of SKF algorithms. A brief review of the SKF algorithm has identified approximately 16 fundamental improvements [37]. Table 1 lists the types of modifications applied to the SKF algorithm and the specific mechanism employed. Most of the improvements are made to increase the exploration capability of the SKF algorithm to avoid premature convergence. Researchers have enhanced the SKF algorithm in the prediction, measurement, and estimation steps.

References [38]–[40] use opposition-based learning as the prediction operator. This mechanism helps the algorithm explore a wider area by considering both the predicted and opposite solutions. Integrating a second algorithm can help the SKF algorithm explore a wider range of search areas. This second algorithm was used to predict the solutions because the original SKF algorithm lacked a prediction mechanism. PSO and Gravitational Search Algorithm (GSA) are two algorithms that have been tested as prediction operators for SKF algorithms [41],[42]. In [43]–[45], different approaches were discussed for implementing the algorithms as prediction operators. Reference [46] suggested using the Sine Cosine Algorithm (SCA) to explore and exploit the simulated measurement, which was proven effective. Reference [47]

suggests considering the opposite solution after the estimation step to enhance the exploration of the SKF algorithm.

Researchers have observed that this mechanism prevents the algorithm from prematurely converging to a local optimum by utilizing the current optimum solution to generate the opposite population. Another way to improve exploration behavior is by introducing a mutation operator. In [48], a mutation operator was introduced after the estimation step. A mutated solution was used to replace one of the solutions. Depending on the fitness value, the mutated solution either replaces the best-so-far solution, leads the search in the next iteration, or replaces any solutions. Improving exploration behavior is not the only way to enhance the performance of an algorithm. In [49], the authors enhanced the SKF algorithm's exploitability by adding an exponential term to the estimation equation. This compelled the algorithm to take larger steps in the early optimization stages and proved effective.

Finally, the SKF algorithm can be improved by changing its update mechanism. In the original SKF algorithm, agents update their solution estimation after completing all agents' fitness evaluation and prediction measurement cycles. However, [50] showed that allowing an SKF agent to update its estimation separately with the entire population resulted in better performance. Therefore, the update mechanism in SKF shows promises for improving its ability to explore and exploit resources. References [51]–[53] researched further on adapting the update mechanism by switching between synchronous and asynchronous updates to further improve performance. Based on these findings, the impact of the iteration strategy on SKF algorithm is influenced by the presence of memory.

TABLE I
IMPROVEMENTS TO THE EXPLORATION AND EXPLOITATION CAPABILITY OF THE SKF ALGORITHM

Ref.	Type of Modification	Specific Mechanism	Behavior Improved
[38]-[40]	Prediction operator	Opposition-based learning	Exploration
[41]-[45]	Prediction operator	Hybrid with another algorithm	Exploration
[46]	Measurement method	Hybrid with another algorithm	Exploration and exploitation
[47]	Estimate equation	Opposition-based learning	Exploration
[48]	Estimate equation	Mutation operator	Exploration
[49]	Estimate equation	Exponential term	Exploitation
[50]-[53]	Update mechanism	Adaptive switching	Exploration and exploitation

In summary, SKF's search behavior is driven primarily by its predict-measure-estimate cycle. Each step uniquely contributes to the algorithm efficiency:

1) *Predict step*: This step allows each agent to project its current state into the future. This projection introduces an opportunity for exploration, considering that future states can

lead to an optimal solution. Variants using opposition-based learning and hybrid approaches during the prediction step help explore a wider area, facilitating the exploration of a broader region of the search domain. These approaches significantly enhance the exploration capability of SKF, thus reducing the risk of premature convergence.

2) *Measure step*: Each agent's prediction was refined using a simulated measurement process in the measurement step. A balance between exploring new areas and exploiting existing knowledge is crucial. Adapting a suitable algorithm that can balance exploration and exploitation in this step may improve the accuracy of the SKF algorithm.

3) *Estimate step*: The estimating step involves updating the agent's position based on simulated measurements. Dynamic adjustment during this step aids in further exploration or exploitation throughout the search process. Variants using opposition-based learning and mutation enhance the SKF algorithm's ability to escape local optima by exploring more regions containing better solutions. One might also decide to refine good solutions effectively to avoid being trapped in local optima.

