PARTICLE SWARM OPTIMIZATION WITH INDIVIDUALLY BIASED PARTICLES FOR RELIABLE AND ROBUST MAXIMUM POWER POINT TRACKING UNDER PARTIAL SHADING CONDITIONS

Baktybekov K.[1, 2], Kurmanbay A.*[1], Sakhanov K.[1], Syzdykov A.[1], Mukhamediyev A.[1]

[1]National Company "Kazakhstan Gharysh Sapary", Nur-Sultan, Kazakhstan, a.kurmanbay.sh@gmail.com
[2]National Research Nuclear University, MEPhI, Moscow, Russia

Efficient power control techniques are an integral part of photovoltaic system design. One of the means of managing power delivery is regulating the duty cycle of the DC to DC converter by various algorithms to operate only at points where power is maximum power point. Search has to be done as fast as possible to minimize power loss, especially under dynamically changing irradiance. The challenge of the task is the nonlinear behavior of the PV system under partial shading conditions. Depending on the size and structure of the photovoltaic panels, PSC creates an immense amount of possible P-V curves with numerous local maximums - requiring an intelligent algorithm for determining the optimal operating point. Existing benchmark maximum power point tracking algorithms cannot handle multiple peaks, and in this paper, we offer an adaptation of particle swarm optimization for the specific task.

Keywords: *maximum power point tracking, partial shading, meta-heuristics, particle swarm optimization.*

Introduction

Increasing interest in the solar industry and advances in computing capability led to the development of more forward-looking and efficient power control techniques previously ignored due to computational difficulty and challenges in testing. Maximum power point tracking related reliability issues was one of the tasks for solar technology to overcome and opportunity to gain 25-30% additional electrical efficiency [1]. To resolve this matter which might led to more than 70% power loss [2] - a myriad of MPPT procedures with different principles behind - ensuring operation on optimal points was developed [3 - 7]. Some of them owe their functionality to the tested by time control methods history of which links back to many decades if not centuries (calculus-based) [3 - 4]. While there are plenty of products of recent advances in computational science (artificial neural networks (ANN), meta-heuristics, Etc.) [6 - 7] In general, peak finding algorithms for MPPT could be classified into two distinct categories [8]: direct control (DIRC) and soft computing (SC) techniques. If the first mentioned methods use sampling or modulation based active control procedures, the later decides by exploring the performance of the system. DIRC approaches have serious limitations when it has to deal with non - linear behavior of the function and get easily trapped at local maxima. Such a shortcoming is extremely undesirable, as the output of photovoltaic systems shows a high level of non-linearity under non-uniform solar irradiance. Depending on the structure of the PV panel - more than n! (here n number of PV cells) different P-V curves could occur under operating conditions. Hence, MPPT requires a robust algorithm that has a general structure and able to deal with any output of the PV system in a timely manner. When calculus cannot provide means to achieve the desired solution without going through all search space - heuristics are used to obtain a more rapid result.

However, global optimization of function lacking convexity - remains an unsolved mathematical challenge. Monte Carlo based methods in the form of heuristics [9] cannot guarantee an exact solution every time, if at all. Nevertheless, they are proven to be the most superior technique existing today for dealing with non-convexity. One promising SC method is PSO nature-inspired algorithm [10] derived from the social behavior of birds, which relies on swarm intelligence to locate the global optimum. PSO has a beautiful structure that is easy to hybridize for further performance enhancement and adjustment to suit numerous tasks. In this paper, we present an adaptation of PSO to track the maximum power of the PV system. The reliability of the algorithm was extensively tested on various scenarios imitating real-life conditions.

The most common and well-known algorithms for MPPT are Perturb and Observe P&O [3] and Incremental Conductance INC [6] - both of which are direct control methods. The performance of new algorithms is usually compared to well established standard techniques. Furthermore, [11-13] mentioned that modifications done to the PSO should take into account the fundamental principle behind and provide numerical computations. However, under PSC conditions - benchmark MPPT algorithms (INC, P&O) don't function properly without improvements. Therefore, we use basic PSO as a reference to any additions done to the original recipe, and the enhancement procedure is constructed around physical and to a less extent - biological principles. Furthermore, we attempted to preserve the generality of the algorithm and focus on architectural improvements rather than parameter configuration so that procedure developed could be adapted to many other optimization and control procedures.

