International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 26-33, September-2026

https://doi.org/10.47001/IRJIET/2026.109003

Comparative Evaluation of Evolutionary Algorithms for PV Maintenance Scheduling: Application to Najaf Climate

Mustafa Husham Abbas

College of Health and Medical Techniques, Al-Furat Al-Awsat Technical University, 54001 Najaf, Iraq

E-mail: mustafa.abbas@atu.edu.iq, ORCID: 0009-0000-0227-7466

***Abstract* - Hot, Arid climates lead to performance drops of Renewable energy assets with heat problems and dust buildup. In one year. This makes a scheduling problem with 2 to the power of 365 possible schedules. Previous studies have explored these through experiments or a single kind of optimization trick. They often don't compare their results with statistics so it is hard to say which way really works better. The objective of this paper is to find the way to search when there is a fixed amount of time to check things. It compares four methods, a Genetic Algorithm, a sigmoid based Binary Particle Swarm Optimization, a newer V-shaped BPSO and a random search. Each of these methods were tested 30 times. Each had a budget of 6,000 checks. The data used were collected from 4215 days of weather data in Najaf, Iraq. It also contained heat loss and dust models. The Genetic Algorithm resulted in the average profit of 94.22 with a slight variation. The next one was V-shaped BPSO with 33.44. The sigmoid BPSO and the random search lost money with -98.23 and -130.04, respectively. A Mann-Whitney U test showed that the Genetic Algorithm performed much better than the others and this result is not due to chance. Additional robustness tests indicated that the difference between BPSO and GA was decreased when a V-shaped transfer function was used instead of the Sigmoid. It made GA more profitable reaching 140.40 ± 0.31 when the number of evaluations was increased to 30,000. Sensitivity analyses showed that the ranking was robust to changes in energy prices and cleaning costs. For another loss model including heat-dust interactions, the results of the algorithms did not change much. GAs schedule focuses on cleaning in the months with the most dust build up. This provides a real-world test of the results. This research offers proven advice for choosing algorithms for similar scheduling problems, not only in the solar energy sector.**

Keywords: Genetic algorithm, Particle swarm optimization, Combinatorial optimization, Maintenance scheduling, Photovoltaic soiling, Renewable energy.

I. INTRODUCTION

This paper discusses a problem that photovoltaic systems face in hot and dusty areas and that is the loss of energy caused by the high temperatures and the dust accumulation. Solar panels become less efficient when they get too hot. This is called derating and it occurs when the temperature is above the standard 25°C used for testing [7] [8]. At the time dust gathers on the surface of the panels. This is called soiling [1] [5]. Both issues cause the system to generate power. Keeping the panels clean mitigates this loss, but cleaning them costs in labor and water. These costs mean that when to clean the panels is not a question of keeping them clean. It's all about getting the timetable. The aim is to increase revenue, not reduce cleaning costs or eliminate dirt. This goal makes the problem harder than it seems. The study transforms the problem into a schedule with 365 choices, one for each day of the year. Every choice is either clean the panels or not clean. That gives you 2^365 schedules. That is a number so large that checking every schedule is impossible. Random guessing would be too slow and too demanding on the computer in real time. The study solves this problem by using combination methods that can find schedules without testing all the possibilities. The research uses swarm intelligence techniques and specific algorithms, such as Genetic Algorithms and Particle Swarm Optimization. These methods employ simulation of processes (evolution, group behavior) to rapidly search large sets of solutions. Most of the work done on the degradation of solar panel in different climates considers how much power is lost because of heat and dust. Most of those studies are physics-based. They rarely tell you when to clean. Genetic Algorithms or Particle Swarm Optimization are often used on the side in metaheuristic research for energy problems. Usually, these studies do not compare the algorithms at all. Usually, each algorithm is given a computer time. It is difficult to determine if one algorithm is better because of its method or just because it had more time. A fair comparison is when all algorithms are given the evaluation budget. The paper contributes three things to scheduling panel maintenance. It first formulates the cleaning problem as an optimization problem. It creates a fitness function that

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 26-33, September-2026
https://doi.org/10.47001/IRJIET/2026.109003

incorporates conditions such as weather and operating constraints. Secondly, it tests three strategies; a Genetic Algorithm, a Binary PSO adapted to this problem and a random search. All 3 are tested under the fitness evaluation limit. This guarantees that any performance differences are due to the search method itself, not the computer power. Third, it shows the model on a case study with satellite data. It uses data from the NASA POWER data base and MODIS aerosol optical depth measurements. The study area is Najaf city, Iraq. This is a location that has not been investigated for this problem before. The results are specific to the maintenance of solar panels, but the framework developed here can be applied to scheduling problems with a multitude of binary decisions over time. The problem structure and its equations can be used to solve problems in logistics, operations, or planning.

