A Rapid Feasibility Checking for Reconfiguration of Mismatched PV Arrays

Dafang Zhao, Fukuhito Ooshita, and Michiko Inoue
Nara Institute of Science and Technology, Japan
Email: {zhao.dafang.yu7, f-ooshita, kounoe}@is.naist.jp

***Abstract*—Power generation efficiency of Photovoltaic (PV) arrays is significantly affected by partial shading and PV cell damage. Partial shading or PV cell damage induces mismatched power generation among PV panels and causes the efficiency loss of power generation. Power generation of mismatched PV arrays can be recovered by re-configuring connection of PV panels. Recently, an efficient and effective reconfiguration method is proposed where it first selects candidates of configurations with approximation and then find the best one with precise power simulation. However, we found that some of configuration candidates are not able to be realized and the method does not show any systematic way to identify such a feasibility. In this paper, we propose an algorithm to rapidly check feasibility that a given configuration candidate can be actually formed by given PV panels. The experimental results demonstrate that the proposed method rapidly checks the feasibility with a small error rate (this part will be written with actual values).**

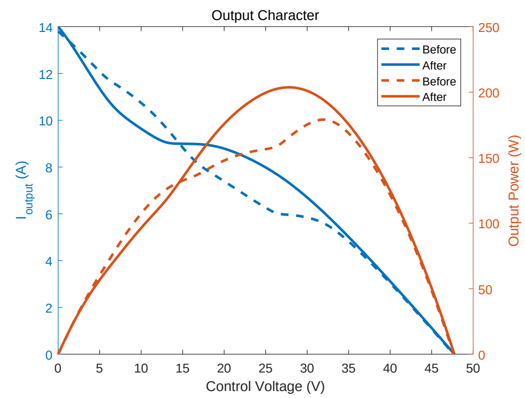


Fig. 1: Power generation before and after reconfiguration

I. Introduction

As fossil fuel depletion and environmental pollution become more serious, green and renewable energy has become necessary for a sustainable society and environment. Photovoltaic (PV) systems receive significant attention since the sun has unlimited energy and PV arrays can be easily scaled up. However, due to the nature of a PV cell, which is a basic component of a PV array, PV arrays are sensitive to partial shading and PV cell damage. PV cells could not uniformly generate power when PV cells experience different irradiances or some of cells have physical defects, and such a mismatched condition might accelerate heating and aging of PV cells and cause further damaging. To prevent PV cells from damaging, bypass diodes are usually placed in PV arrays and they are turned on in mismatched conditions. However, it causes a significant loss of power generation.

PV arrays are hierarchically constructed such that a group of PV cells form a PV module, a group of PV modules form a PV panel, and a group of PV panels form a PV array. Power generation of mismatched PV arrays can be recovered by reconfiguring connection of these components. Fig.1 shows an example of power generations of a PV array with a partial shading before and after reconfiguration. The current is improved in a middle range of control voltage (20V - 30V), and then the maximum power is improved.

Though several reconfiguration methods have been proposed, most works consider reconfiguraion in PV cell level [2], [3], [5], [6]. However, cell level reconfiguration requires a significantly high computation time and a large number of switches for reconfiguration. In addition, they require a special PV panels with capability of switching, and cannot be applied to the system constructed with standard PV panels. From a practical view, reconfiguration of connection among PV panels is a realistic solution since a PV panel is manufactured as a physical one panel with two terminals and PV paneles can be flexibly interconnected.

Recently, an efficient and effective reconfiguration method [9] is proposed where it first selects candidates of configurations with approximation and then finds the best one with precise power simulation. However, the candidates are specified in a PV module level though PV modules could not be fully reconfigured. Actually, we found that some of configuration candidates are not able to be realized. However, the paper [9] does not show any systematic way to identify such a feasibility.