C. Visualizing SKF Search Behavior

In addition to the solution quality, a thorough analysis of meta-heuristic behavior should be conducted so that better improvements can be made to the algorithms [54]. The SKF's exploration and exploitation behavior in solving the unimodal Brown function is reflected in the convergence curve in Fig. 6. The convergence curve depicts how the SKF's estimate evolves over 500 iterations. The y-axis measures the fitness value of the best solution so far (X_{true}). In the initial stages, the SKF prioritizes exploration to successfully traverse the search space to locate the region containing the global minimum. This is evident from the steep decrease in fitness values found during the first few iterations. Because SKF focuses on a promising region, the curve exhibits a gradual descent.

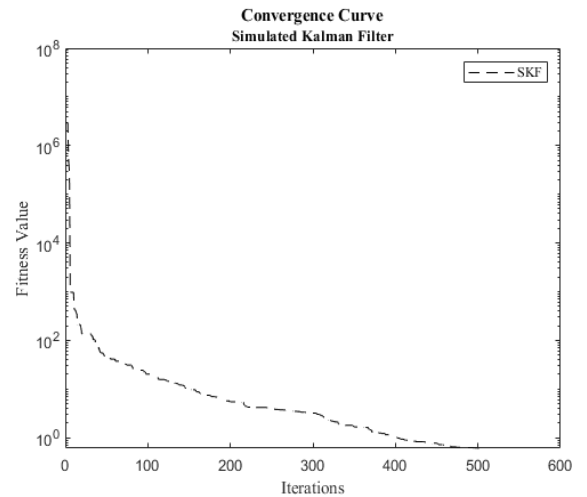


Fig. 6 Convergence curve for Simulated Kalman Filter

This section demonstrates the visualization of social dynamics and movement style in SKF to understand the algorithm's search behavior characteristics further.

1) Social Dynamics:

Social dynamics are used to understand how information is shared between individuals in a population [23]. Reference [55] suggests that incorporating problem-specific dynamics into metaheuristics could improve performance in certain scenarios. Two characteristics can be observed under social dynamics: improvement frequency and population turnover. SKF has a fixed population size. Therefore, the improvement frequency matrix of 30 individuals $\times$ 500 iterations will contain a value of '1' if the individual's fitness improves at the specific iteration or a value of '0' if otherwise. As all agents in the SKF algorithm make their own estimation of the optimum solution and continue to the next iteration, their improvement statuses are logged consistently after the estimation step. The colored rectangles in Fig. 7 are plotted with iterations along the y-axis and SKF agents on the x-axis. Each blue-colored rectangle for each iteration represents where an individual improves fitness. From Fig. 7, we can see the freshness of information in SKF agents throughout the 500 iterations from the existence of the blue rectangles for all agents sporadically from the start until the end. There is no large uncolored area that can be seen, as all agents continue to improve their own estimation and share the information with the whole population in every iteration. The solution with the highest fitness score is identified, leading the search for the subsequent iteration. This cycle continued until the stopping conditions were satisfied. Comparing this finding with the findings by Hayward and Engelbrecht in [23], we can see the similarity in the information freshness between the SKF and DE algorithms. The population turnover figure is interpreted as a heatmap in Fig. 7, plotting the number of individuals that changed positions in each iteration.

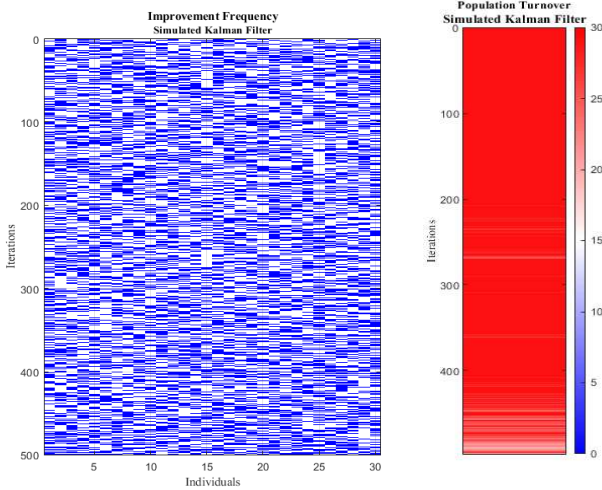


Fig. 7 Basic visualization of improvement frequency combined with visualization of population turnover (with a 1% threshold) for the Simulated Kalman Filter algorithm.