II. RELATED WORK

Dust- and heat-related PV degradation has received sustained attention from Iraqi and regional researchers. Dahham *et al.* [1] developed a particle-force-balance model of dust accumulation on PV surfaces and quantified wind-driven scraping effects using data from a research site in Najaf, Iraq, finding performance-enhancement percentages of 4.85-10.9% attributable to wind, depending on dust type. Jendar *et al.* [2] conducted a five-month field study in north-east Iraq and showed that short-circuit-current-based soiling measurements underestimate true power loss by approximately 8%. Chaichan *et al.* [4] compared dust composition and its photovoltaic impact across four Iraqi regions, including the Karbala-Najaf desert corridor. Abdulrazzaq *et al.* [3], studying Baghdad in summer 2022, reported a counterintuitive interaction in which dust storms coincide with temporary temperature drops, partially offsetting thermal losses at extreme temperatures — an interaction not captured by simple additive degradation models, including the one used as this paper's case-study environment. Saidan et al. [5] and Kasim [6], among others, provide earlier experimental baselines for dust-driven degradation and mitigation (including two-axis tracking) in Baghdad. None of these studies formulates the resulting maintenance decision as a combinatorial optimization problem, nor do they compare search algorithms under a controlled evaluation budget. This is the gap the present paper addresses: it treats the physical degradation model as a fixed, cited environment and focuses the contribution on the algorithmic question of how best to search the resulting decision space.

III. CASE-STUDY ENVIRONMENT

3.1 Data Source

The case-study environment uses 4,215 days (1 January 2015 - 1 August 2026) of daily data for Najaf, Iraq (32.03°N, 44.34°E), merged from two NASA sources, summarized in Table 1.

Table 1: Data Sources Used to Construct the Fitness Function

Variable	Source / Product	Role in Fitness Function
T2M, T2M_MAX	NASA POWER (daily point API)	Cell temperature -> thermal derating
RH2M, WS2M	NASA POWER (daily point API)	Recorded for case-study characterization
AOD (550 nm, Deep Blue)	MODIS/Aqua MYD08_D3.061 (Giovanni)	Daily soiling-accumulation category
ALLSKY_SFC_SW_DWN	NASA POWER (daily point API)	Available daily energy (kWh/m^2/day)

The AOD record (MODIS/Aqua, Deep Blue algorithm, land-only, 550 nm) has a mean value of 0.450 (std. 0.261) with 14.9% of days missing due to cloud cover or retrieval quality flags; missing days are excluded from the climatological profile used to build the representative test year.

3.2 Physical Degradation Model

The Nominal Operating Cell Temperature (NOCT) approximation allows an estimation of the cell temperature from the maximum daily ambient temperature. The calculation is based on the formula T_cell = T_ambient + (NOCT - 20) x (G / 800), with NOCT = 45°C and the reference irradiance G = 1000 W/m2 [7]. The thermal efficiency loss is modelled by the standard linear temperature coefficient model, L_heat = max (0, β × (T_cell - 25°C)) × 100%, with β = 0.4%/°C, in the range 0.29-0.52%/°C reported for crystalline-silicon modules in the literature [7], [8]. The daily soiling accumulation rate is assigned to one of four tiers based on AOD (0.05, 0.15, 0.3, and 0.6 %/day for AOD < 0.3, < 0.5, < 1.0, and ≥ 1.0, respectively) of a magnitude consistent with the directly measured range of 0.15-0.3%/day in northeast Iraq [2]. Soiling loss grows continuously over days, resetting to zero only on a chosen cleaning day, and is bounded at 25% reflecting the saturating soiling behaviour found experimentally [5]. Total daily loss = thermal loss + soiling loss, and daily net energy value = (available irradiance) x (1 - total loss) x (energy price), with a fixed cost associated with cleaning events. All reported profit figures are based on an illustrative 1 kW