In this paper, we propose an algorithm to rapidly check feasibility that a given configuration candidate is actually formed by given PV panels. The proposed method can efficiently check the feasibility while identifying most feasible cases accurately. The experimental results demonstrate the effectiveness of the proposed method where ...

The remaining of this paper is organized as follows. Section II introduces a PV array which is a targeting PV system in this paper. Section III gives related works. Section IV defines a feasibility, and Sections V and VI give the details of algorithm and evaluation. Section VII shows the experimental results. Conclusions are provided in the last section.

II. Photovoltaic array

A PV system or PV array is composed of PV panels that have two terminals of plus and minus and can be interconnected. There are two common connection styles for PV arrays, *series-parallel* array and *total-cross-tied* array. In a series-parallel array, PV panels are connected in series, and multiple series connections are connected in parallel. In a total-cross-tied array, parallel connections of PV panels are connected in series. In this paper, we focus on series-parallel arrays, however the basic idea of the proposed method can be applied also to total-cross-tied arrays. Hereafter, we simply call a series-parallel array a *PV array*.

Figure 2 shows an example of a PV array. A PV array is a parallel connection of (PV) strings where a string is a series connection of PV panels. A PV panel is a series connection of PV modules, and a PV module is a parallel connection of a series of PV cells and a bypass diode. A typical PV panel is composed of three PV modules each of which has 12-24 PV cells. A PV module can generate power for a given voltage according to its I-V characteristics as shown in Fig.3(a). The I-V characteristics is affected by irradiance level and physical damage of PV cells. Figure 3(a) also shows a typical degradation of a I-V characteristics where generated current is reduced with some ratio while keeping the voltage range. When a PV panel has a partial shade, that is its PV modules have different irradiance levels and hence different I-V characteristics, the PV panels might have multiple peaks (maximum power points, MPPs) in power generation as shown in Fig.3(b).

To find out accurate I-V characteristics of a PV panel, we need a time consuming power simulation. However, we can roughly understand I-V characteristics of a PV panel (and also a string) as follows. When a control voltage is low, PV modules with high irradiance level are active while PV modules with low irradiance level are inactive with turning on their bypass diodes. In this case, a high current can flow in the PV panel. When control voltage is high, some of bypass diodes become turning off and the corresponding PV modules become active. In this case, the current level is reduced and generated power sometimes increases and sometimes decreases. The generated power at MPPs can be roughly estimated. In Fig.3(b), one PV panel has three PV modules with different irradiance levels. The peak currents for these modules are 5A, 3A, and 1.5A, respectively. Voltages at MPPs are roughly 20V, 40V, 60V in Fig.3(b), those are roughly multiples of a voltage of MPP for one PV module (around 20V in this case). At the first MPP (MPP1), only one PV module with a peak current of 5A is active at a control voltage of 20V, so the generated power is 100W. At the second MPP (MPP2), two PV modules with peak currents of 5A and 3A are active at a control voltage of 40V. In this case, the current level of the PV panel is 3A since two PV panels are connected in series and they have to have the same current level, and the generated power is 120W. At the third MPP (MPP3), the current level of the PV panel is 1.5A at a control voltage of 60V, and the generated power is 90W.

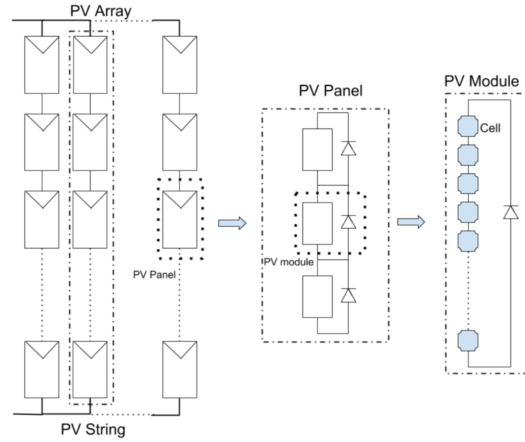
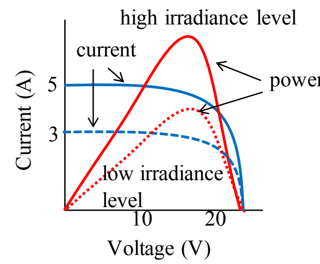
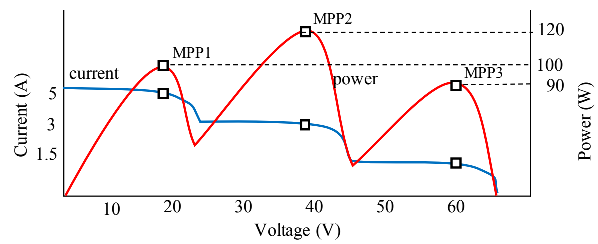