1. Modeling PV system

The response of the PV panel to varying environmental conditions is an objective function Equation 1 to the PSO algorithm which will sense the I-V and point to the optimal operational voltage by setting the duty cycle of the DC/DC converter Figure 1. Stated differently, the task of the controller is to locate an operational voltage under which the power output of the system is maximized.

$$max_x \, f(x) \quad \text{subject to} \quad x \in \chi \tag{1}$$

Modeling of the PV system response done as shown in [19] using the basic unit as a single-diode PV cell.
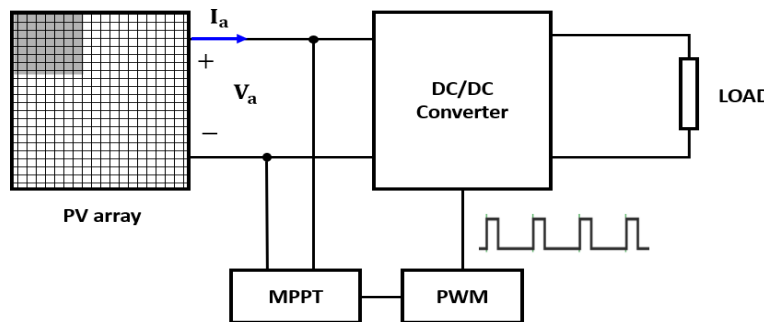

Fig.1. PV system simplified scheme

For more precise and comprehensive models we refer readers to [15]. Further analysis included simulation of the electrical circuitry of PV cells connected in a series-parallel configuration. Results are presented in the Figure 2.
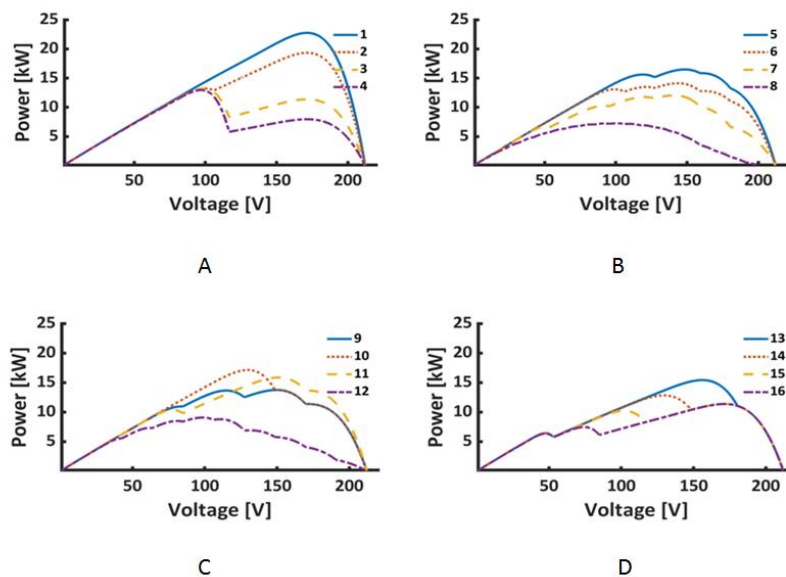

Fig.2. Possible outputs of PV array under PSC (test scenarios for PSO)

They were selected to represent wide variety of multi-modal situations that MPPT algorithm has to deal in real life conditions. Figure 2 A contains 4 curves and 1 curve is the result when the solar panel is not shaded, and the other curves are the result of partial shading. Figures 2 B, 2C and 2 D show the results of partial shading of the solar panel depending on the shading location on the panel. The number of possible peaks dictated by the number of PV elements connected, and takes place due to cells going out of operation range in a non- synchronous manner. Similarly to PSC conditions, non-uniform temperature also causes such peaks. A Significant reduction in electrical efficiency could take place if the MPPT algorithm gets trapped in one of the local maxima Figure 2. Test scenarios represent the output of 20 x 20 solar array connected in a series-parallel connection.

In a brief summary of characterizing the P - V curves of an (n x n) array, at most (n - 1) local maxima could be observed. Depending on temperature and irradiance non-uniformity, peaks might shift in any direction - making it hard to derive straightforward algorithms for MPPT. Although not always correct, what works for complex cases should operate well for more ordinary ones. Therefore, we selected a relatively big array to test the peak finding algorithm and extract difficulties it might face. As will be seen in the preceding sections, some PSC conditions cause more challenges for the MPPT algorithm rather than others. Therefore, to prove its validity - modeling and extraction of as many cases as possible stands extremely important. In most of the MPPT literature - the algorithm is tested on a relatively small array size where it might successfully determine the correct peak among few. However, the functionality of the algorithm can not be proven if not tested in extreme cases. Furthermore, in our testing's, we found that with increasing numbers of peaks, the complexity of tasks for algorithm rises as well.

