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-oosita, 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 an efficiency loss of power generation. Power losses of mismatched PV arrays can be recovered by reconfiguring connection of PV panels. In this paper, we introduce a *feasibility check problem* of PV panel configuration. This problem identifies whether a connection among PV panels can be configured from a given PV module level solution. We also propose an algorithm for the *feasibility check problem*. The experimental results demonstrate that the proposed algorithm can identify feasible configurations more than 32,000X faster than the exhaustive search with around 0.6% errors.**

***Index Terms*—photovoltaic, feasibility, reconfiguration, mismatch**

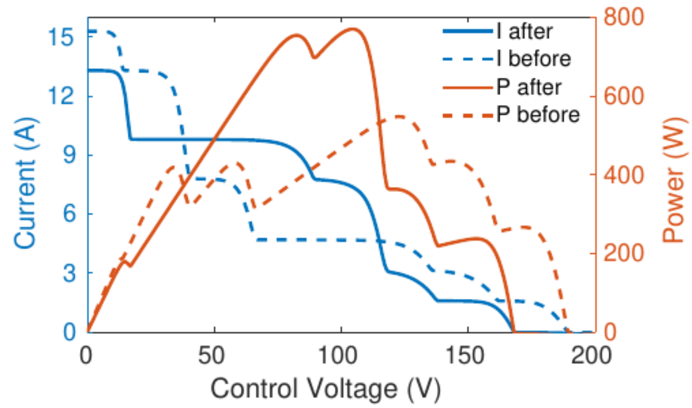


Fig. 1: Power generation before and after reconfiguration

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 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 they experience different irradiances or some of them 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, the operation of bypass diode will cause a number of PV cells to stop deliver power, and it will hence result in 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. We distributed shading cells non-uniformly to a PV array with 3×4 PV panels and applied power simulation. Before reconfiguration, there are multiple peaks in a power - voltage curve with similar power levels, while, after we reconfigured connection among PV panels, some peaks exhibit much higher power levels than others and the maximum output power increased by 35%.

Though several reconfiguration methods have been proposed, most works consider reconfiguration in PV cell or PV module level [1]–[5]. 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 panels can be flexibly interconnected.

Reconfiguration in PV panel level has been investigated [6]–[8]. A PV array reconfiguration using genetic algorithm (GA) was proposed in [6]. Though it can give a new configuration, computing cost is significantly high and the algorithm cannot generate the best configuration precisely. Hu et al. also addressed PV panel reconfiguration where they formulate a nonlinear integer programming problem to optimize power generation by reconfiguration [7]. Orozco-Gutierrez et al. proposed an efficient and effective reconfiguration method [8] where it first selects candidates of configurations using the product of approximated currents and voltages, then finds the best one with precise power simulation. However, these 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 [8] 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 able to form 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 it can identify the feasibility of configurations with a very small false negative rate of less than 1%.

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 commercial 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 - c). The I-V characteristics is affected by irradiance level and physical damage of PV cells. Figure 3(a - c) also show a typical degradation of a I-V characteristics where generated currents are 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(d).

To find out accurate I-V characteristics of a PV panel, we need to apply 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 increasing, some of bypass diodes become turning off and the corresponding PV modules become active. In that case, the current flow in the panel is determined by current of module with lowest irradiance level, and generated power sometimes increases and sometimes decreases. The generated power at MPPs can be roughly estimated. In Fig.3(d), one PV panel has three PV modules with different irradiance levels. Current levels at MPPs for these modules are 4.3A, 3.0A, and 1.5A, respectively. Voltage levels at MPPs are roughly 20V, 44V, and 66V in Fig.3(d), 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 current of 4.3A is active at a control voltage of 22V, so the generated power is 94.6W. At the second MPP (MPP2), two PV modules with currents of 4.3A and 3.0A are active at a control voltage of 44V. In this case, the current level of the PV panel is 3.0A since two PV panels are connected in series and they have to have the same current level, and the generated power is 132W. At the third MPP (MPP3), the current level of the PV panel is 1.5A at a control voltage of 66V, and the generated power is 99W.