Fig. 2: PV array, string, module and panel



(a) PV module



(b) PV panel

Fig. 3: I-V characteristics

In a PV arary, we have two constraints. PV panels in the same string have the same current level, while all the strings have the same control voltage. When considering reconfiguration of PV panel connection, we should find out the best configuration while considering these constraints.

III. Related works

Some of methods that reconfigure connection of PV panels are proposed. A PV array reconfiguration using genetic algorithm (GA) is proposed in [1]. Though it can give a new configuration, but computing cost is significantly high and the algorithm cannot generate the best configuration precisely. An algorithm based on particle swarm optimization to find switch matrix topology is proposed [8].

TABLE I: Extracted currents of MPPs

	Panel 1	Panel 2	Panel 3	Panel 4
MPP1	3.10A	3.09A	0.52A	2.47A
MPP2	2.98A	2.55A	0.50A	1.53A
MPP3	1.55A	2.48A	0.46A	0.48A

TABLE II: Approximated currents of MPPs

	Panel 1	Panel 2	Panel 3	Panel 4
MPP1	3A	3A	0.5A	2.5A
MPP2	3A	2.5A	0.5A	1.5A
MPP3	1.5A	2.5A	0.5A	0.5A

Orozco-Guierrez et al. proposed an efficient reconfiguration algorithm for mismatched PV arrays orozco2016optimized. This is the most related work to this paper. The method utilizes information of MPPs of every PV panel that are extracted using an online monitoring [11] and a power estimation [10]. Then currents and voltages of MPPs are approximated and grouped into a small number of classes so that the number of possible combinations, or a search space, is reduced. For example, Table I shows extracted currents for 4 panels each of which has 3 modules, and they are approximated into four current levels as shown in Table II. As mentioned in Section II, if some panel has m MPPs with current level of I or larger, m modules can be active with current level of I. Table III shows the maximum number of active modules for each current level.

Then power values are approximated for possible configurations. Table IV shows an example to form a PV array with two strings where control voltages are approximated as multiples of 20V (the number of active modules per string times 20V). The method has a simple feasibility check as follows.

(Feasibility 1) For the n-th highest current level I_n, the number of active modules in a system is $N(I_n)/n$ or less.

Where, $N(I)$ denotes the total number of active modules for a current level I. For example, consider a candidate current pair (3A. 2.5A). In this case, the highest and the second highest current levels are 3A and 2.5A, respectively, and the total number of active modules for 3A and 2.5A are 3 and 6, respectively (see Table III). Therefore, the number of active modules in the first string is 3 or less, and the number of active modules in the second string is also $6/2 = 3$ or less. Consequently, the number of active modules per string is at most 3 for a candidate current pair (3A. 2.5A). Table IV has values in the cells when Feasibility 1 is satisfied.

The method finds the largest power value from possible candidate configurations (330W for current pair (3A, 2.5A) with three active modules in this case) and evaluates the case more precisely along with the cases with close values to the best value since the analysis are given with approximated values. In the case of Table IV, 330W, 320W and 300W are selected for further evaluation.