The bluer the color, the fewer the number of individuals changing positions, and inversely, the redder the color, the larger the number of individuals changing positions at every iteration. In this case, the same threshold as in [23] was used to account for a specific position change size. Only position changes that were more than 1% of the space size problem were logged. In the case of SKF, all agents change positions at every iteration except for the best so-far solution. By

eliminating insignificant position changes, the minimum number of agents changing positions in an iteration is 18. This differs from the DE algorithm because less than half of the DE population changes positions in each iteration. Except for GA, which has a persistent population turnover of around 25%, SKF, like other metaheuristics, has a reduced number of individuals, making a turnover of significant size towards the end of the search due to exploitation kicks in. Based on the social dynamics of the SKF algorithm, we can see that the amount of good information is fresh and well-distributed among all individuals throughout the population. From the improvement frequency and population turnover, we can see that the number of movements in the SKF algorithm is highly related to the number of improvements made.

A bar chart showing each individual's total number of improvements is illustrated in Fig. 8 to provide a quick comparison of which individuals improved the most. Based on the figure, the spread across the number of individuals shows uniform behavior, meaning that all agents are equally responsible for searching, and no elitist strategy is employed.

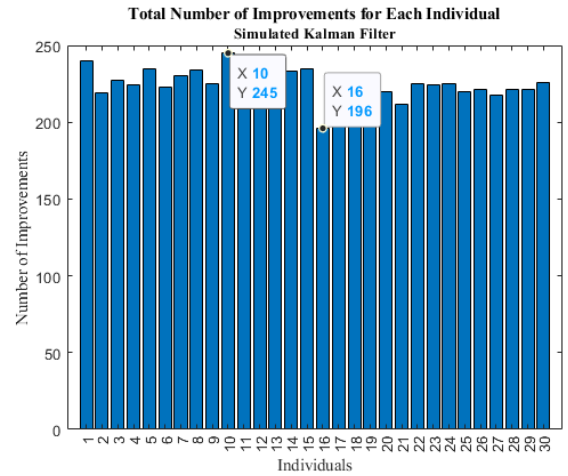


Fig. 8 Quick comparison of the total number of improvements for each individual in the Simulated Kalman Filter.

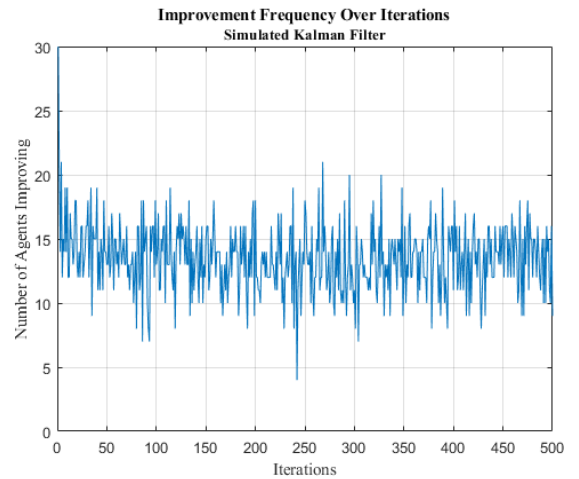


Fig. 9 Number of agents improving for each iteration in the Simulated Kalman Filter.

Out of the 15000 position changes (30 agents $\times$ 500 iterations), approximately 6749 movements resulted in locations with better fitness values. On average, approximately 45% of the movement by each SKF agent

exploits near the best solution, whereas the remaining 55% explores the search space. The improvement frequency status over the iterations fluctuates around 13 (median value), as illustrated in Fig. 9. Over 500 iterations, most of the time, around 12-16 agents (approximately 40%–53%) make positive improvements in every iteration. This high percentage indicates how well the algorithm engages the agents involved in the search process for a better solution.

In summary, the improvement frequency and population turnover in metaheuristics are essential factors in balancing exploration and exploitation, ultimately affecting their capability to escape local optima and efficiently reach optimal solution [56].

