Irradiance-Driven Partial Reconfiguration of PV Panels

Daniele Jahier Pagliari, Sara Vinco, Enrico Macii and Massimo Poncino
Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy
name.surname@polito.it

***Abstract*---Adaptive reconfiguration of a photo-voltaic (PV) panel by means of a switch network is a well-known approach to tackle shading issues dynamically and with a reasonable cost. Most of these approaches assume however that the entire panel is reconfigurable, resulting in high installation costs due to the large wiring overhead required by this solution.**
In this work we propose an architecture in which only a portion of the panel is made reconfigurable, while minimizing the loss in the extracted power with respect to a fully reconfigurable solution. The key feature of our approach is the use of environmental (irradiance and temperature) data to determine the reconfigurable subset at design time.
Simulation results show that, by reconfiguring only about 50-70% of a panel, it is possible to achieve up to 45% power increase with respect to a static topology, while losing less than 5% power with respect to full reconfiguration.

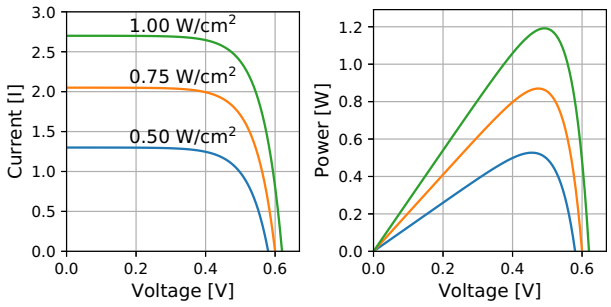
I. Introduction

One of the major sources of power losses in PV panels is the presence of shading conditions, induced by fixed or transient obstacles typical of urban environments [1]. These losses are mainly due to the electrical configuration of the panel: in each series string, even a single partially shaded module limits the overall string current, thus reducing the resulting power.
In order to mitigate such losses, two solutions are typically used, often in conjunction. At the individual PV module level, the use of bypass diodes partially limits the effects of shading [1]. At the panel level, *adaptive reconfiguration* by means of a switch network is a well-known approach that can overcome shading issues (or, more in general, conditions of variable irradiance) in a dynamic way. A number of works levering reconfiguration have been proposed in the literature and demonstrated on the field [2], and the relative architectures are quite mature and ready for commercial deployment. Existing solutions, however, assume that *the whole panel* is reconfigurable, i.e., any connection pattern of the modules is possible. Full reconfigurability can be an option for larger panels, but on smaller ones its installation costs can be hard to amortize; while the cost of switches might be negligible, the wiring overhead can become substantial [3].
As a matter of fact, full reconfiguration is somehow an over-design: for instance, modules that always have comparable irradiance (low or high) over time could be reliably connected in series without the need of reconfiguration. As this simple example suggests, the key to such selective reconfiguration is to exploit environmental data (mainly irradiance and temperature) that affect the power generation of a PV module.
In this work we address the issue of optimizing the selection of the reconfigurable portion of a PV panel by using historical irradiance and temperature data. We model the problem as an optimization framework in which we jointly minimize the number of required switches (a proxy for the total installation cost) and the loss of generated power with respect to a fully reconfigurable panel. More specifically, at design time, our method is based on forming *clusters* of PV modules that are similarly irradiated over time; the modules in each cluster are statically connected to each other, thus reducing installation costs. At runtime, our method combines these clusters, as opposed to single modules, to form the panel series strings.
Simulation results show that panels constructed with the proposed approach can achieve total extracted energy values comparable to those of a fully reconfigurable solution, by only reconfiguring ½ to ¾ of the modules.

II. Background

A. PV Cells, Modules, and Panels

A photovoltaic (PV) cell is described by a current-voltage (I-V) characteristic curve, which plots, at a given temperature, the variation of I and V as a function of the irradiance G [1] (Figure 1a). As G increases, the open-circuit voltage V_{oc} (point corresponding to $I = 0$) increases logarithmically and the short-circuit current I_{sc} (point corresponding to $V = 0$) increases proportionally.



(a) I-V (b) P-V

Fig. 1: I-V/P-V curves of a PV cell for different irradiance.