2. Original PSO structure

Since the introduction to the scientific world, PSO found application in various engineering problems, and many modifications were offered by researchers. Before introducing our recipe, we go through a very brief review of the original PSO structure. Here our goal is to decompose the components of the algorithm for further analysis and adaptation to MPPT. PSO search for optimal solution by moving through search space with certain velocity as described in Equation 2 - 3:

$$v_i^{k+1} = w v_i^k + c_1 r_1 \left(P_{best} - x_i^k \right) + c_2 r_2 \left(G_{best} - x_i^k \right) \tag{2}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{3}$$

where: w – inertia; v^k - velocity of the i-th particle at iteration k; c_1 - cognitive acceleration constant; x^k - position of the i-th particle at iteration k; c_2 - social acceleration constant; P_{best} - best solution obtained so far by the particle; r_1, r_2 - random number in the range of [0, 1]; G_{best} - best solution obtained so far by the swarm

---

Algorithm 1 Original Particle Swarm Optimization

1: Initialize Particle population $\{ x_i \}, i \in [1, N]$
2: Initialize velocity of each particle $\{ v_i \}, i \in [1, N]$
3: Initialize the best-so-far position of each individual:, $i \in [1, N]$
4: Define Topology (**GBEST** topology)
5: Define social & cognitive acceleration constraints.
6: Define limits of velocity and position
7: While (k < maximum iteration set) Iterate
   through equations 2 - 3

2.1. Damped mass spring system analogy

Heuristic methods are inspired either by nature or physics [9]. In other words optimization problem is described as physical system where ground state needs to be found or as a biological system where fittest individuals needs to be described. For instance one of the most powerful technique in the arsenal of heuristic optimization tools simulated annealing [16] (SA) is imitation of physical process - cooling, while genetic

algorithms are derived from theory of evolution. Therefore, fundamental knowledge behind algorithm is a key for understanding it and altering it's behavior according to our needs. It was noticed that mechanics of particle movements could be described by stochastic damped mass spring system [17], which at each iteration has a center of oscillation around point Equation 4:

$$o_i^k = \frac{c_1 p_i^k + c_2 g_i^k}{c_1 + c_2} \tag{4}$$

We modify PSO according to this underlying physical principle and control exploration and exploitation characteristics of the algorithm which will be suitable for our problem.

2.2. Convergence analysis

Continuing on an assumption of stochastic damped mass spring analogy convergence analysis of the algorithm [12,13],[17], Etc. could be accomplished from control theory. Where stability of linear system could be confirmed if it lies within the first order stability region: Equation 5.

$$\{(\omega, \varphi) : |\omega| < 1, 0 < \varphi < 2(\omega + 1)\} , \tag{5}$$

where $\varphi = \frac{c_1 + c_2}{2}$

This knowledge allows to fine-tune parameters of the algorithm with guaranteed convergence and enhances understanding of the search mechanics. Furthermore, the concept of "center of mass" - opens doors to a more intelligent particle movement across search space and could be exploited to regulate the position and velocity of the particles.

2.3 Challenges in PSO algorithm

2.3.1. Premature convergence

In some instances due to limitation of search procedure, a feasible solution is not reachable, and Monte Carlo based methods converge on sub-optimal points. The exploration mechanism of the algorithm ensures that enough points were explored in the search space to conclude. However, in an extremely large search space for the algorithm, to converge it has to rely on collected information and execute decisions. Tricky entropy barriers or so-called golf-hole landscapes are known to cause challenges for heuristic techniques. In the case of PSO ensuring diversity of solutions is controlled by cognitive acceleration while the social coefficient assures exploitation. However, this is not only parameters that influence the mechanics of the particle movement, the architecture of the algorithm could be altered for additional enhancements. The problem of premature convergence is one of the primary directions in the research of PSO and an enormous number of solutions are offered in the literature by various researchers: swarms running on parallel [18],[7], introducing predators to the model [19], inserting invisible walls in search space [20], enforcing diversity [21,22] Etc. Ensuring the reliability and reproducibility of the control stands even more valuable when an algorithm is used in the engineering problem. Therefore, our main focus during this research is to test means of achieving a reliably functioning PSO and techniques which could enhance exploration characteristics of the PSO while preserving speed.

2.3.2. Complexity

Most of the benchmark algorithms (DIRC) used for the MPPT are simple and easy to realize. While for PSO - parameter selection needs to be accomplished for balanced exploration and exploitation. The task is usually achieved by another algorithm and optimization problem on its own. In literature, examples of training PSO by PSO or other Evolutionary algorithms to perform specific optimization tasks at an optimal time. However, such an option doesn't work for MPPT, as was mentioned before more than n! possible outcomes of the PV system need control. Therefore, promoting a necessity to preserve - general structure of algorithm, to find the peak of many possible outputs of the PV panels.

In addition to it, extra complexity of PSO is its architecture, which is harder to optimize rather than parameters. Population size, velocity and position limiting rules, information exchange policies, and many other details that make the algorithm very flexible with the vast capability and adaptability also make it complex to optimize.