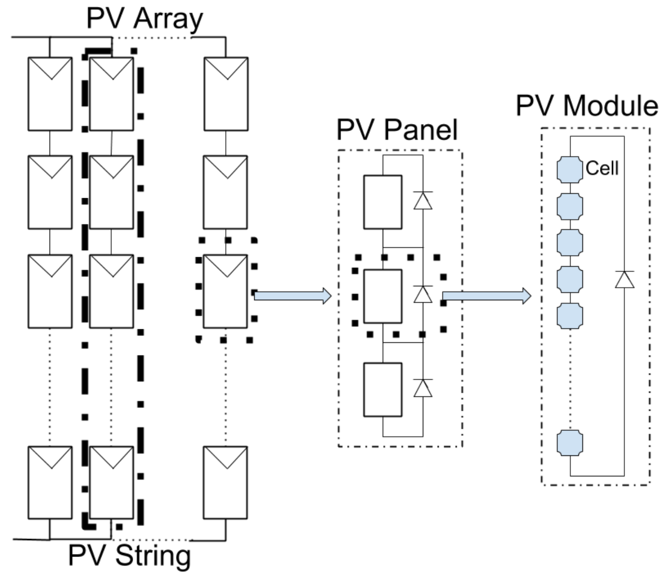


Fig. 2: PV array, string, module and panel

TABLE I

EXTRACTED CURRENTS OF MPPS

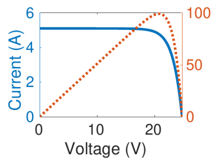
	Panels			
	$P1$	$P2$	$P3$	$P4$
MPP1	3.10A	3.09A	0.52A	2.47A
MPP2	2.98A	2.55A	0.50A	1.53A
MPP3	1.55A	2.48A	0.46A	0.48A

In a PV array, 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.

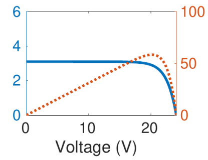
III. Orozco-Guierrez's method

In this section, we briefly introduce a method proposed by Orozco-Guierrez et al. [8] as the most related work to this paper. Orozco-Guierrez et al. proposed an efficient reconfiguration algorithm for mismatched PV arrays. The method utilizes information of MPPs of every PV panel that is extracted using an online monitoring [9] and a power estimation [10].

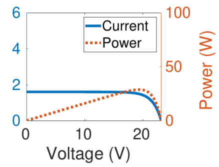
For the method in [8], first, close values of extracted currents or voltages are approximated and grouped into a



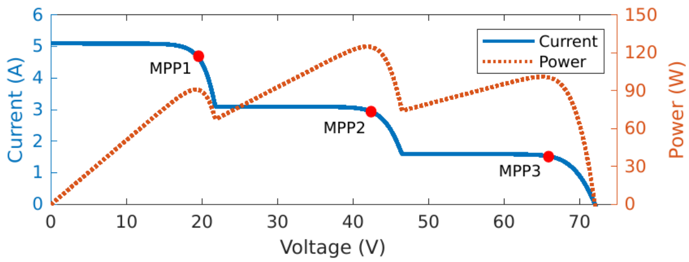
(a) PV module 1



(b) PV module 2



(c) PV module 3



(d) PV panel

Fig. 3: I-V characteristics

TABLE II

APPROXIMATED CURRENTS OF MPPS

	Panels			
	$P1$	$P2$	$P3$	$P4$
MPP1	3A	3A	0.5A	2.5A
MPP2	3A	2.5A	0.5A	1.5A
MPP3	1.5A	2.5A	0.5A	0.5A

small number of classes so that the number of possible combinations, or a search space, is reduced. Then possible combinations of reconfiguration are enumerated. For each candidate configuration its generated power value is estimated by [10] along with careful estimation of possible errors. 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. Control voltage for one PV model is approximated to 20V. Approximated power values for possible configurations are shown in Table IV. 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), and power values of strings are obtained as products of current and voltage values. For example, when current levels of two strings are 3A and 2.5A and the number of active modules per string are 3, the control voltage is 60V, and two strings generate powers of 3A $\times$ 60V $=$ 180W and 2.5A $\times$ 60A $=$ 150W, and the total generated power is approximated about 330W.

When constructing an approximated power matrix like Table IV, the possibility that an array can be formed for a given current sequence and the number of active modules per string (it is called *feasibility*) is examined. The method provides a simple feasibility check as follows.