By multiplying the values of the current axis by the voltage values, we obtain the P-V curve (Figure 1b), whose maximum (the so-called Maximum Power Point - MPP), corresponds to the optimal (I,V) conditions for extracting power from the panel. These conditions are typically enforced by an ad-hoc circuit called MPP tracker (or MPPT).
A *PV module* is the interconnection of a number of cells in series (usually 36 or 72). A *PV panel* interconnects various PV modules according to a series/parallel organization in order

to increase the output power; as a general rule of thumb, increasing the number of series (parallel) modules increases the output voltage (current).
Typical panel interconnections are the parallel-of-series (PS), in which series strings of modules are connected in parallel, or the series-of-parallel (SP or total-cross tied - TCT), in which parallel modules are connected in series.

B. Partial Shading and Reconfigurability

Besides topology, the actual output power of a panel will also depend on the different irradiance conditions of each module. This is particularly critical for series strings of modules. Let us consider a string of m modules, and assume that one of the modules is partially shaded (thus extracting smaller power than the others), while all other modules have maximal irradiance. Given the series connection, the module providing the smallest current would restrict the available current of the string, i.e. $I_{string} = \min_{i=1,\dots,m} I_{module,i}$. Conversely, the voltage of the string would not be affected significantly, considering that the module voltage has a weaker dependence on irradiance, hence $V_{string} \approx \sum_{i=1,\dots,m} I_{module,i}$. In practice, however, such a worst-case current situation should be avoided. The shaded, low-irradiance module tends to behave like a semiconductive resistance, and dissipates the power being generated by the high-irradiance modules, thus risking possible damage.
This problem is usually tackled by adding a *bypass diode* in reverse parallel to each module [1]. These diodes provide an alternative path for the current when a module is shaded, thus avoiding the above ''clogging'' effect. The price paid is a loss in total voltage of the string due to the bypass of the shaded module, plus the voltage drop of the conducting diode ($\approx 0.6V$). Moreover, the presence of diodes yield I-V curves of the string with multiple ''knees'', thus generating sub-optimal local maxima that complicate the MPPT algorithm [4].
Figure 2 summarizes the above discussion for a small example with $m = 4$. The example clearly shows how bypass diodes sacrifice the voltage of one module in order to keep a higher current in the string.

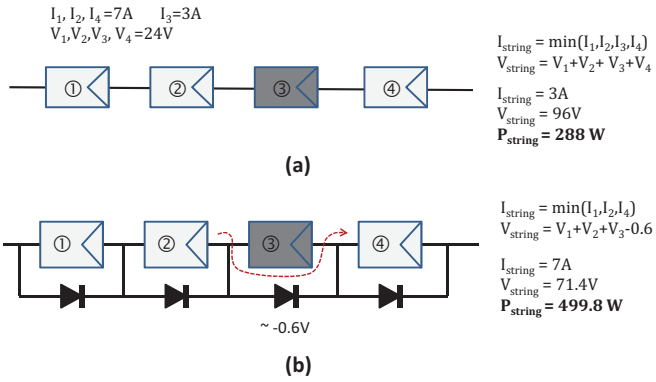


Fig. 2: Series string with a shaded module: (a) diode-less; (b) using bypass diodes.

When it comes to the whole panel, however, shading conditions can clearly affect multiple strings; even with diodes, this could have a significant impact on the total extracted power. For this reason, the idea of reconfigurable panels via a switch matrix has been devised, in which the topology can be adapted to the irradiance condition to reduce power losses.