3.PSO adaptation to MPPT

3.1.Particle initialization & landscape meshing.

The starting position of particles is known to affect the convergence of the PSO [18]. For instance, initialization using Voronoi tessellation reported improving the performance of the algorithm in terms of finding better solutions [23]. In the original PSO, particles are initialized randomly across the search space [10]. Thus, however, occasionally cause unwanted grouping of the particles in a specific section of the landscape, therefore, trap them in local minima or maxima. It is especially evident when fewer particles used. While more particles, in general, lead to an increase in reliability, it comes with a cost of calculation time. To avoid particle trapping into local maxima due to initialization, we decided to come up with a procedure that manually distributes particles uniformly. Although various methods of position initialization exist, we decided to stick on the simplest segmentation inspired by meshing in the field of numerical computations or cells of cellular automata.

Uniform initialization procedure - for the search space in the range of [0,X] each particle out of N is randomly initialized at the range of [0,X/N] , [X/N, 2X/N] ,etc. (2D example is shown in Figure 4) This is could be considered as (Manual enhancement to search) as particles are forced to be distributed across the search space, while in random initialization they might get stuck in specific region if the start was unlucky enough.
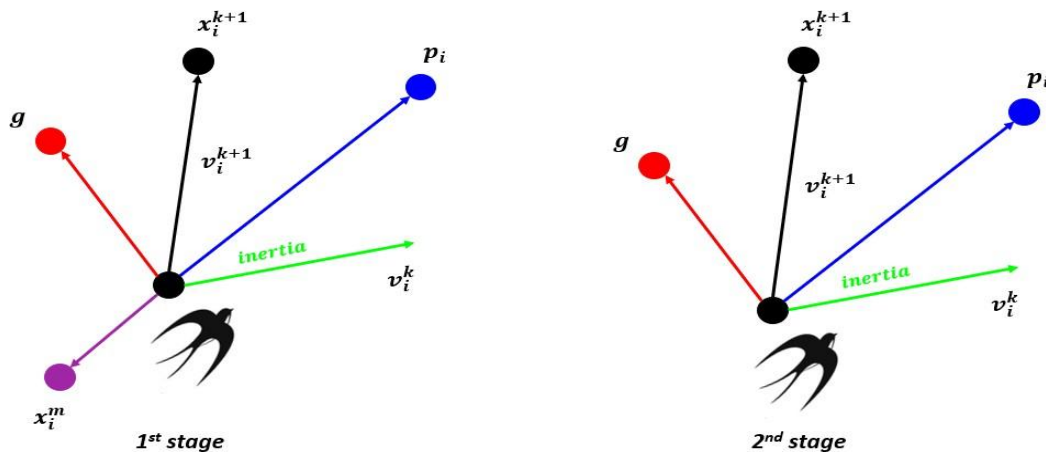

Fig.3. Visualization of the particle movement

Such an initialization could be effectively used further for creating diversity and warrant a uniform search of the landscape. Midpoints of segments, if utilized as an additional component - would give each particle an individuality bias. It will urge them to investigate the initialized section with higher probability - therefore, study its surroundings before converging on a specific spot. Equation 6:

$$v_i^{k+1} = wv_i^k + c_1 r_1 \left(P_{best} - x_i^k \right) + c_2 r_2 \left(G_{best} - x_i^k \right) + c_3 r_3 \left(X_i^m - x_i^k \right) , \tag{6}$$

where: c_3 - acceleration constant; r_3 - random number in the range of [0, 1]; X^m - midpoint of the initialization section of the particle.

Now after initialization the new center of oscillation at each iteration shifts to:

$$o_i^k = \frac{c_1 p_i^k + c_2 g_i^k + c_3 x_m^i}{c_1 + c_2 + c_3} \tag{7}$$

Our attempts could be compared to the stud like operator in genetic algorithms [24]. An additional term is not rare in PSO design - fully informed PSO [25] influenced by a neighborhood, and a global solution is known for delivering better results. Furthermore, [22] term introduced to push away particles from the best position produced additional diversity to the search. Our extension dedicated to enforcing uniform exploration of the search space. It also prevents unnecessary over-exploration of a specific region - which might lead to premature results and waste time. Put differently - it is a term that corresponds to the learning

of the swarm members about the landscape they are exploring and prioritizing the search space after necessary information was collected. Therefore, leaving behind unsuccessful search regions only after enough exploration. c_3 could be described by condition to segment the PSO into two distinct stages or phases where its behavior reshapes:

$$c_3 = \begin{cases} c_3, & if\ k < j \\ 0, & otherwise \end{cases}$$

(8)

During the first phase, we limit the dynamic range of particles by X_m^i, but with one eye in the global best solution. In the following stage, when enough information about the landscape was acquired - particles recombine in one segment to merge efforts and look nearby the best solution to possibly improve the result. Stability and convergence analysis is an essential part of the algorithm. However, for the initial stage — when the particles set to explore — we don't have to arrange acceleration parameters necessarily according to the condition in Equation 5. In contrast, configure them to warrant uniform and unbiased exploration.