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 26-33, September-2026

https://doi.org/10.47001/IRJIET/2026.109003

reference system and on illustrative unit costs (energy price \$0.10/kWh, cleaning cost \$2/event); they are not fitted to a specific local tariff and should be replaced with site-specific values for deployment-scale decisions. This model does consider heat and dust loss to be additive and it does not include the interaction that offsets them at extreme temperatures as reported by Abdulrazzaq *et al.* [3]. This is mentioned as a limitation in Section VII. A climatological profile for a representative year is created by averaging each environmental input for each day of the year over all available years. This leads to a continuous 366-day profile providing a uniform fitness evaluation environment for the three algorithms presented in the paper.

IV. PROBLEM FORMULATION

Let me explain the decision variable, in this model. The decision variable is a vector $x \in \{0,1\}^{366}$. This binary vector shows whether each day d is chosen as a cleaning day. If $x_d = 1$ then the day is a cleaning day. The objective function is designed to maximize profit.

$$f(x) = \Sigma_d [P_{kW} \times G_d \times (1 - L_d(x))] \times price\_energy - (\Sigma_d x_d) \times cost\_clean \quad (1)$$

Here, G_d indicates the available solar irradiance for day d, and L_d(x) denotes the accumulated loss on that day, influenced by the cleaning decisions made on previous days. This setup highlights the path-dependent nature of soiling and the complexity associated with the function f, which is non-separable across the 366 days. Therefore, a holistic approach is required instead of analyzing each day in isolation, as it entails navigating a 366-dimensional binary problem. This complexity emphasizes the significance of the algorithmic comparisons described in Section V.

V. OPTIMIZATION ALGORITHMS

All four methods utilize the same 366-dimensional binary representation and are evaluated using the identical protocol outlined in Section V.E.

5.1 Genetic Algorithm (GA) [9]

A population of 30 candidate schedules is randomly initialized and rigorously evaluated over 200 generations. In each generation, the top half, termed elite individuals, are preserved for subsequent generations. The remaining candidates are generated through single-point crossover involving two randomly chosen elite parents. Additionally, a bit-flip mutation is implemented on each gene of the offspring independently, with a mutation probability set at 1/366, resulting in an average of one mutation per offspring.

5.2 Binary PSO (BPSO) [10], [11]

The swarm of thirty particles initialized with binary positions and continuous velocities in between minus one and plus one. The swarm updates the velocity of each particle according to the rules of Particle Swarm Optimization for more than two hundred iterations. The inertia weight of the swarm in this training is 0.6. Sets the cognitive and social coefficients to 1.5. The velocity updates are intended to move each particle toward its best position and toward the global best position of the swarm. To avoid saturation of the sigmoid transfer function, the swarm limits each velocity to the range. Six, add six. After updating the velocity, the swarm resamples the position of each particle setting the position to one with a probability based on the sigmoid of that particle's velocity using the Binary PSO transfer function model from 1997. As stated in Section VI-D, the S-shaped sigmoid transfer function in Binary PSO is a known limitation.

5.3 V-Shaped Binary PSO (V-BPSO) [12]

Mirjalili and Lewis [12] identified a limitation in Binary Particle Swarm Optimization (BPSO) due to S-shaped transfer functions, like the sigmoid function, which constrain particles to binary states (0 or 1) during iterations, potentially negating beneficial swarm structures. To overcome this limitation, they proposed V-shaped transfer functions that enable probabilistic toggling of a particle's state, enhancing stability when the likelihood of change is low. To evaluate this new method in conjunction with Genetic Algorithms (GA), a modified version of BPSO was created. This variant maintains the previous framework from Section V.B but employs a V-shaped transfer function defined as $V(v) = |\tanh(v)|$. In this model, a bit is toggled based on a probability dictated by V(velocity) rather than being resampled directly, with all other parameters, such as swarm size, inertia, coefficients, and evaluation budget, held constant to ensure the transfer function is the only differing variable in the analysis.