2) Movement Style:

Understanding movement style helps us see how individuals' movements affect algorithm improvement [23]. Metaheuristic algorithms employ various movement styles to navigate around the search space effectively. This diversity in movement styles contributes to the algorithm's performance [57]. We can view this in two ways: the size of the steps taken and the quality of the steps taken. The average change in the step size for each iteration is logged to measure the sizes of the steps taken. Fig. 10 shows the shape of the footprint of the SKF algorithm. No normalization was performed in this case. The higher the peaks, the larger the number of steps taken. Looking at the SKF step footprint pattern, we can see a sharp decrease in step size at the beginning of the iteration. This pattern is similar to that of the initial PSO pattern. However, after that, we can see that the step sizes stabilize, with some minor fluctuations. This indicates that SKF agents make more minor adjustments as they refine their estimation of the optimal solution. The large spikes in the earlier iterations suggest that the algorithm makes significant changes based on exploring the simulated measurements.

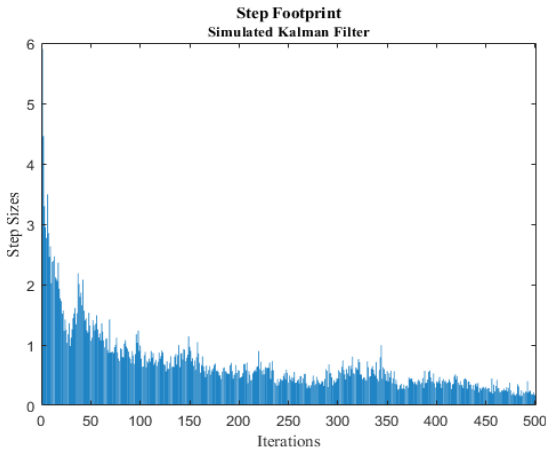


Fig. 10 Step footprint visualization for the Simulated Kalman Filter.

Fig. 11 shows the quality of the step obtained through the step fitness figure. The change in fitness values before and after an estimate for each iteration was logged regardless of positive or negative changes. Then, the average fitness changes were used for visualization. We used a scatter plot on a log scale following [2]. The distribution of large to minor data points is essential when evaluating the fitness value pattern. In contrast to Fig. 10, the step fitness plot shows that the most significant improvement occurred. The scatter plot

shows a large change in fitness at earlier iterations. By comparing Fig. 10 and 11, we can see the influence of the size step on its quality. With an increasing number of iterations, the change in fitness decreases. This shows that SKF agents converge towards a stable estimate around the optimal solution. The occasional significant shift in fitness during the second half of the search indicates the SKF algorithm's ability to continue exploring. This behavior is a unique characteristic of SKF and is not observed in DE, ES, or PSO.

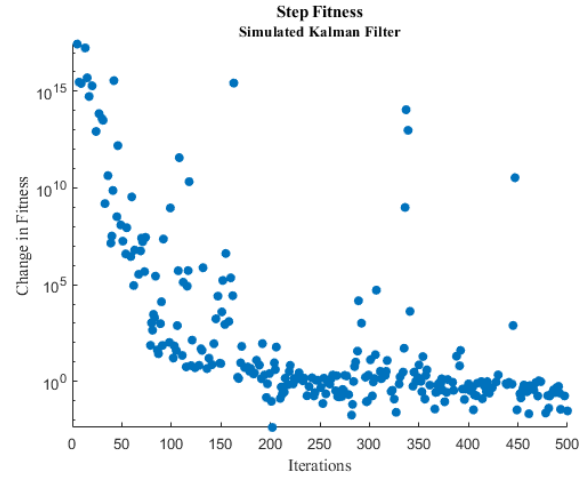


Fig. 11 Step fitness visualization for the Simulated Kalman Filter.

Interestingly, initialization methods can significantly affect step sizes and quality. The diagonal linear uniform initialization (DLU) technique improves convergence speed and solution accuracy by adopting subspace sampling instead of the whole space [58]. This update mechanism of the algorithm enhances both precision and convergence rate.

IV. CONCLUSION

In conclusion, it is essential to understand the search behavior to improve optimization research and practice. With a complete understanding of the SKF's operations and behavior, we can hone the algorithm to solve problems more effectively and use it in different fields. Our study proved the unique search behavior of the SKF algorithm through its social dynamics and movement style. Our research confirms that the SKF algorithm effectively balances exploration and exploitation even with high convergence. Future research could focus on mapping the behaviors of improved SKF algorithms using the same measure to systematically assess which changes have the most significant impact on the algorithm's effectiveness. In addition to investigating versions of SKF algorithms, researchers can compare them with other popular methods for better positioning.