III. Related Work

The literature on reconfigurable PV panels is quite vast. The general principle driving reconfiguration is quite straightforward: based on estimated or measured irradiance (and, in some cases, temperature as well), the interconnection among modules is periodically rewired so to minimize the variance of irradiance in each series. This step typically encompasses *sorting* PV modules by irradiance and building the series strings according to that order. The various methods differ essentially in (1) the actual metric used for the sorting, (2) the algorithm used to optimize the metric, (3) the flexibility of the final topology (i.e., the possibility of altering the series/parallel cardinality), and (4) the characteristics (in particular, number and type) of the switches. For an exhaustive comparison of the various strategies, we refer the reader to the recent survey of [2].
In this section we will analyze in deeper details the (few) solutions that split the panel into *static* and *reconfigurable* portions, which are closer to our approach.
In [5], the output power of a baseline, non reconfigurable panel portion is optimized using *redundant* modules. The authors demonstrate a 3x2 topology in which a 2x2 static part is completed by an extra module for each of the 2 series, selected out of a pool of 6 spare modules so to obtain the best operating conditions. While this method yields good improvement, its effectiveness is affected by the number of available spare modules; moreover, it incurs obvious extra costs as 4 of the 10 total modules are disconnected at any time.
The method of [6] assumes a TCT interconnection and connects extra modules in parallel to the rows with lower voltages (i.e., with lower irradiance). The resulting panel will thus exhibit a variable topology, and rows with uneven numbers of modules. Both previous approaches, therefore, do not really envision a *partial reconfiguration*: rather, they aim at complementing a baseline fixed panel using spare modules to compensate for irradiance imbalance.
The approach closest to ours is the one of [3]; here, physically contiguous PV modules are grouped in macro-cells of fixed size, which are then used as atomic units of reconfiguration. Although these macro-cells are somewhat similar to our clusters, in our case they are allowed to have variable sizes and be physically spread, thus permitting more flexibility in adapting to the historical irradiance behavior. Moreover, [3] allows variations in the series/parallel topology of the panel at runtime. While this enhances flexibility, it complicates the power extraction process, requiring a DC/DC converter or an inverter to support a wide range of currents and voltages, thus hindering its efficiency. For these reasons, in our work, reconfiguration is performed whilst keeping the overall panel topology constant.

IV. Irradiance-Driven Partial Reconfiguration

A. Partially Reconfigurable PV Panels

In the following we will assume a m-series, n parallel PS topology ($m \times n$ in short). Let $N = m \times n$.
Figure 3 shows the conceptual structure of a fully reconfigurable panel, i.e., in which any module can be attached to

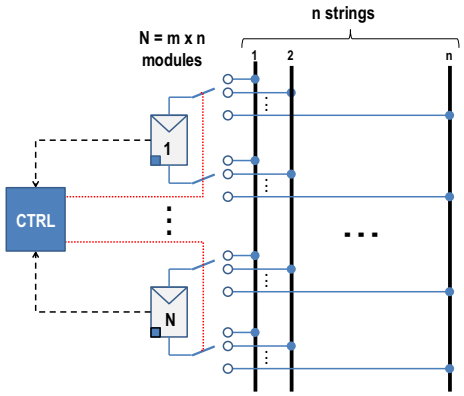


Fig. 3: A Fully Reconfigurable PV Panel.

any string. The drawing depicts the electrical architecture of the panel, not the actual spatial placement of the modules. The bus-based architecture shown is typically preferred to a fully-connected one for obvious modularity reasons; with the former, full reconfiguration requires $2N$ n-pole switches. The figure also shows the control path that implements the reconfiguration; at run time, a controller receives from each module (dashed black lines) the readings from the attached irradiance and temperature sensors (small blue squares). Based on sensor information from all modules, the controller runs an algorithm that determines to which string a module needs to be connected, and actuates the corresponding switches (red dotted lines). Clearly, the two switches associated to a panel are always connected to the same string.

While the sheer cost of switches and cables of this implementation could be not so relevant, full reconfigurability can have an impact on the installation costs and times, as all the extra wiring complicates the deployment of the system. As anticipated in Section I, however, a full reconfiguration can be avoided with marginal power losses. Indeed, the objective of reconfiguration is to assign modules with comparable irradiance to the same series string; however, there might exist regions of the panel area that consistently have similar (high or low) irradiance values over time. Modules in those areas can be *hard-wired* to the same string, without the need of being made reconfigurable.

In order to illustrate this concept, consider the example of Figure 4. The drawing only represents the *spatial* layout of a 18-module panel, which will be then electrically connected to form a $m \times n = 6 \times 3$ topology as shown in Figure 3. For simplicity, assume that no diodes are used, so the current in each string is constrained by the least irradiated module.

Suppose that (e.g. because of orientation) the area with yellow (red) background represents a region of the panel that most often has a relatively high (low) irradiance. Evidently, even with full reconfiguration, the modules belonging to the same area will most likely end up in the *same series string*; indeed, as explained in Section III, most controllers try to minimize the irradiance *variance* in each string, in order to reduce current bottlenecks. Consequently, the modules within each of these regions can be safely *hard-wired in a series*, forming what we call a *cluster* of modules.

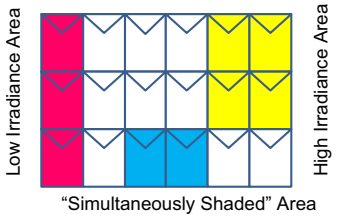


Fig. 4: A Case for Partial Reconfiguration.