5.4 Random Search and Evaluation Protocol

We took totally 6000 uniformly random binary vectors. The function f (x) value was kept for analysis. This baseline sampling technique demonstrates the effect of unguided sampling under a fixed budget. The study evidences the importance of this baseline, as it is part of the verification of any benefit coming from the use of a metaheuristic approach. This check is necessary to verify that observed benefits are a result of the method itself and not a budget that is used only for sampling. See details in Section VI of the study. All algorithms were tested under a budget of 6,000 function evaluations per run. This design enables us to see performance variations based on search strategies, not on different amounts of computation. Each algorithm was run 30 times with a

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 26-33, September-2026

https://doi.org/10.47001/IRJIET/2026.109003

random seed from 0 to 29. This reduces the effect of random starting points on the outcomes. The results are reported as values and standard deviations over these runs. We used statistical significance tests with the Mann Whitney U test, a non-parametric approach that has proved useful for the comparison of stochastic optimization algorithms. We also performed a one factor at a time sensitivity analysis. For each scenario, we performed 5 repetitions, to investigate the ranking of the algorithms under two economic factors, energy pricing and cleaning costs. Finally, we performed an experiment with 10 independent runs for each algorithm with a larger budget of 30,000 evaluations to confirm that the rankings obtained from the 6,000-evaluation budget are stable.

VI. RESULTS AND DISCUSSION

6.1 Main Comparison (30 Independent Runs)

Table 2 presents the mean and standard deviation values for the optimal schedule identified by each algorithm, based on 30 independent experimental runs. These runs utilized varying random seeds, all executed within a comprehensive evaluation budget of 6,000 evaluations.

Table 2: Results across 30 independent runs, equal 6,000-evaluation budget each

Algorithm	Mean Net Profit ± SD	Min / Max	n
Genetic Algorithm	**94.22 ± 8.36**	72.70 / 113.60	30
V-Shaped BPSO	33.44 ± 10.93	5.30 / 52.62	30
Binary PSO (sigmoid)	-98.23 ± 4.75	-105.79 / -83.93	30
Random Search	-130.04 ± 5.38	-139.72 / -115.80	30

The Genetic Algorithm shows results than other ways of finding the best solution. It gets an average profit than the V-Shaped Binary Particle Swarm Optimization. The profit is 60.78 units higher. On the hand the original sigmoid-based BPSO and the random search both have negative average profits. This shows that the Genetic Algorithm works better. The Genetic Algorithm is 192.45 units than the sigmoid BPSO and 224.26 units better than the random search. This shows how good the Genetic Algorithm is at making profit. Also Figure 1 shows a picture of how the Genetic Algorithm, the sigmoid BPSO and the random search all improve over time. It shows how the best profit changes as the number of fitness evaluations increases. This again shows how good the Genetic Algorithm is, at making the best results compared to the others.

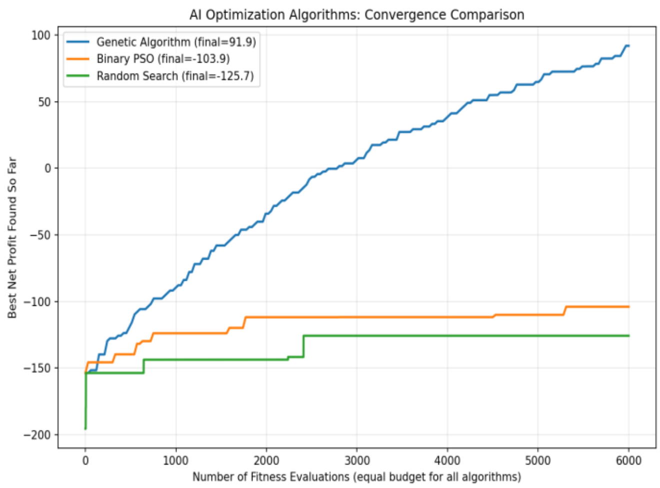

Figure 1: Convergence comparison: best net profit found so far vs. number of fitness evaluations, for an identical 6,000-evaluation budget shared by all three primary methods

I notice that the paper reports two results on the performance of different scheduling algorithms. A previous test used a 12-dimensional model capable of generating 4,096 possible schedules. All three algorithms performed approximately the same in that search space. Sometimes random search surprisingly. Overcome the more complex metaheuristic approaches when the computer had enough time. This is because of the limited space where the random search can try the possibilities so that the random search can catch up to the GA and PSO. In comparison, the present work employed a more complex 366-dimensional daily model. This was done to show how important the algorithm choice is and Table II displays the results clearly. The search has a flat convergence curve as shown in Fig. 1Tell us that the random search does not really get better from its initial guesses.