ACKNOWLEDGMENT

This research is funded by a matching grant from the Multimedia University in collaboration with Universitas Indonesia (Matching Grant MMUI/240004).

REFERENCES

- [1] D. Brahim, M. Kobayashi, M. E. A. Elaissaoui Elmeliiani, M. Al Ali, and T. Khatir, "Metaheuristic Optimization Algorithms: an overview," *HCMCOU Journal of Science – Advances in Computational Structures*, Feb. 2024, doi: 10.46223/hcmcoujs.acs.en.14.1.47.2024.

- [2] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, and B. Abdollahzadeh, "Advances in Sparrow Search Algorithm: A Comprehensive Survey," *Archives of computational methods in engineering: state of the art reviews*, vol. 30, no. 1, pp. 427–455, Aug. 2022, doi: 10.1007/s11831-022-09804-w.
- [3] N. Khanduja and B. Bhushan, "Recent Advances and Application of Metaheuristic Algorithms: A Survey (2014–2020)," Springer Singapore, 2020, pp. 207–228. doi: 10.1007/978-981-15-7571-6_10.
- [4] H. Mohammadzadeh and F. S. Gharehchopogh, "A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection," *International Journal of Communication Systems*, vol. 34, no. 3, Nov. 2020, doi:10.1002/dac.4670.
- [5] S. M. Almufti, A. Ahmad Shaban, R. Ismael Ali, J. A. Dela Fuente, and Z. Arif Ali, "Overview of Metaheuristic Algorithms," *Polaris Global Journal of Scholarly Research and Trends*, vol. 2, no. 2, pp. 10–32, Apr. 2023, doi: 10.58429/pgjsrt.v2n2a144.
- [6] A. P. Piotrowski, M. J. Napiorkowski, J. J. Napiorkowski, and P. M. Rowinski, "Swarm Intelligence and Evolutionary Algorithms: Performance versus speed," *Inf. Sci. (Nij.)*, vol. 384, pp. 34–85, 2017, doi: 10.1016/j.ins.2016.12.028.
- [7] O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, L. Abualigah, and R. A. Zitar, "Multiclass feature selection with metaheuristic optimization algorithms: a review," *Neural Computing and Applications*, vol. 34, no. 22, pp. 19751–19790, Aug. 2022, doi: 10.1007/s00521-022-07705-4.
- [8] H. T. Sadeeq and A. M. Abdulazeez, "Metaheuristics: A Review of Algorithms," *Int. J. online Biomed. Eng.*, vol. 19, no. 9, pp. 142–164, 2023, doi: 10.3991/ijoe.v19i09.39683.
- [9] K. Rajwar, K. Deep, and S. Das, *An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges*, vol. 56, no. 11. Springer Netherlands, 2023, doi: 10.1007/s10462-023-10470-y.
- [10] N. H. Abdul Aziz, Z. Ibrahim, S. Razali, and N. A. Ab Aziz, "Estimation-based Metaheuristics: A New Branch of Computational Intelligence," *Natl. Conf. Postgrad. Res. 2016(NCON-PGR)*, pp. 469–476, 2016.
- [11] R. Toscano and P. Lyonnet, "Heuristic kalman algorithm for solving optimization problems," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 39, no. 5, pp. 1231–1244, 2009, doi:10.1109/TSMCB.2009.2014777.
- [12] Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, and M. S. Mohamad, "Simulated Kalman Filter: A novel estimation-based metaheuristic optimization algorithm," *Adv. Sci. Lett.*, vol. 22, no. 10, pp. 2941–2946, 2016, doi: 10.1166/asl.2016.7083.
- [13] N. H. Abdul Aziz, Z. Ibrahim, N. A. Ab Aziz, M. S. Mohamad, and J. Watada, "Single-solution Simulated Kalman Filter algorithm for global optimisation problems," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 43, no. 7, 2018, doi: 10.1007/s12046-018-0888-9.