IV. Feasibility

In the method [9], to execute a power simulation for a PV array, we need to assign panels into strings to realize a candidate configuration. For example, a candidate configuration for

TABLE III: Maximum number of active modules

	Panel 1	Panel 2	Panel 3	Panel 4	total
3A	2	1	0	0	3
2.5A	2	3	0	1	6
1.5A	3	3	0	2	8
0.5A	3	3	3	3	12

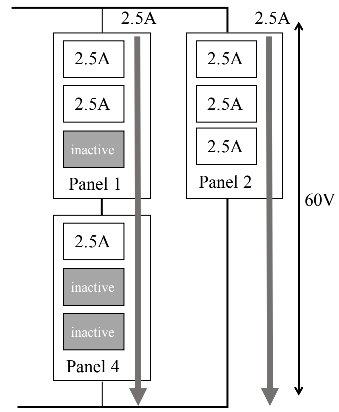


Fig. 4: Feasible configuration

a current pair (2.5A, 2.5A) with 3 active modules is realized by assign Panel 1 and Panel 4 to the first string and Panel 2 to the second string as shown in Fig.4. However, we could not find feasible assignment for a current pair (3A, 2.5A) with 3 active modules or a current pair (2.5A, 1.5A) with 4 active modules though they are expected to generate higher powers 330W or 320W. That is, the condition Feasibility 1 is a necessary but not sufficient to check the actual feasibility.

Now we will define the feasibility. For a PV array A with s strings, let $M_{A,i}(I)$ denote the number of modules that can be active with a current level I in the i-th string. Let $Q = (Q_1, Q_2, \ldots, Q_s)$ be a sequence of currents required for strings. A sequence of currents Q is feasible with m modules if and only if it is possible to form a PV array A such that

$$M_{A,i}(Q_i) \geq m \tag{1}$$

holds for each string $i (1 \leq i \leq s)$.

V. An algorithm to identify feasibility

In this section we proposed an algorithm to identify feasibility.

A. Input And Output Of Algorithm

The input of algorithm is the information of PV array and Configuration candidates. The information of PV array contains a set of panels and the number of working modules in each panel for each current values. Table VI shows the information of a PV panel from Table V for configuration $\{$0.5A, 0.5A, 3A, $Q_M = 6$ $\}$. It terms that for panel P2, 3 modules are able to work on String1 or String2 and 2 modules are able to work on String3. Due to one panel can only connected to single string. So the output of algorithm is the feasibility answer for a configuration. That is, the algorithm

TABLE IV: Approximated power

# modules per string	currents of strings (A)									
	(3,3)	(3,2.5)	(3,1.5)	(3,0.5)	(2.5,2.5)	(2.5,1.5)	(2.5,0.5)	(1.5,1.5)	(1.5,0.5)	(0.5,0.5)
1	120W	110W	90W	70W	100W	80W	60W	60W	40W	20W
2	-	220W	180W	140W	200W	160W	120W	120W	80W	40W
3	-	**330W**	270W	210W	**300W**	240W	180W	180W	120W	60W
4	-	-	-	-	-	**320W**	240W	240W	160W	80W
5	-	-	-	-	-	-	**300W**	-	200W	100W
6	-	-	-	-	-	-	-	-	240W	120W

TABLE V: Infeasible Example

	P1	P2	P3	P4	P5	P6	P7	P8	P9
0.5A	***	***	*	*	***	**	*	***	*
1.5A	***	***	*	*	***		*		
3A	**	**	*	*	**				

TABLE VI: Information of PV Array

	P1	P2	P3	P4	P5	P6	P7	P8	P9
String1-0.5A	3	3	1	1	3	2	1	3	1
String2-0.5A	3	3	1	1	3	2	1	3	1
String3-3A	2	2	1	1	2	0	0	0	0

determines whether there exists an assignment of panels to strings that realizes the configuration required.