There are however other modules that can be hard-wired into a cluster, i.e., those placed in regions whose irradiance varies widely over time, for example due to shading, but is *uniform across the region at all times* (blue background in Figure 4). To form a string, these clusters will be connected to either the high-irradiance ones (yellow) or low-irradiance ones (red) at different times, depending on their measured irradiance.

The remaining modules (white background) are those whose irradiance profile is poorly correlated to that of any other one. These cannot be hard-wired with any other module, without constraining them to frequently operate in sub-optimal conditions; thus, they should be reconfigured individually (singleton clusters). Notice that the presence of bypass diodes does not change this basic principle, since they just mitigate the worst-case effect of shaded modules.

In general, the modules in each hard-wired series need not be physically contiguous, although this is the most common scenario. Also, the number of modules in each region might be different from the desired series length m. Thus, the reconfiguration controller should be modified to handle ''partial strings'', as will be explained in Section IV-C.

Two observations are worth being drawn from this conceptual example. First, the decision about which modules should be hardwired and which should not cannot be based only on their *average* irradiance over the considered time interval; vice versa, the entire irradiance time profile is required. Second, and more important, since the above decision must be made at design time, prior to the installation of the system, *historical estimates* of irradiance in the target area are needed to locate regions such as those depicted in Figure 4.

B. Identification of Reconfigurable and Hard-wired Modules

We model the design-time problem of finding regions of the panel that do not need reconfiguration as a *graph clustering*. Our algorithm only optimizes the electrical *connection* between PV modules, and assumes that the *position* of the modules in the installation area is pre-determined and cannot be changed. The typical scenario is that of a roof fully covered with modules, in which positions are not a degree of freedom (although the method also applies to ''sparse'' installations).

The main input to our optimization procedure is a 2-d array $R = [r_{i,t}],\ 0 \leq i < N,\ 0 \leq t < T$, representing the historical irradiance time profile of the roof; specifically, each element contains the estimated irradiance in the area occupied by module i at time t.

From this input, we build a weighted, complete, undirected graph $G = (V, E)$ in which the vertices $v_i \in V,\ 0 \leq i < N$ correspond to the PV modules, and the edge weights $e_{i,j} \in$

$E,\ 0 \leq i < j < N$ represent the irradiance ''similarity'' among modules i and j. We compute such similarity as:

$$e_{i,j} = \frac{T}{\sum_{t=0}^{T} \|r_{i,t} - r_{j,t}\|} \quad (1)$$

that is, the inverse of the average irradiance difference (in absolute value) among the two modules over time. Two modules with similar irradiance in the entire time span will therefore generate a larger $e_{i,j}$, and vice versa.
Next, the actual clustering is performed according to a greedy algorithm as follows:

1) The largest edge weight $e_{i,j}$ is selected, and its source and destination vertices are considered. If they respect all constraints (see below), the corresponding modules are added to the same cluster.
2) Vertices v_i and v_j are *merged*, i.e., they and their connecting edge are eliminated, and a new node v_k is created representing the newly generated cluster. The weights of the outgoing edges from v_k are computed as: $e_{k,l} = min(e_{i,l}, e_{j,l}),\ \forall 0 \leq l < N,\ l \neq i, j.$
3) Steps 1 and 2 are repeated until there is at least one edge respecting the constraints.

The min function used to update the edges comes from the observation that, if a new module has to be added to a partially formed cluster, it should have good similarity with *all* modules already in it. As example of the first two steps of the proposed graph clustering for a toy graph is shown in Figure 5.

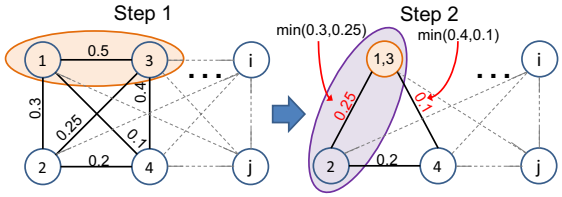


Fig. 5: Example of steps in the PV module clustering algorithm.