The introduction of the new $_n$ term makes it possible to prioritize search space according to obtained results, and change the center of oscillation to particles located in low priority regions. Nevertheless to not make algorithm too complex for our application we didn't implement hierarchy or introduce diversity control mechanism to regulate c_3 or X_m^i like in [33-34] rather used $j = 18$ as a 18 iterations invested on reliability. Depending on function which will be optimized number might vary, while for our case starting from $j = 15$ we observed no failure. This is roughly the size of the tested array or # of possible peaks. c_3 was set to $c_3 = c_1 + c_2$. Although different interpretations of the biological principle behind PSO exists - the analogy between human communal activity, the social behavior of the birds, theory of evolution, Etc. We justify our additional term on the fact that no matter what will be the eventual destination of biological life, in the initial stage - it is bounded to the geography it was born and start exploring from what is reachable. In other words, diversity in nature is preserved by geographical fact as well.
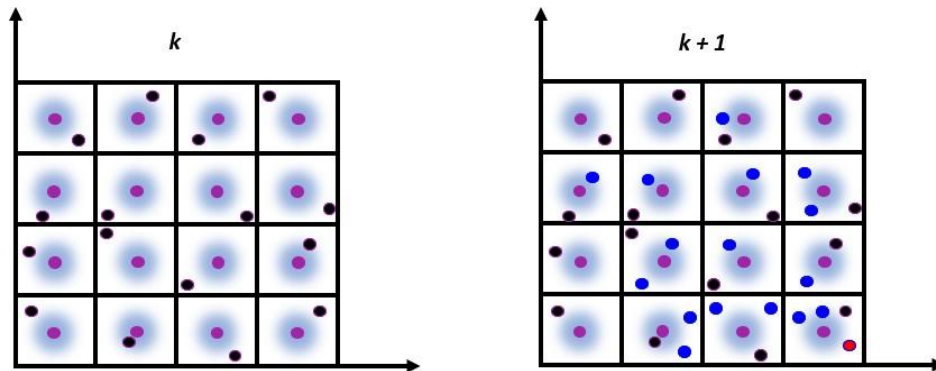

Fig.4. Visual representation of segmented initialization and dynamics of particle movement.

Moreover, the complex behavior of biological life forms probably cannot be comprehensibly described by two terms. Therefore, supplemental information ensuring the permanent diverse attitude of each element seems a reasonable addition to us. This discussion on biological analogy is just our interpretation and hypotheses. We leave the scientific explanation for social and biology experts.

3.2. Velocity Initialization & Limiting

The velocity of the particles, just like their position - could be initialized randomly or at specific values. How it is regulated during the simulation is more critical. In the first studies of the PSO [10], it was reported that without velocity limiting particles could jump out from the boundaries of the search space. Therefore, to control the dynamic range of each swarm member, their velocities limited by the Equation 9.

$$If\ \ |v_i| > v_{\{max\}},\ v_i = \frac{v_i v_{\{max\}}}{|v_i|}$$

(9)

For our recipe we went by rule of thumb and v_{max} for the first stage was to jump at most 15% of the search space and for the second stage 25%.

3.3. Adjusting Inertia

Another important parameters of the PSO algorithm is inertia. The importance of inertia to the convergence rate are reported by many researchers. Limiting the inertia according to certain rule might result in faster convergence.

However, reduced velocity could trap the particles in local maximum as their velocity would be too low to escape local maxima as search progress. In this section several inertia update rules are presented:

Constant value could be used to control the inertia by [26] Setting inertia a constant value is simple solution.

Linear - control of the inertia could result in reduced convergence time [27]. Usually upper and lower bounds of inertia w_{min} and w_{max} are set to [0.4, 0.9]

$$w^k = w_{max} - \frac{k(w_{max} - w_{min})}{k_{max}} \qquad (10)$$

Similar to the linear, exponential control of the inertia was experimented in [11] to accelerate convergence.

$$w^k = w_{min} + (w_{max} - w_{min}) \exp\left(-3\left(\frac{k}{k_{max}}\right)^2\right) \qquad (11)$$

Inertia update rule [10] which regulates particle movement according to explored distance are shown in Equation 12 & Equation 13. Significant gain in speed could be obtained by online regulation of inertia.

$$w'^k = w_{max} - (w_{max} - w_{min}) \frac{d_{max} + d_{min} - d_{mean}^{k-1}}{d_{max}} \qquad (12)$$

$$w^k = \frac{w_{min}}{2} + r\, w'^k \qquad (13)$$

Here d is voltages which were tested to locate GM and r is random number in the range of [0, 1]. Benefit of exploiting statistics of visited points is getting rid of need to determine maximum number of iterations which is required for other methods.