TABLE III

MAXIMUM NUMBER OF ACTIVE MODULES

	Panels				
	$P1$	$P2$	$P3$	$P4$	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

(Feasibility 1)[1]For the n-th highest current level I_n in a current sequence for strings, the number of active modules in a configuration candidate is $N(I_n)/n$ or less, where $N(I)$ is 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 [8], 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 a current pair (2.5A, 2.5A) with 3 active modules is realized by assign panel P2 to the first string and panels P1 and P4 to the second string as shown in Fig.4. However, we could not find any 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 necessary but not sufficient to give a actual feasibility result.

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 $i (1 \leq i \leq s)$.

[1]In [8], only the case of $n = 2$ is considered, and we extended the condition to general n.

TABLE IV

APPROXIMATED POWER

# modules per string	currents sequence for 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

V. An algorithm to identify feasibility

A. Feasibility Problem

In this section, we propose an algorithm to identify feasibility. First we will formulate the feasibility problem that identifies the feasibility for a given current sequence and the number of active modules per string as follows.

Definition 1 (*Feasibility Problem*):
Input: Information of MPPs (approximated values), a current sequence Q, the number of m of active modules per-string.
Output: Whether Q with m modules is feasible or not.

B. Outline of the algorithm

In the proposed algorithm, we try to configure a PV array and identify the feasibility of configuration. For a given current sequence $Q = (Q_1, Q_2, \ldots, Q_s)(Q_i \geq Q_j$ if $i \leq j)$, we will assign panels to strings in the order of $Q_1, Q_2, \ldots, Q_s$. When selecting a panel to a string, we will consider how much we lose an opportunity that the panel is more effectively used for other current level. For example, if we select Panel $P1$ in Table III for a string with a current level 3A, two modules can be active while the remaining one module becomes inactive. The remaining one module still is not active if $P1$ is used for a current level 2.5A, while it can be active if it is used for current level 1.5A or 0.5A. We will consider that, when selecting $P1$ for a current level 3A, we do not have any loss

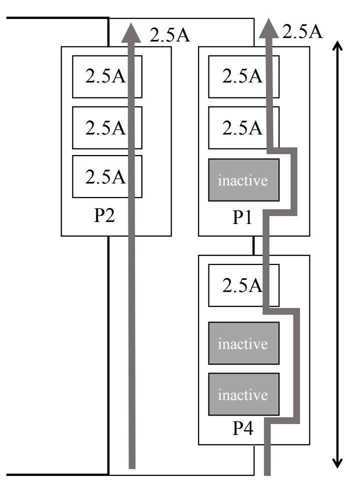


Fig. 4: Feasible configuration for a current pair (2.5A, 2.5A) with 3 active modules.

for a current level 2.5A but lose one module for 1.5A and 0.5A. The algorithm select panels so as to minimize such losses.

Let $M_p(I)$ denote the number of modules that can be active at current level I in a PV panel p. For a given current sequence $Q = (Q_1, Q_2, \ldots, Q_s)(Q_i \geq Q_j$ if $i \leq j)$, a loss of selecting a panel p at the $k-1$-th string for the k-th string is defined as $Loss(p, k) = M_p(Q_k) - M_p(Q_{k-1})$.

To assign panels for the n-th string, the following steps are applied. Let m be the required number of active modules.

1) Sort unselected PV panels in a lexicographically ascending order of $Loss(p, n+1)$, $Loss(p, n+2)$,…, $Loss(p, s)$ and $M_p(Q_n)$.
2) Select PV panels until selecting m or more active modules for a current level Q_n.
3) Cancel redundant PV panels so that the number of active modules at a current level Q_n is minimized.
4) Swap selected PV panels with unselected PV panels so that the number of selected PV panels is minimized.

Before explaining more details, we first show an example. Table V shows how to select PV panels for the first string where the required number m of active modules is 5. Panels are sorted by $Loss$ and $M_p(Q_1)$ (the number of active modules for the first string) in the order of $P1$, $P2$, …, $P7$. In the example, panels $P2$ to $P6$ are at the same loss level $(1, 0)$. PV panels are selected for each loss level. First, $P1$ is selected, then panels at the second loss level are selected in the order of $P2$, $P3$, …. After selecting until $P5$, the number of active modules exceed $m(=5)$. In this case, if some redundant panels exist, these panels are canceled. In the example, $P4$ is canceled. Finally, if some of selected PV panels can be swapped with less number of unselected panels, they are swapped. In the example, $P2$ and $P3$ are swapped with $P6$. Consequently, three panels $P1$, $P5$ and $P6$ are selected.