- [14] T. Ab Rahman, Z. Ibrahim, N. A. Ab Aziz, S. Zhao, and N. H. Abdul Aziz, "Single-Agent Finite Impulse Response Optimizer for Numerical Optimization Problems," *IEEE Access*, vol. 6, pp. 9358–9374, 2018, doi: 10.1109/ACCESS.2017.2777894.
- [15] T. Ab Rahman, N. A. Ab Aziz, Z. Ibrahim, N. H. A. Aziz, M. I. Shapiai, and J. S. Suri, "Multi-Agent Finite Impulse Response Optimizer for Numerical Optimization Problems," *ResearchSquare*. 2020, doi: 10.21203/rs.3.rs-129794.
- [16] Z. Musa, Z. Ibrahim, M. I. Shapiai, and Y. Tsuboi, "Cubature Kalman Optimizer: A Novel Metaheuristic Algorithm for Solving Numerical Optimization Problems," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 33, no. 1, pp. 333–355, 2023, doi: 10.37934/araset.33.1.333355.
- [17] Z. Musa, Z. Ibrahim, and M. I. Shapiai, "Multi-Agent Cubature Kalman Optimizer: a novel metaheuristic algorithm for solving numerical optimization problems," *Int. J. Cogn. Comput. Eng.*, vol. 5, no. Mar. 2023, pp. 140–152, 2024, doi: 10.1016/j.ijcce.2024.03.003.
- [18] B. Morales-Castañeda, E. Cuevas, A. Rodríguez, F. Fausto, and D. Zaldivar, "A better balance in metaheuristic algorithms: Does it exist?," *Swarm and Evolutionary Computation*, vol. 54, p. 100671, Mar. 2020, doi: 10.1016/j.swevo.2020.100671.
- [19] C. Zhang *et al.*, "An adaptive balance optimization algorithm and its engineering application," *Advanced Engineering Informatics*, vol. 55, p. 101908, Jan. 2023, doi: 10.1016/j.aei.2023.101908.
- [20] R. García-Ródenas, L. J. Linares, and J. A. López-Gómez, "A Memetic Chaotic Gravitational Search Algorithm for unconstrained global optimization problems," *Applied Soft Computing*, vol. 79, pp. 14–29, Mar. 2019, doi: 10.1016/j.asoc.2019.03.011.
- [21] J. Jiang, X. Yang, X. Meng, and K. Li, "Enhance chaotic gravitational search algorithm (CGSA) by balance adjustment mechanism and sine randomness function for continuous optimization problems," *Physica A: Statistical Mechanics and its Applications*, vol. 537, p. 122621, Sep. 2019, doi: 10.1016/j.physa.2019.122621.
- [22] Z. Tang, S. Tao, K. Wang, Y. Todo, J. Shi, and S. Gao, "A Novel Optimization Algorithm Inherited From Gravitational and Spherical Search Dynamics," Dec. 2020, pp. 91–96. doi:10.1109/iscid51228.2020.00027.
- [23] L. Hayward and A. Engelbrecht, "How to Tell a Fish from a Bee: Constructing Meta-Heuristic Search Behavior Characteristics," *GECCO 2023 Companion - Proc. 2023 Genet. Evol. Comput. Conf. Companion*, pp. 1562–1569, 2023, doi: 10.1145/3583133.3596338.
- [24] H. Hendy, M. I. Irawan, I. Mukhlash, and S. Setimin, "A Bibliometric Analysis of Metaheuristic Research and Its Applications," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 9, no. 1, pp. 1–17, 2023, doi:10.26594/register.v9i1.2675.
- [25] M. A. Lones, "Mitigating Metaphors: A Comprehensive Guide to Recent Nature-Inspired Algorithms," *SN Comput. Sci.*, vol. 1, no. 1, pp. 1–24, 2020, doi: 10.1007/s42979-019-0050-8.
- [26] Z. Ibrahim *et al.*, "A Kalman filter approach for solving unimodal optimization problems," *ICIC Express Lett.*, vol. 9, no. 12, pp. 3415–3422, 2015.