3.4. Acceleration constants (social, cognitive)

As mentioned before, tuning acceleration constants (parameter configuration) is done to improve the performance of the PSO to accomplish a specific task. However, due to a wide range of outputs from the PV system - calibrating the acceleration constant seems quite a difficult task if even possible. Therefore, we rely more on the improvement of the algorithm architecture and preserve its generality (the ability to deal with a variety of functions). Upon recommendations in literature we selected c_1 and c_2 as (1.5 & 2.05)

3.5. Topology Selection

Information exchange policies between particles of a swarm exhibits a substantial consequence on the mechanics of their movement, so, to the algorithm performance. Topology modifications result in vastly varying results. It was observed in very first experiments on cellular automata by von Neumann that information exchange between cells can lead to the vast diversity of results.

In the case of PSO particles, this is no exception - topology receives important purpose in algorithm functionality. For instance, two completely independent swarms could run on parallel due to complete isolation (absence of communication) Figure 5b. A myriad of arrangements for information exchange exists. Some of them force particles to explore by delaying propagation of information [18] Figure 5c. However, our goal is to locate a global maximum within the least amount of trials possible. For this reason, large swarms don't suit our purpose. In our adaptation - the number of particles was limited to as small as three, and we followed the original information exchange structure Figure 5a GBEST topology.

Algorithm 2 Modified PSO

1: Initialize Particle population *(In uniform sections N = 3)* **{ x_i}, $i \in$ [1, N]**

2: Initialize velocity of each particle **{ v_i}, $i \in$ [1, N]**

3: Initialize the best-so-far position of each individual:, $i \in$ **[1,N]**

4: Define Topology (**GBEST** topology)

5: Define social & cognitive acceleration constraints.

6: Define limits of velocity and position *(For stages 1 & 2)* .

7: While (k < maximum iteration set)

    1st Phase:

        Iterate through equations 6 - 3

            Set: w = 0.72

    2nd Phase:

        Iterate through equations 2 - 3

            Adaptive inertia update
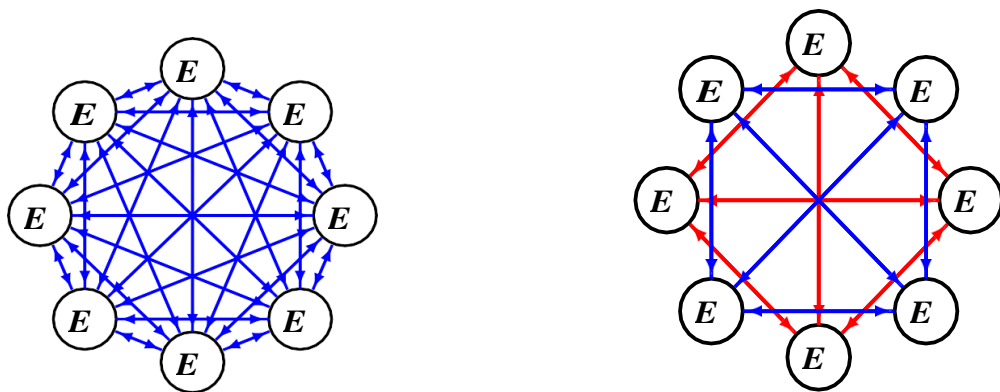
(a) Information exchange in **GBEST** topology   (b) Swarms running on parallel in **GBEST** topology
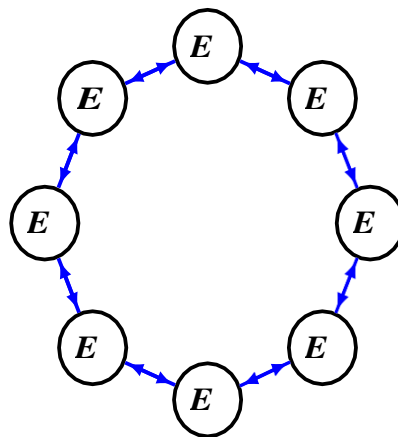
(c) RING (**IBEST**) topology

Fig.5. Visualization of various PSO topology *(here E is particle)*

4.Simulations

To our surprise - steps (# of attempts by particles in total, till global best was found) required by the modified version of the algorithm didn't differ considerably from the offered test scenarios. Examining the literature, we found that similar results were observed by. Meshing the landscape offers reliability close to the exact solution even for a population size of 3 as we received no failures for 1000 runs for all test scenarios.