B. Algorithm

Let the S for the number of PV strings and P be the number of PV panels, $M_{(i,p)}$ is the number of modules on panel p at current value i. For each panel we consider the different vector $Loss[p,k] = M_{(i_{k-1},p)} - M_{(i_k,p)}, (k \leq S)$. This vector means if we use panel p on current k it will loss $Loss[p,k]$ working modules in current $k-1$. Without loss of generality, for each panel p, $Loss^*[p] = (Loss[p,k_1], Loss[p,k_2], ..., Loss[p,k_S])$ and $Loss^*[p_i] \leq Loss^*[p_j]$ holds for any $i \leq j$. After sorting all panels in lexicographic order by vector $Loss^*[p]$, we will have a order for arranging panels in different location.

To assign panels in right location, following steps are required.

(1) Selecting panel from sorted panels until selected contain enough PV modules working at the highest desired current value.
(2) Remove selected panels which don't have PV modules working at the highest desired current value for further more selection.
(3) In selected panels, try to find a combination that provided exact number of modules working at the highest desired current value.
(4) Swap selected panels to find a combination with minimum number of panels. This SWAP will follow several priorities.
(5) If SWAP approved, step 4 will be repeated till number of panels are minimized.
(6) Otherwise, none of SWAP will be approved. Evaluate selected panels with feasibility requirement in IV equation (**??**).
(7) If there remain any un-selected panel, as them to different PV strings that balance the string length.

We explain step (3) and (4) that finding a local optimized combination of a desired current value. To do this, we begin with sorting panels by vector $Loss^*[p]$. That will let a panel with minimum module losses be the first one in a string with highest current value. When $Loss^*[p_a] = Loss^*[p_b]$ which means panel a and panel b's module losses are same and they are in same loss level (LV). By using vector LV we can divide selected panels into different groups.

To get exact number of working modules, we start to search panels form last loss level to first loss level. In each LV remove the panel x which $M_{(j,x)} \leq Q_M - \sum(M_{(j,\text{Selected Panels})})$. This step will cut the number of panels and increase flexibility.

TABLE VII: SWAP Example

	Selected Panels					Swap Panels	
	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$
High	1	1	1	1	2	2	2
Low	1	1	1	2	3	3	3
$Loss^*[p]$	0	0	0	1	1	1	1

For swapping panels, panels with same LV as selected panels can be swapped. In order to not loss flexibility, swapped panles' $Loss^*[p]$ should less than unswap panels'. For example as Table VII, panel $P1 - P5$ are selected panels. Panel $P6$ and $P7$ have the same $Loss^*[p]$ vector as $P5$, so they are able to swap. Due to $P1$ and $P2$'s $Loss^*[p]$ are less than $P6$'s, we can not swap them. On the other hand, we can swap panel $P1$ and $P4$ with $P6$. Keeping the same module losses and further reduce the number of panels. Pseudo codes are given below to give details to summarize the process of proposed algorithm.

C. Evaluation

Although the pilot example refers to three strings only, the extension of the method to any number of parallel connected strings is the same.

To evaluate the performance of proposed algorithm, we compare the result with exhaustive searching algorithm for 300 random shadow distributed PV array. For each PV array, contains **how many - how many** PV panels connected into **how many - how many** PV strings. For each PV panels, it will have how many - how many working current value.

Algorithm 1: Sort panel by lexicographic order

Input: $M_{(i,p)}$: The number of modules in panel p at current i; P: Number of panels; S: Number of strings; Q_M: The minimum number of modules for each string;
Output: Feasibility Result;
Conf: Configuration Result;