The results show that GA outperforms BPSO due to the way each algorithm changes the values. The GA employs a single point crossover that swaps schedule blocks between two parents. This preserves patterns such as runs of cleaning and non-cleaning days, and aids the overall solution. Conversely, BPSO updates each of the 366 bits nearly independently, using a velocity and sigmoid formulation. That independence can break up patterns, which explains why BPSO levels off early as seen in Fig. 1. A side analysis, not part of the optimization, also shows that the cleaning days chosen by the GA tend to cluster in months with Aerosol Optical Depth (AOD) especially from March, to May. This clustering suggests that GA found not only a best schedule but also a schedule that is consistent with real seasonal trends in the case study data.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 9, pp 26-33, September-2026

https://doi.org/10.47001/IRJIET/2026.109003

6.2 Statistical Significance

A single time running an algorithm can give results that don't really show how good the algorithm is because of the randomness that comes from the starting point. To check if the results are reliable a Mann-Whitney U test was done on the data from 30 times the algorithm was run. The test looked at two comparisons: Genetic Algorithm (GA) against Binary Particle Swarm Optimization (BPSO) and GA against search. In both cases the highest U value of 900 was found. This shows that GA did better than the two methods every time across all pairs of starting points and there were 900 checks (30 pairs of starting points). The result was very important statistically ($p < 0.001$) with $p = 1.51 \times 10$^-11 for both comparisons. Also, BPSO did much better than search with the same highest U value and same level of importance. The box plot in Figure 2 backs up these findings showing that the results for the three methods are completely separate across the 30 runs. This confirms that the ranking is strong and not a result of randomness, from one run.

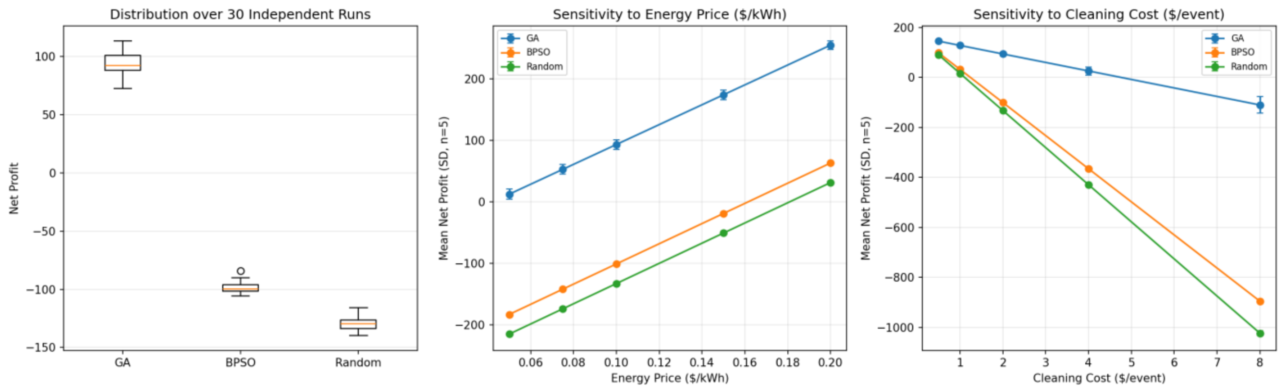

Figure 2: Left: distribution of final net profit over 30 runs per algorithm. Center/right: sensitivity of mean net profit to energy price and cleaning cost

6.3 Sensitivity Analysis

Section 3.B analyses a sensitivity check on energy prices and cleaning costs using a one-factor-at-a-time approach. The baseline values are set at $0.10 per kWh for energy and $2 per event for cleaning costs. These values are varied in sensitivity tests over ranges and five runs are made for each setting. The results indicate that with the increment of the energy prices from $0.05 to $0.20/kWh, the mean profits of the Genetic Algorithm (GA) increase from 12.78 to 255.12. In contrast, the profits from the Biologically-Inspired Particle Swarm Optimization (BPSO) and random search algorithms are still less than those of GA for all tested price points and the gap is increasing with the rise of energy prices. As we can see, as cleaning costs rise from $0.5 to $8/event, profits drop for all three algorithms, which is to be expected. With the cleaning cost of $8/event, the profit of GAs becomes negative at -110.44. However, it still outperforms BPSO and random search by 784.63 profit units and 912.55 profit units respectively. Thus, GAs are more successful not only because of favorable economics, but also because they schedule cleaning events. The benefits of GAs are inherent and not simply due to the specific example unit costs as GA is always better than BPSO and random search in all ten parameter combinations tested. This stability indicates the robustness and superiority of GA as discussed in algorithm comparisons in Section VI-A.