The additional term cannot guarantee a uniform exploration of the search space. But adds a probability to explore the region where particles are initialized and contribute to the final answer.

In Table 1 results of numerical experiments were we compare original PSO with linear and adaptive inertia update to the modified version are presented. Uniform initialization was implemented only for the later to point on its importance. Our interpretation of convergence is when all particles are within 1 % range of GM. Failure is counted if voltage at GM differs from the real GM more than 1 %. However, all the failures we detected ended up converging in wrong peak. Every optimization problem needs adjustments of some kind [28], and there is no optimization algorithm which could effectively tackle every single problem [29]. However, for MPPT controller we did our best to preserve the generality of the algorithm to deal with wide variety of PV system outputs.

Table 1 - Simulation results (here **PSO1** - is original PSO with linear inertia, **PSO2** - is PSO with adaptive inertia, PSO3 - is modified PSO)

Test cases	PSO1	PSO2	PSO3	PSO1	PSO2	PSO3
	Fails for 1000 runs			Expected convergence		
1	0	0	0	194.39	42.56	73.01
2	0	6	0	196.70	41.91	72.92
3	155	183	0	213.39	42.28	95.63
4	106	108	0	216.70	42.42	86.53
5	0	30	0	198.08	38.14	72.18
6	0	36	0	195.02	39.18	72.08
7	0	32	0	197.80	40.35	72.92
8	1	35	0	206.54	42.60	75.37
9	14	340	0	193.67	41.16	83.17
10	0	0	0	204.15	40.11	72.64
11	1	14	0	192.07	39.06	72.81
12	1	5	0	213.60	42.22	75.77
13	0	4	0	197.91	40.76	72.78
14	5	40	0	210.90	39.23	73.22
15	103	239	0	193.24	39.86	74.74
16	9	67	0	194.51	42.63	73.44

Conclusion

Although adding extra parameter and segmenting the algorithm increases the undesirable complexity of the initial structure. The measure is inevitable as the exploration and exploitation stages require different mechanics of the particle movement across search space. Including a constant term that vanishes after some period when enough information was collected showed good reliability even with a population size of 3 particles. It could be considered as meshing the objective function prior to calculation. Our adaptation of PSO demonstrated superior characteristics in terms of reliability and showed good speed. As further research, it will be interesting to see if PSO could be hybridized with machine learning algorithms to enhance its intelligence more significantly. Most biological creatures, if not all - exhibit learning features and utilize the information they collected over time to improve the quality of the process they are performing. While PSO uses comprehensibly only from several points, and the vast majority of the data it collected remains unused. Our attempts to manually guiding the exploration quality of PSO show the necessity of another component for the algorithm to learn from the past search and avoid stagnation and inefficient exploration.

Acknowledgements

REFERENCES

1 Petrone G., Spagnuolo G., Teodorescu R. Reliability issues in photovoltaic power processing systems. *IEEE Transactions on Industrial Electronics*, 2008, Vol. 55, no. 7, pp. 2569–2580.

2 Bruendlinger R., Bletterie B., Milde M. Maximum power point tracking performance under partially shaded pv array conditions. *Proc. of the 21st Eur. Photovolt. Sol. Energy Conf., Berlin, Germany*, 2006, pp. 2157–2160.

3   Abdelsalam A. K., Massoud A. M., Ahmed S. High-performance adaptive perturb and observe mppt technique for photovoltaic-based microgrids. *IEEE Transactions on Power Electronics*, 2011, Vol. 26, No. 4, pp. 1010-1021.

4   Tey K. S. and Mekhilef S. Modified incremental conductance algorithm for photovoltaic system under partial shading conditions and load variation. *IEEE Transactions on Industrial Electronics*, 2014, Vol. 61, No. 10, pp. 5384–5392.

5   Koutroulis E., Blaabjerg F. A new technique for tracking the global maximum power point of pv arrays operating under partial-shading conditions. *IEEE Journal of Photovoltaics*, 2012, Vol. 2, No. 2, pp. 184-190.

6   Seyedmahmoudian M., Mekhilef S., Rahmani R. Maximum power point tracking of partial shaded photovoltaic array using an evolutionary algorithm: A particle swarm optimization technique. *Journal of Renewable and Sustainable Energy*, 2014, Vol. 6, No. 2, pp. 023102. https://doi.org/10.1063/1.4868025

7   Mao M., Duan Q., Zhang L., Chen H. Maximum power point tracking for cascaded pv-converter modules using two-stage particle swarm optimization. *Nature Scientific Reports*, 2017, pp. 9381. DOI: 10.1038/s41598-017-08009-7

8   Verma D., Nema S., Shandilya A. M. Comprehensive analysis of maximum power point tracking techniques in solar photovoltaic systems under uniform insolation and partial shaded condition. *Journal of Renewable and Sustainable Energy*, 2015, vol. 7, No. 042701. 042701. DOI: <u>10.1063/1.4926844</u>

9   Salamon P., Sibani P., Frost R. *Facts, Conjectures, and Improvements for Simulated Annealing*. Society for Industrial and Applied Mathematics, 2002, 127 p.