C. Algorithm

The proposed algorithm first obtains loss levels for each panel, and assigns panels for strings in the order of $1, 2, \ldots, s$ where $Q_i \geq Q_j (i \leq j)$ holds for a required current sequence $Q = (Q_1, Q_2, \ldots, Q_s)$. The pseudo code is given in Algorithm 1. To assign panels to the n-th highest current Q_n, the following 4 steps are applied.

Step 1 sorts unselected panels in a lexicographically ascending order of $Loss(p, n+1)$, $Loss(p, n+2)$,…, $Loss(p, s)$

TABLE V

AN EXAMPLE OF THE PROPOSED METHOD ($m = 5$)

	Panels						
	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$
$M_p(Q_1)$	1	1	1	1	2	2	1
$M_p(Q_2)$	1	2	2	2	3	3	2
$M_p(Q_3)$	1	2	2	2	3	3	3
$Loss(p, 2)$	0	1	1	1	1	1	1
$Loss(p, 3)$	0	0	0	0	0	0	1
step 2	√	√	√	√	√		
step 3	√	√	√		√		
step 4	√				√	√	

and $M_p(Q_n)$. This sorting step gives a basic selection priority for all unselected panels. A panel with less losses for higher current levels are preferred. For example, a panel $P1$ has $(Loss(P1, 2), Loss(P1, 3)) = (0, 0)$, that is there is no loss for further assignment, and it has the highest priority. The last value $M_p(Q_n)$ (the number of active modules at a current level Q_n in panel p) is used to select the minimum number of modules for a required current level, and it will be described in Step 3.

Step 2 selects sorted panels one by one until reaching the required number of active modules. In this selection step, panel with no active modules are skipped.

Step 3 optimizes the number of active modules. In Step 3, if the number of active modules exceeds the required number m, redundant panels are canceled if exist. A panel is *redundant* if the number of active modules is still m or more even if the panel is canceled. Redundant panels are searched for each loss level. Starting with a loss level where the last panel is selected, we check the number of active modules $M_p(Q_n)$ for each panel p where Q_n is a required current level for the current string. Redundant panels are selected and canceled from panels with less number of $M_p(Q_n)$ in the same loss level. If the number of active modules still exceeds m after canceling, the same procedure is applied to the previous loss level. This procedure is repeated until selecting an exactly m active modules or checking all the loss levels. This step accurately finds redundant panels if exist in the following reason. Since the number of active modules in one panel is 0, 1, 2, or 3 as assumed in Section II and panels are selected from panels with less active modules in the same loss level, if the number of active modules exceed m, the surplus is 1 or 2. That means, the last selected panel has 2 or 3 active modules, and panels with 1 or 2 active modules have already selected if exist.

Step 4 optimizes the number of selected panels. In Step 4, we swap selected panels and an unselected panel in the same loss level to reduce the number of selected panels. There are 3 swap rules as shown in Table VI. This swap procedure keeps the number of active modules while reducing the number of panels. The step increases the number of unselected panels and hence increase the flexibility for succeeding selection.

TABLE VI

SWAP RULE

The numbers of active modules	
selected panels	unselected panels
1, 1, 1	3
1, 1	2
1, 2	3

Algorithm 1: Feasibility Check Problem

Input:
$Q = Q_1, Q_2, \ldots, Q_s$: required current for each string;
m: the minimum number of active modules per string;
M_p: the number of active modules in each panel p;
s: the number of strings;
Output: Feasibility Result;

for each string $k (2 \leq k \leq s)$ and panel p **do**
  $Loss(p, k) = M_p(Q_k) - M_p(Q_{k-1})$
end
for *j = 1 to s* **do**
  if $\sum M_p(Q_j)$ for unselected panels $< m$ **then**
    return : Feasibility = NO
  end
  PS = a set of unselected panels with $M_p(Q_j) > 0$;
  // step 1
  sort PS in a lexicographically ascending order of $Loss(p, j + 1)$, $Loss(p, j + 2)$, …, $Loss(p, s)$, $M_p(Q_j)$;
  // step 2
  select panels in the order in PS until $\sum M_p(Q_j)$ for selected panels $\geq m$;
  //step 3
  group selected panels LV_1, LV_2,…, LV_h with loss level (LV_1 is the lowest level);
  for *i = h to 1* **do**
    find and cancel redundant panels in LV_i in the order in PS
  end
  //step 4
  for *i = h to 1* **do**
    apply swap rules for selected panels in LV_i and an unselected panel in the same loss level
  end
end
return : Feasibility = YES