6.4 Effect of Transfer Function

The analysis shows that changing the transfer function in Binary Particle Swarm Optimization (BPSO) from the S-shaped (sigmoid) function to the V-shaped version proposed by Mirjalili and Lewis makes a difference. This change improves the result from -98.23 ± 4.75 to 33.44 ± 10.93. That's a gain of 131.67 units with all settings like swarm size, coefficients and budget staying the same. It suggests that earlier performance differences seen in BPSO were mostly due to flaws in the sigmoid transfer function, not any weakness in how particle swarm optimization works. I notice the sigmoid function forces each of the 366 bits to become either zero or one at every step. I see this often wipes out solutions that had already formed breaking up optima. The V-shaped function avoids this problem by using a complement-with-probability rule. It changes bits gently so existing good solutions remain stable. This matches better with the nature of the problem described in Section 4. With these improvements Genetic Algorithms (GA) still perform better than V-BPSO under the test conditions. A Mann-Whitney U test gives a value of nine hundred and a p-value of 1.51×10^-11. This separation is based on samples of size thirty, in each group. Therefore, GA wins statistically. The new transfer function helps a lot. GA does not remove the edge that GA holds.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 9, pp 26-33, September-2026
https://doi.org/10.47001/IRJIET/2026.109003

6.5 Effect of Evaluation Budget

After the evaluation of the Genetic Algorithm (GA) with the budget of 6,000 evaluations it was clear that the Genetic Algorithm (GA) could be improved. This implied that a larger budget of 30,000 evaluations was spent. To keep the time reasonable, we did 10 independent runs of each algorithm. The results show that the mean profit by Genetic Algorithm (GA) jumped from 94.22 ± 8.36 to 140.40 ± 0.31. That is a rise of 49%. Suggests that the Genetic Algorithm (GA) was previously underestimated. The small standard deviation at the larger budget 0.31 versus 8.36 shows that the Genetic Algorithm (GA) finds similar solutions independently of the starting point. The V-BPSO algorithm also has improved a lot. It rose from 33.44 ± 10.93 to 91.71 ± 6.48 (174%). This makes the V-BPSO performance more comparable to the GA. However, the sigmoid BPSO algorithm was slightly modified. It went from -98.23 to -83.97. Still in the black. The random search algorithm also changed a little from -130.04 to -125.41 showing that it does not work well in large search space. The results show that the order of performance is the same for both budgets: GA>V-BPSO>sigmoid BPSO>Random. For the increased budget, the profit difference between the Genetic Algorithm (GA) and the V-BPSO algorithm declined from 60.78 to 48.69 units. This means that if they tested more and had a bigger budget their profits could be even closer.

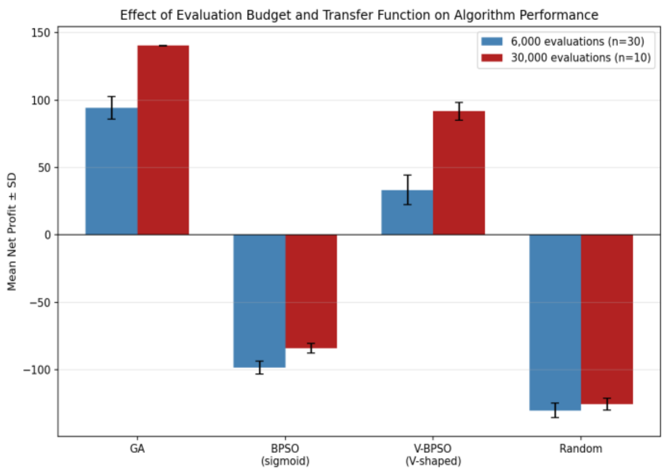

Figure 3: Mean net profit (± SD) at the original 6,000-evaluation budget (n=30) and an extended 30,000-evaluation budget (n=10)