for k = *1 to S-1,* **each** *panel p* **do**
  $Loss[p,k] = M_{(i_{S-k},p)} - M_{(i_{S-k+1},p)}$
end
PSet = $\{p_1, p_2, ..., p_P\}$
for j = *S to 1* **do**
  n = 1;
  for each $p \in$ *PSet* **do**
    $Loss^*[p] = (Loss[p, S-j+1], Loss[p, S-j+2], ..., Loss[p, S])$
    Let $\alpha_1, \alpha_2, ..., \alpha_{|PSet|}$ be indices such that $Loss^*[p_{\alpha i}] \leq Loss^*[p_{\alpha j}]$ holds for **any** i $\leq$ j
    /* Sort Panels in lexicographical order by *Loss* */
  end
  while SUM = $\sum_{k=1}^{n} M_{(i_j,\alpha_k)} < Q_M$ **do**
    if n == $PSet$ **then**
      return : Feasibility = NO
    end
    n++;
  end
  $TempConf$ = { p_{α_1}, p_{α_2}, p_{α_3},..., p_{α_n} };
  forall the *panels in* $TempConf$ **do**
    remove panels which $M_{(j,\alpha)} == 0$
  end
  Over = SUM - Q_M;
  Do Algorithm 2
  $PSet$ = $PSet$ - Conf[j];
end
return : Feasibility = YES

The computational time required for finding optimal configuration and accuracy of feasibility judgment in the proposed algorithm and exhaustive search algorithm are compared in Table **??**. As shown in Table **??**, proposed algorithm achieved less computing time compare with exhaustive search algorithm and with high accuracy.

VI. Simulation

In this section, we consider using 3 different size PV array, which are 3×4 array, 60×4 array and 40×15 array. The whole process though SPICE simulation with the configuration result by proposed algorithm. The PV cell's parameters are shown in Table VIII.

A. 3×4 Array

In a 3×4 array, there are 12 panels connected in parallel for 4 strings, for each string contains 3 PV panels. The shadow

Algorithm 2: find exact # of modules

Input: $TempConf$, n, $Loss^*[p]$,Over
Output: Conf[j]

i = h = 1; $LV[h]$ = $\{p_{\alpha_1}\}$;
while $i \neq |PSet|$ **do**
  i++;
  if $Loss^*[p_{\alpha_i}]$ == $Loss^*[p_{\alpha_{i-1}}]$ **then**
    $LV[h] = LV[h] \cup \{p_{\alpha_i}\}$
    continue;
  else if $i > n$ **then**
    break;
  else
    h++;
    $LV[h]$ = $\{p_{\alpha_i}\}$
  end
end
while *h≠0* **do**
  if *Over* == *0* **then**
    break;
  else
    L=$\left|\{p_i \in LV^*[h'] \mid h' \leq h-1\}\right|$
    $l^* = \left|\{p_i \in LV^*[h] \mid M_{(j,\alpha_i)} \leq Over\}\right|$
    Let l be the minimum index l' such that $\sum_{t=l'}^{l^*} M_{(j,\alpha_{L+t})} \leq Over$
    $Over = Over - \sum_{t=l'}^{l^*} M_{(j,\alpha_{L+t})}$;
    $TempConf$ = $TempConf$ - $\{p_{L+l}, p_{L+l+1}, ..., p_{L+l^*}\}$;
    $h--$;
  end
end
$U = \{p_{\alpha_1}, p_{\alpha_2}, ..., p_{\alpha_{i-1}}\}$;
$N = U - TempConf$;
repeat
  Let $\beta_1, \beta_2, \beta_3, ..., \beta_{|TempConf|}$ be new indices of $TempConf$;
  Let $\gamma_1, \gamma_2, \gamma_3, ..., \gamma_{|N|}$ be new indices of N;
  for x = *1 to* $|N|$ **do**
    forall the p_β *in* $TempConf$ **do**
      if $\sum M_{(j,\beta}$==$M_{(j,\gamma_x)}$ **then**
        $TempConf$ = $TempConf - \{p_\beta\} + \{p_{\gamma_x}\}$;
        $N = N - \{p_{\gamma_x}\}$;
        Continue;
      end
    end
until x==$|N|$;