VI. Evaluation

To evaluate the proposed method, we prepared MPP informations of 100 mismatched PV arrays and applied our algorithm. We also applied an exhaustive searching algorithm for comparison. MPP information is setup as follows for each PV array. Each PV array has 6 - 24 panels and the number of

PV panels is randomly determined. Three current values are randomly assigned to the three modules in each panel from 3 - 8 current values, where the number of possible current values are also randomly determined for each PV array. The number of strings after reconfiguration is 3 - 6 and it is randomly determined for each PV array.

We first applied the method [8] to select configuration candidates. Condition **Feasibility 1** is applied to exclude a part of infeasible configurations, and then configurations that generate more than 77% of the maximum power are selected from the remainings as candidates as mentioned in [8]. In our experiments, 486 candidates are selected from 100 PV arrays as shown in Table VII. That is, for each PV arrays, 4 - 5 candidates are selected in average.

Table VII shows the results of feasibility checking by the proposed algorithm and the exhaustive search. Since the exhaustive search checks all the combinations of panel connections for each configuration candidate, it accurately identifies the feasibility of every configuration. Though the exhaustive search identified total 327 feasible configurations, the proposed method identified 324 configurations and 3 configurations are misidentified. The proposed algorithm had a few false negative cases, but its rate was less than 1%. That is, it can correctly identify feasibility for most cases. Indeed, the proposed algorithm found at least one candidate configuration as feasible among 4 - 5 candidates for each PV arrays.

The calculation was performed using an Intel i7-4770 Quad-Core processor and 16 GB of RAM memory. The time used by the proposed feasibility check method was 3ms, which is significantly lower than the 98s needed by the exhaustive search algorithm.

In conclusion, the proposed method provides a solution for checking the feasibility of PV configurations with an error lower than 1% but execution time are significantly lower than the one required by exhaustive search algorithm.

TABLE VII

COMPARISON BETWEEN EXISTED METHODS AND PROPOSED ALGORITHM

	Proposed Algorithm	Exhaustive search Algorithm
Number of PV Array	100	
Number of MPP candidates	486	
Feasible Candidates	324	327
Infeasible Candidates	162	159
Error Rate	0.62%	0%
Ave. times per Array	0.003s	98s

VII. Conclusion

Power generation of mismatched PV arrays can be recovered by reconfiguring connection of PV panels. In this paper, we introduced a feasibility check problem to configure PV panel connections from PV module connections and proposed an algorithm that efficiently and effectively solves the problem. The experimental results show that the proposed algorithm can identify feasible configurations more than 32,000X faster than the exhaustive search with around 0.6% errors. The proposed method is useful to identify or configure actual PV panel connections from PV module level specifications or connections. Since there are many works of PV module level reconfiguration and PV panel level reconfiguration using PV module level optimization as an intermediate solution, the proposed algorithm is useful to actually find PV panel connections from PV module level solutions. PV panel level reconfiguration is a practical and realistic solution for PV arrays composed of standard PV panels, and the proposed method contributes to find efficient and effective solution for PV panel level reconfiguration.

Reference

[1] 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.

[2] 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.

[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] 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.

[5] P. Udenze, Y. Hu, H. Wen, X. Ye, and K. Ni, “A reconfiguration method for extracting maximum power from non-uniform aging solar panels,” *Energies*, vol. 11, no. 10, p. 2743, 2018.

[6] 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.

[7] Y. Hu, J. Zhang, P. Li, D. Yu, and L. Jiang, “Non-uniform aged modules reconfiguration for large-scale pv array,” *IEEE Transactions on Device and Materials Reliability*, vol. 17, no. 3, pp. 560–569, 2017.

[8] 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.

[9] 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.

[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.