6.6 Robustness to Physical Model Specification

The loss model used in this study ignores the heat-dust interaction that was described by Abdulrazzaq *et al.* These researchers showed that dust can cool modules in Baghdad during very hot summers. This cooling effect leads to energy loss than what happens with clean modules. To see how this simplification changes the results a new Model B was made. In Model B the rate at which dirt builds up is cut in half on days when the cell temperature goes over 70 degrees Celsius. This change is meant to show the findings in a way and not copy them exactly because the exact math was not provided. When Model B was tested four different algorithms were used under the conditions as explained in Section VI-A. The results for deviation were very similar to those from Model A. For the GA algorithm the numbers were 94.22±8.36 compared to 93.32±9.89. V-BPSO had 33.44±10.93 versus 35.80±9.20. Sigmoid-BPSO showed -98.23±4.75 against -97.73±4.68. The random algorithm gave -130.04±5.38 compared to -129.86±5.30. A statistical test called the Mann-Whitney U test showed that the results remained statistically significant (p=1.51×10^-11) for GA when compared to the algorithms in Model B. This same order—GA then V-BPSO, then sigmoid-BPSO then random—and the statistical significance show that the results are reliable. It proves that the findings are not just because of the loss function used originally but are also true when changes are made that include some real-world behavior, like dust cooling effects.

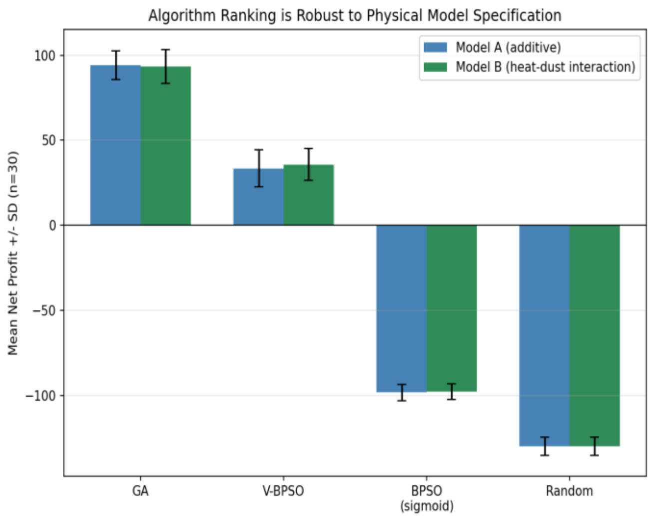

Figure 4: Mean net profit (± SD, n=30) under the additive loss model (Model A) and an alternative model incorporating a heat-dust interaction (Model B)

VII. LIMITATIONS AND FUTURE WORK

In Section 5-F the algorithmic ranking appears stable when one alternative to the loss model is used. However, this section does not examine the wind- and humidity-dependent soiling modifiers that were studied in a companion analysis. Testing those modifiers would be a next step to confirm robustness. The sensitivity analysis in Section VI-C changes two economic parameters at a time using a small 5-run sample. A full joint sensitivity study that uses the 30-run budget would make the economic robustness claim stronger. In the extended-budget comparison in Section VI-E the authors used 10 runs of 30 to keep runtimes practical. Figure 3 shows that GA and V-BPSO have not yet converged after

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 26-33, September-2026
https://doi.org/10.47001/IRJIET/2026.109003

30 000 evaluations. Therefore, the difference between these algorithms as the budget grows further is still unknown. The AOD record in this study comes from a satellite, Aqua/MYD08_D3. If we added the Terra product the 14.9 % missing-data rate caused by cloud cover would drop. Finally, the paper compares three variants to a random baseline. To build on this we should run a validated comparison that includes other methods such, as simulated annealing, ant colony optimization or a hybrid GA-PSO. We should keep the equal-budget, repeated-run protocol. Doing so and applying the problem set-up and algorithm comparison to other high-dimensional binary maintenance-scheduling problems outside of solar PV would be a natural next step.