When deciding whether to merge two vertices, we impose two constraints. First, the size of the cluster resulting from the merge should always be smaller than m, i.e., the desired series string length. Second, since only modules with very similar irradiance profiles should be hard-wired, just a (possibly small) fraction of the largest edges weights is considered for merging. Specifically, we impose a percentile threshold ϵ, where for example $\epsilon = 10$ corresponds to considering only the largest 10% weights in the original graph. The threshold is a tunable parameter of our algorithm, and allows one to explore the trade-off between system complexity and the amount of reconfiguration. Large values of ϵ correspond to considering more nodes as ''similar'', hence yielding larger clusters, that is, more hard-wiring and less reconfiguration.
Despite the constraint on cluster size, large values of ϵ may anyway lead to a clustering that does not allow the formation of a $m \times n$ configuration. For example, if a panel contains 12 modules, to be connected as $m \times n = 4 \times 3$, the algorithm might yield 4 clusters of size 3 each, which clearly cannot be combined to match the target topology. Our algorithm checks for these corner cases and simply returns an error when they occur - no solution can be found. These situations are however infrequent for typical values of ϵ.

C. Runtime Partial Reconfiguration Algorithm

A good heuristic that proved effective for fully-reconfigurable PS topologies consists of *sorting* modules by decreasing irradiance (as measured by the sensors), and assigning the first m modules to the first series string, the next m to the second string, and so on [2]. However, this principle cannot be used as is when cluster of modules are present. For instance, if $m = 4$ and at a given time the two most irradiated clusters have size 2 and 3, the controller cannot simply connect them, as this would modify the topology, creating a string of length 5. To solve this issue we adopt the recursive search algorithm reported as pseudo-code in Figure 6, where $+$ and $-$ signs indicate concatenation and difference operations among sets.

```
 1: procedure RECONFIGURATION(sensor_readings)
 2:     ordered_clusters = argSortDescending(sensor_readings);
 3:     for p ∈ [1:n] do // iterate over the n series strings
 4:         (sel_clusters,--) = BUILDSTRING(∅, ordered_clusters);
 5:         ordered_clusters = ordered_clusters - sel_clusters;
 6:     end for
 7: end procedure
 8:
 9: procedure BUILDSTRING(current, available)
10:     if totalStringLength(current) = m then   //string completed
11:         return (current, SUCCESS)
12:     else if totalStringLength(current) > m then //string exceeded
13:         return (∅, FAILURE)
14:     else
15:         for i ∈ 0:length(available) do
16:             (clusters, outcome) =
17:                 BUILDSTRING(current + available[i], available[i+1:end]);
18:             if outcome = SUCCESS then
19:                 return (clusters, SUCCESS)
20:             end if
21:         end for
22:     end if
23: end procedure
```

Fig. 6: Runtime Partial Reconfiguration Algorithm.

The top-level procedure (*Reconfiguration*) still begins by sorting the clusters by decreasing irradiance (line 2). If each module is equipped with one sensor, the minimum among the readings from each cluster is used.
Next, each of the n series strings in the $m \times n$ topology is created through the recursive procedure (*BuildString*). The latter takes two inputs: the current tentative set of clusters used for building the series ($current$), initially equal to the empty set, and the remaining available clusters ($available$), sorted by decreasing irradiance. The two trivial cases of the recursion occur when the current set contains m modules (line 11), in which case a solution has been found, and when the series length has been exceeded (line 13), i.e., the solution is unfeasible.
If none of the two trivial cases occurs, the procedure iterates through the available clusters (line 15), starting from the one with highest irradiance, and calls itself after adding one cluster to the current set (lines 16-17). As soon as a valid solution is found, it is immediately returned (line 19), since due to the irradiance-based sorting, no further combination of clusters could yield a string with better overall irradiance.

Notice that, as unfeasible clusterings have been excluded at design time (Section IV-B), the procedure is guaranteed to find at least one solution (although possibly sub-optimal from the point of view of energy) before the end of the for loop.

V. Experimental Results

A. Power Model of the PV Panel

Our optimization requires a power model for individual PV modules that can also accounts for the presence of bypass diodes. A practical model that satisfies these requirements is the one proposed in [4]. This model derives the power $P(G,T)$ of a PV module from data available in a datasheet as a function of irradiance G and temperature T. One important aspect of the model is that it also account for the inter-dependence between G and T (higher irradiance raises temperature, which has some effect on the power output of the module); moreover, it also includes an aging factor for being used on existing installations. The power for a string of modules is obtained by carefully accounting for the presence of bypass diodes, by summing the individual I-V curves of the modules in the string; the MPPT is then extracted from the I-V curve of the complete string. Finally, the model sums the I-V curves of each string to achieve the global power value.