- [27] N. H. Abdul Aziz *et al.*, "Simulated Kalman Filter with Randomized Q and R Parameters," *Proc. Int. Conf. Artif. Life Robot.*, vol. 22, no. Icarob 2017, pp. 711–714, 2017, doi: 10.5954/icarob.2017.gs11-6.
- [28] N. H. Abdul Aziz, Z. Ibrahim, S. Razali, T. Adeola Bakare, and N. A. Ab Aziz, "How Important the Error Covariance in Simulated Kalman Filter?," *Natl. Conf. Postgrad. Res. 2016*, pp. 315–320, 2016.
- [29] N. H. Abdul Aziz, Z. Ibrahim, N. A. Ab. Aziz, and S. Razali, "Parameter-Less Simulated Kalman Filter," *Int. J. Softw. Eng. Comput. Syst.*, vol. 3, no. February, pp. 129–137, 2017, doi:10.15282/ijsecs.3.2017.9.0031.
- [30] J. Kudela and R. Matousek, "New Benchmark Functions for Single-Objective Optimization Based on a Zigzag Pattern," *IEEE Access*, vol. 10, pp. 8262–8278, Jan. 2022, doi: 10.1109/access.2022.3144067.
- [31] I. R. Meneghini, M. A. Alves, A. Gaspar-Cunha, and F. G. Guimarães, "Scalable and customizable benchmark problems for many-objective optimization," *Applied Soft Computing*, vol. 90, p. 106139, Feb. 2020, doi: 10.1016/j.asoc.2020.106139.
- [32] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194, 2013, doi:10.1504/ijmmno.2013.055204.
- [33] A. Bolufe-Rohler and S. Chen, "A Multi-Population Exploration-only Exploitation-only Hybrid on CEC-2020 Single Objective Bound Constrained Problems," Jul. 2020. doi:10.1109/cec48606.2020.9185530.
- [34] N. H. Abdul Aziz *et al.*, "A Tutorial on Population-based Simulated Kalman Filter," *Mekatronika*, vol. 1, no. 2, pp. 23–32, 2019, doi:10.15282/mekatronika.v1i2.4895.
- [35] B. Crawford *et al.*, "Q-Learnheuristics: Towards Data-Driven Balanced Metaheuristics," *Mathematics*, vol. 9, no. 16, p. 1839, Aug. 2021, doi: 10.3390/math9161839.
- [36] S. M. Ebrahimi, S. Hasanzadeh, and S. Khatibi, "Parameter identification of fuel cell using Repairable Grey Wolf Optimization algorithm," *Applied Soft Computing*, vol. 147, p. 110791, Sep. 2023, doi: 10.1016/j.asoc.2023.110791.
- [37] N. Hidayati *et al.*, "A brief review of simulated Kalman Filter Algorithm –variants and applications," *F1000Research 2021 101081*, vol. 10, p. 1081, Oct. 2021, doi:10.12688/f1000research.73242.1.
- [38] K. Lazarus *et al.*, "An opposition-based simulated kalman filter algorithm for adaptive beamforming," *Proc. 2017 IEEE Int. Conf. Appl. Syst. Innov. Appl. Syst. Innov. Mod. Technol. ICASI 2017*, no. 4, pp. 91–94, 2017, doi: 10.1109/ICASI.2017.7988354.
- [39] B. Muhammad, Z. Ibrahim, M. I. Shapiai, M. S. Mohamad, K. Z. M. Azmi, and M. F. M. Jusof, "Oppositional learning prediction operator with jumping rate for simulated kalman filter," *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 0–5, 2019, doi:10.1109/ICCISci.2019.8716382.
- [40] Z. Ibrahim *et al.*, "An oppositional learning prediction operator for simulated kalman filter," in *Proceedings - 3rd International Conference on Computational Intelligence and Applications, ICCIA 2018*, 2018, pp. 195–199, doi: 10.1109/ICCIA.2018.00044.
- [41] B. Muhammad *et al.*, "A new hybrid simulated Kalman filter and particle swarm optimization for continuous numerical optimization