VIII. CONCLUSIONS

This paper looked at renewable-energy maintenance scheduling as a dimensional binary combinatorial optimization problem. It compared a Genetic Algorithm with two Binary PSO transfer-function variants and a random-search baseline. All methods were tested under the evaluation budget repeated over 30 independent runs using a case-study environment built from real NASA meteorological and aerosol satellite data for Najaf, Iraq. The Genetic Algorithm achieved the mean net profit and significantly outperformed all other methods. This result was confirmed using the Mann-Whitney U test with a p-value than 0.001. A sensitivity analysis showed this ranking remained stable across different assumptions about energy prices and cleaning costs. The results also held when the original physical loss model was replaced with another one that included heat-dust interaction effects. Switching from the sigmoid transfer function in BPSO to a modern V-shaped alternative closed most of the performance gap between BPSO and the Genetic Algorithm. When the evaluation budget was increased five-fold the gap narrowed further while the Genetic Algorithm still maintained its lead. This suggests that implementation choices—like which transfer function is used—search budget size and how the physical loss model is specified all play roles in algorithm performance. Relying on one run one BPSO variant or one loss model could give misleading impressions about how good an algorithm really is. A smaller version of the problem with only 12 dimensions showed no meaningful differences between any of the methods. This indicates that algorithm choice matters mainly when the search space becomes large enough that unguided sampling stops being effective. These findings provide a statistically supported and reproducible method for choosing algorithms in high-dimensional binary scheduling problems. Renewable-energy maintenance serves as one example, among possible applications.

REFERENCES

[1] Dahham, I. A., Zainuri, M. A. A. M., Abdullah, A. A., & Fauzan, M. F., "Modeling the Effect of Dust and Wind Speed on Solar Panel Performance in Iraq," *Energies*, vol. 16, no. 17, p. 6397, 2023.

[2] Jendar, G. A., Hasnawi, L. A., Hassana, Q., Ceran, B., & Mohamad, A., "Experimental Investigation of Soiling and Temperature Impact on PV Power Degradation: North East-Iraq as A Case Study," *Diyala Journal of Engineering Sciences*, vol. 15, no. 4, 2022.

[3] Abdulrazzaq, O. A., Atewi, S. N., Tuama, R. A., Hasan, A. K., & Mohammed, M., "Revealing the Hidden Benefits of Dust for Photovoltaic Performance in Hot Climates," *Iraqi Journal of Industrial Research*, vol. 11, no. 1, 2024.

[4] Chaichan, M. T., Kazem, H. A., Al-Waeli, A. H. A., &Sopian, K., "The effect of dust components and contaminants on the performance of photovoltaic for the four regions in Iraq: a practical study," *Renewable Energy and Environmental Sustainability*, vol. 5, p. 3, 2020.

[5] Saidan, M., Albaali, A. G., Alasis, E., &Kaldellis, J. K., "Experimental study on the effect of dust deposition on solar photovoltaic panels in desert environment," *Renewable Energy*, vol. 92, pp. 499-505, 2016.

[6] Kasim, N. K., "New Technique for Treatment of the dust accumulation from PV solar panels surface," *Iraqi Journal of Physics*, vol. 8, no. 12, 2010.

[7] Faiman, D., "Assessing the outdoor operating temperature of photovoltaic modules," *Progress in Photovoltaics: Research and Applications*, vol. 16, no. 4, pp. 307-315, 2008.

[8] Mohammad, A. T., & Al-Shohani, W. A. M., "Numerical and experimental investigation for analyzing the temperature influence on the performance of photovoltaic module," *AIMS Energy*, vol. 10, no. 5, pp. 1026-1045, 2022.

[9] Holland, J. H., Adaptation in Natural and Artificial Systems, *University of Michigan Press, Ann Arbor*, 1975.

[10] Kennedy, J., & Eberhart, R., "Particle swarm optimization," *Proc. IEEE Int. Conf. on Neural Networks (ICNN'95)*, vol. 4, pp. 1942-1948, 1995.

[11] Kennedy, J., & Eberhart, R. C., "A discrete binary version of the particle swarm algorithm," *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 4104-4108, 1997.

[12] Mirjalili, S., & Lewis, A., "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013.

ISSN (online): 2581-3048
Volume 10, Issue 9, pp 26-33, September-2026
https://doi.org/10.47001/IRJIET/2026.109003

Citation of this Article:

Mustafa Husham Abbas. (2026). Comparative Evaluation of Evolutionary Algorithms for PV Maintenance Scheduling: Application to Najaf Climate. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(9), 26-33. Article DOI https://doi.org/10.47001/IRJIET/2026.109003

****