B. Irradiance and Temperature Data Generation

As mentioned, the design-time decision presented in Section IV-B requires an estimate of environmental data. To this purpose, we use historical irradiance/temperature data obtained with the GIS-based infrastructure of [7]. Input GIS data are expressed through a *Digital Surface Model* (DSM), which is a high-resolution raster image representing terrain elevation of the building of interest. The DSM allows to recognize obstacles over the surface (e.g. chimneys) and to estimate the evolution of shadows over time, with 15-minute temporal resolution. The irradiance trace over time is obtained by combining weather data, retrieved from weather stations, with the shadow model. Temperature data is also gathered from weather stations. The temperature distribution over the roof is derived from the ambient temperature T_{amb} with a term depending on G, as follows [8]: the temperature $T(G)$ over a grid cell of the roof is modeled as $T(G) = T_{amb} + k \cdot G + T_c$, where $k = \frac{\alpha}{h_c} = 0.05 \frac{W}{K \cdot m^2}$ is the ratio between the absorptance of the roof and its radiative loss factor, T_c is the operating cell temperature, calculated as in [8], and G is the irradiance estimated through the shadow model. The same dependency is incorporated in the power model of [4] described in Section V-A.

C. Experimental Setup

We have implemented our optimization framework using Mathworks MATLAB R2018a on a PC equipped with an Intel Core i7@2.2Ghz and 32GB RAM.

We have used the same power model parameters of [4], corresponding to a Mitsubishi PV-MF165EB3 module [9]. We have then tested our approach on two lean roofs of industrial buildings, whose characteristics are reported in Table I. The number of panels that can be installed on each roof is computed assuming total coverage of roof area with panels having a ''portrait'' orientation. Irradiance and temperature profiles for the roofs have been obtained with the method of Section V-B using 1 year of historical data.

TABLE I: Target roofs characteristics

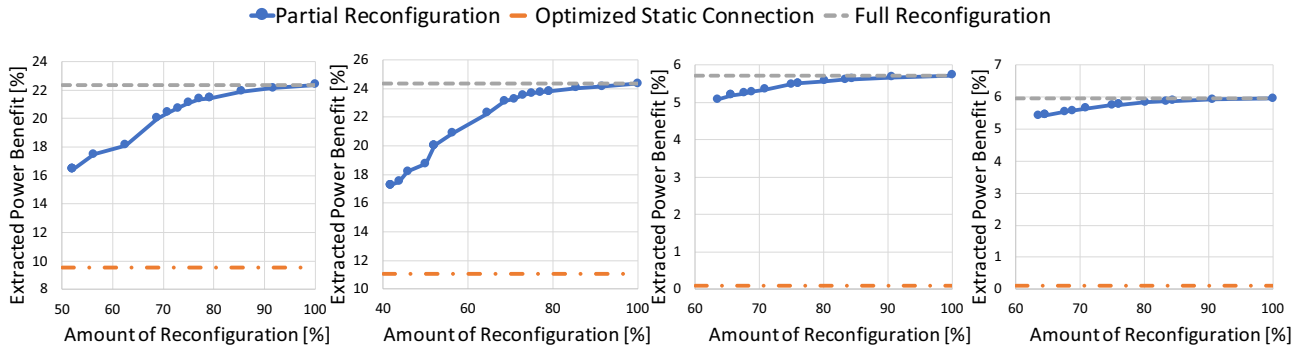
Roof	Orientation	Inclination	Size	# Panels
Roof 1	S-S/W	26°C	8m×12m	48
Roof 2			8m×24m	96

D. Simulation Results

We have compared our proposed partial reconfiguration with three alternative solutions: (i) a straight-forward *static* (i.e. not reconfigurable) connection in which modules are added to the same series based on physical proximity, in a column-wise fashion; (ii) an optimized static connection based on aggregated knowledge of historical irradiance proposed in [10], [11]; (iii) a fully-reconfigurable bus-based connection, which uses the *sorting* runtime algorithm described in Section IV-C. For partially or fully reconfigurable solutions, we assume a reconfiguration interval equal to the time granularity of the available irradiance/temperature data, i.e., 15 minutes.