- problems,” *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 22, pp. 17171–17176, 2015.
- [42] B. Muhammad, Z. Ibrahim, M. F. Mat Jusof, N. A. Ab Aziz, N. H. Abd Aziz, and N. Mokhtar, “A Hybrid Simulated Kalman Filter - Gravitational Search Algorithm (SKF-GSA),” *Proc. Int. Conf. Artif. Life Robot.*, vol. 22, no. 1, pp. 707–710, 2017, doi:10.5954/icarob.2017.gs11-5.
- [43] B. Muhammad *et al.*, “Four Different Methods to Hybrid Simulated Kalman Filter (SKF) with Particle Swarm Optimization (PSO),” in *The National Conference for Postgraduate Research 2016*, 2016, pp. 843–853.
- [44] B. Muhammad *et al.*, “Four Different Methods to Hybrid Simulated Kalman Filter (SKF) with Gravitational Search Algorithm (GSA),” in *The National Conference for Postgraduate Research 2016*, 2016, pp. 854–864.
- [45] B. Muhammad *et al.*, “Performance Evaluation of Hybrid SKF Algorithms : Hybrid SKF-PSO and Hybrid SKF-GSA,” *Natl. Conf. Postgrad. Res.* 2016, pp. 865–874, 2016.
- [46] M. F. Mat Jusof *et al.*, “A Kalman-Filter-Based Sine-Cosine Algorithm,” in *2018 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS 2018)*, 2018, pp. 137–141, doi: 10.1109/I2CACIS.2018.8603711.
- [47] K. Z. Mohd Azmi, Z. Ibrahim, D. Pebrianti, M. F. Mat Jusof, N. H. Abdul Aziz, and N. A. Ab.Aziz, “Enhancing simulated Kalman filter algorithm using current optimum opposition-based learning,” *J. Intell. Manuf. Mechatronics*, vol. 01, no. 01, pp. 1–13, 2019, doi:10.15282/mekatronika.v1i1.157.
- [48] N. Ahmad Zamri, N. A. Nor, T. Bhuvaneswari, N. H. Abdul Aziz, and A. K. Ghazali, “Feature Selection of Microarray Data Using Simulated Kalman Filter with Mutation,” *Processes*, vol. 11, no. 8, 2023, doi:10.3390/pr11082409.
- [49] M. F. Mat Jusof *et al.*, *Simulated Kalman Filter Algorithm with Improved Accuracy*, vol. 538. Springer Singapore, 2019.
- [50] N. A. Ab. Aziz, Z. Ibrahim, N. H. A. Aziz, and T. Ab. Rahman, “Asynchronous simulated Kalman filter optimization algorithm,” *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 44–49, 2018, doi:10.14419/ijet.v7i4.27.22478.
- [51] N. A. A. Aziz, Z. Ibrahim, N. H. A. Aziz, M. Mubin, N. Mokhtar, and M. I. Shapiai, “A fitness-based adaptive synchronous-asynchronous switching in simulated kalman filter optimizer,” 2019, doi:10.1109/ICCISci.2019.8716393.
- [52] N. A. Ab. Aziz, T. Ab Rahman, and N. H. Abdul Aziz, “Fitness-evaluated Adaptive Switching Simulated Kalman Filter Algorithm with Randomness,” *Mekatronika*, vol. 1, no. 2, pp. 45–65, 2019, doi:10.15282/mekatronika.v1i2.4986.
- [53] N. A. A. Aziz *et al.*, *A Diversity-Based Adaptive Synchronous-Asynchronous Switching Simulated Kalman Filter Optimizer*, vol. 632. 2020.
- [54] M. Sharma and P. Kaur, “A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem,” *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1103–1127, Feb. 2020, doi: 10.1007/s11831-020-09412-6.
- [55] R. A. Briers and P. H. Warren, “Population turnover and habitat dynamics in Notonecta (Hemiptera: Notonectidae) metapopulations,” *Oecologia*, vol. 123, no. 2, pp. 216–222, May 2000, doi:10.1007/s004420051008.
- [56] A. Kaveh and H. Yousefpour, “Comparison of Three Chaotic Meta-heuristic Algorithms for the Optimal Design of Truss Structures with Frequency Constraints,” *Periodica Polytechnica Civil Engineering*, Jul. 2023, doi: 10.3311/ppci.22594.
- [57] Y. Zhang, Z. Lei, Z. Tang, Y. Todo, Z. Zhang, and S. Gao, “A Spherical Search-based Archive Update Mechanism for Self-adaptive Differential Evolution,” Mar. 2020, pp. 173–178. doi:10.1109/icaais49377.2020.9194937.
- [58] Q. Li, W. Gao, and Y. Bai, “Improved Initialization Method for Metaheuristic Algorithms: A Novel Search Space View,” *IEEE Access*, vol. 9, pp. 121366–121384, Jan. 2021, doi:10.1109/access.2021.3073480.