TABLE VIII: PV Cell Parameters

Parameter	Value
Short-Circuit Current	0.66A
MPP-Current	0.58A
Open-Circuit Voltage	280V
MPP-Voltage	215V
Max Power Output	124.7W

conditions for PV array are not uniformly, as shown in Fig. **??**. Without reconfiguration, PV array output I-V characteristic as shown in Fig. **??**. The global MPP as how many Watt on how many voltage. After proposed algorithm based reconfiguration, this PV array achieved how many Watt on how many voltage for global MPP. The output power has increased by how many %

B. 60 $\times$ 4 Array

For medium size 60 $\times$ 4 array, on standard test condition the power output are how many KW without any shadow. Fig. **??** presents the shadow distribution amount the PV array which power output are how may KW. After reconfiguration, panel arrangement as illustrated in Fig. **??**. With reconfiguration, power generation increased by how many %.

C. 40 $\times$ 15 Array

As a large scale PV array. Although the algorithm proposed in [9] are hard to converge when the configuration candidates are widely distributed. The algorithm we proposed still calculate the feasible configuration result very fast. 60 configuration candidates though the feasibility judgment algorithm in 269.6s with 20 feasible configuration as shown in Table **??**.

VII. Conclusions

Acknowledgment

Reference

[1] P. L. Carotenuto, A. Della Cioppa, A. Marcelli, and G. Spagnuolo, "An evolutionary approach to the dynamical reconfiguration of photovoltaic fields," *Neurocomputing*, vol. 170, pp. 393–405, 2015.

[2] D. Nguyen and B. Lehman, "An adaptive solar photovoltaic array using model-based reconfiguration algorithm," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2644–2654, 2008.

[3] J. P. Storey, P. R. Wilson, and D. Bagnall, "Improved optimization strategy for irradiance equalization in dynamic photovoltaic arrays," *IEEE transactions on power electronics*, vol. 28, no. 6, pp. 2946–2956, 2013.

[4] S. Malathy and R. Ramaprabha, "Reconfiguration strategies to extract maximum power from photovoltaic array under partially shaded conditions," *Renewable and Sustainable Energy Reviews*, 2017.

[5] Y. Wang, X. Lin, Y. Kim, N. Chang, and M. Pedram, "Architecture and control algorithms for combating partial shading in photovoltaic systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 917–930, 2014.

[6] J. Storey, P. R. Wilson, and D. Bagnall, "The optimized-string dynamic photovoltaic array," *IEEE Transactions on Power Electronics*, vol. 29, no. 4, pp. 1768–1776, 2014.

[7] E. Faldella, G. C. Cardinali, and P. U. Calzolari, "Architectural and design issues on optimal management of photovoltaic pumping systems," *IEEE Transactions on Industrial Electronics*, vol. 38, no. 5, pp. 385–392, 1991.

[8] F. Iraji, E. Farjah, and T. Ghanbari, "Optimisation method to find the best switch set topology for reconfiguration of photovoltaic panels," *IET Renewable Power Generation*, vol. 12, no. 3, pp. 374–379, 2017.

[9] M. Orozco-Gutierrez, G. Spagnuolo, J. Ramirez-Scarpetta, G. Petrone, and C. Ramos-Paja, "Optimized configuration of mismatched photovoltaic arrays," *IEEE J. Photovolt*, vol. 6, no. 5, pp. 1210–1220, 2016.

[10] M. Orozco-Gutierrez, J. Ramirez-Scarpetta, G. Spagnuolo, G. Petrone, and C. Ramos-Paja, "Fast estimation of mpps in mismatched pv arrays based on lossless model," in *Clean Electrical Power (ICCEP), 2015 International Conference on*, pp. 435–440, IEEE, 2015.

[11] P. L. Carotenuto, P. Manganiello, G. Petrone, and G. Spagnuolo, "Online recording a pv module fingerprint," *IEEE Journal of Photovoltaics*, vol. 4, no. 2, pp. 659–668, 2014.