10 Kennedy J., Eberhart R.C. Particle swarm optimization. *IEEE international conference on neural networks. IEEE, Piscataway, NJ, USA,* 1995, Vol. IV, pp. 1942–1948.

11 Fernandez-Martinez J., Garcia-Gonzalo E.   Particle swarm optimisation: time for uniformisation. *Int. J. Computing Science and Mathematics,* 2013, Vol. 4(1), pp. 16 – 33. DOI: 10.1504/IJCSM.2013.054671

12 Trelea I. C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 2003, no. 85, pp. 317–325.

13 Zheng Y., Ma L., Zhang L. On the convergence analysis and parameter selection in particle swarm optimization. *Proceedings of the 2nd Intern. Conference on Machine Learning and Cybernetics,* Wan, 2003, pp. 2 – 5.

14 Kurmanbay A., Baktybekov K., Sakhanov K., Syzdykov A., Mukhamediyev A. Optimization of series-parallel connection of pv array to mitigate negative influence of partial shading conditions. *IOP Conf. Series: Materials Science and Engineering* 2020, 868 012001. DOI: <u>10.1088/1757-899X/868/1/012001</u>

15 King D., Boyson W., et al. *Photovoltaic array performance model.* Sandia National Laboratories, 2004, 43 p.

16 Kirkpatrick S., et al. Optimization by simulated annealing. *Science*, 1983, Vol. 220, No. 4598. pp. 671-680.

17 Fernandez-Martínez J., Garcia-Gonzalo E. The pso family: deduction, stochastic analysis and comparison. *Swarm Intell 3,* 2009, No. 245, pp. 273. https://doi.org/10.1007/s11721-009-0034-8

18 Chen S., Montgomery J. Selection strategies for initial positions and initial velocities in multi- optima particle swarms. *Genetic and Evolutionary Computation Conference.* Dublin, Ireland, 2011, pp. 53 – 60.

19 Higashitani M., Ishigame A., Yasuda K. Particle swarm optimization considering the concept of predator-prey behavior. *IEEE Congress on Evolutionary Computation*, 2006. DOI: <u>10.1109/CEC.2006.1688341</u>

20 Mikki S., Kishk A. Improved particle swarm optimization technique using hard boundary conditions. *Microw Opt Technol Lett,* 2005, Vol. 46, No. 5, pp. 422–426.

21 Riget J., Vesterstrøm J.S. *A diversity-guided particle swarm optimizer – the ARPSO. T*ech. rep., EVA Life Project Group, 2002.

22 Chowdhury S., Tong W., Messac A., Zhang J. A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation. *Struct Multidisc Optim*, 2012, vol. 47, pp. 367–388.

23 . Richards M., Ventura D. Choosing a starting configuration for particle swarm optimization. *Proceeding of the 2004 IEEE Intern. Joint Conference on neural networks,* Budapest, Hungary, 2004, pp. 2309–2312.

24 Khatib W., et al. *The studga: A mini revolution?* in Parallel Problem Solving from Nature. 2006, pp. 683-691.

25 Eiben A., BackT., Schoenauer M., Schwefel H. Parallel Problem Solving from Nature - PPSN V. *Proceedings of the 5th International Conference., Amsterdam, The Netherlands, September 27-30*, Springer, 1998. Available at: <u>www.springer.com/gp/book/9783540650782</u>

26 Simon D. *Evolutionary Optimization Algorithms Biologically-Inspired and Population-Based Ap- proaches to Computer Intelligence*. John Wiley & Sons, Inc., 2013, 685 p.

27 Clerc M.,   Poli R. *Stagnation analysis in particle swarm optimization or what happens when nothing happens. T*ech. rep., Department of Computer Science University of Essex, 2006, 15 p.

28 Shi Y.H. and Eberhart R.C. Parameter Selection in Particle Swarm Optimization. *Proceedings of the 7th International Conference on Evolutionary Programming VII*. 1998, pp. 591 – 600.

29 Konak A., Coitb D., Smithc A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 2006, Vol. 91, pp. 992–1007.

30 Wolpert D., Macready W. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,* 1997, Vol. 1, No. 1, pp. 67–82.

.