The results of this comparison are reported graphically in Figure 7. The graphs refer to two different PS configurations for each roof, thus showing that our approach is effective regardless of the panel topology. The horizontal axis of each plot reports the ''amount of reconfiguration'', i.e. the number of reconfigurable elements (single modules or clusters), normalized to the one of a fully-reconfigurable panel. This value is a proxy of the total system complexity (number of switches, wiring, etc.). The vertical axis reports the extracted power gain of our approach *with respect to the static column-wise connection*. The solution of [10] and the fully-reconfigurable panel are represented as two horizontal lines, as they both have a fixed amount of reconfiguration (0% and 100%, respectively). The points in the curves correspond to different values of ϵ (see Section IV-B). Specifically, ϵ has been increased in steps of 0.5%, starting from 0% (i.e. no clustering) and until the algorithm generated a clustering that guarantees the existence of a feasible connection. Therefore, different roof/topology combinations resulted in a different number of points.

The two target roofs have different irradiance characteristic, thus producing very different result. For Roof 1, full reconfiguration produces a significant power benefit with respect to a static connection (≈22% or 24% depending on topology). Thanks to our partial reconfiguration algorithm, however, a very similar benefit (<2% difference) can be reached while reducing the number of reconfigurable elements to ≈70% (i.e., 34 versus 48). Moreover, even reducing the reconfigurable elements to less than 50%, the extracted power is still significantly larger than the one obtained by the optimized static connection of [10]. For Roof 2, conversely, the benefit of full reconfiguration with respect to a static connection is much more limited (≈ 6%) due to less partial shading. On the other hand, this uniformity of irradiance causes the trade-off curves to be quite flat: our method can extract power within 1% of the full reconfiguration result with only ≈ 65% of reconfigurable elements. Notice also that in this case the method of [10] does not provide any benefit with respect to the straight-forward connection.



(a) Roof 1 ($m \times n = 8 \times 6$) (b) Roof 1 ($m \times n = 12 \times 4$) (c) Roof 2 ($m \times n = 12 \times 8$) (d) Roof 2 ($m \times n = 16 \times 6$)

Fig. 7: Extracted power benefit versus amount of reconfiguration for the two target roofs.

An example of the result generated by our clustering, corresponding to the 70% reconfiguration point for Roof 1, for the $m \times n = 8 \times 6$ topology, is reported in Figure 8. Modules colored in the same way represent hard-wired partial series, whereas white modules are the ''singleton clusters''. As expected, the algorithm tends to aggregate physically close modules, following irradiance similarities.

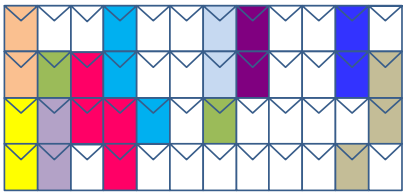
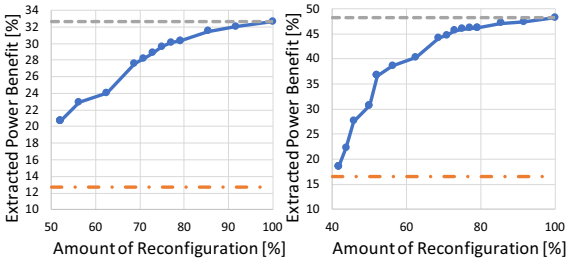


Fig. 8: Example of clustering result. Roof 1, $m \times n = 8 \times 6$, $\epsilon = 4\%$, reconfigurable elements $= 34$ (i.e. 70%).

Finally, Figure 9 shows the same plots of Figure 7 for Roof 1, but assuming a diode-less PV panel model.



(a) Roof 1 ($m \times n = 8 \times 6$) (b) Roof 1 ($m \times n = 12 \times 4$)

Fig. 9: Results for diode-less panel (Roof 1).

As expected, full reconfiguration is even more effective for diode-less panels, producing an extracted power benefit of 33-48% depending on topology. Nonetheless, our partial reconfiguration approach is still attractive in this case; at 70% reconfiguration, the extracted power increase remains within 5% from the full result, and at 50% reconfiguration, it is still significantly larger (approximately double) with respect to the approach of [10]. These results prove that our proposed method can be adopted both on standard and diode-less panels. Moreover, they show that ϵ is an effective parameter to trade-off the extracted power with system complexity. Even with the largest of the two roofs, the clustering of Section IV-B takes $\approx$ 1.5s $\pm$ 0.3s on the target platform (mean and standard deviation computed over 1000 runs), whereas the runtime algorithm of Section IV-C takes $\approx$ 0.0025s $\pm$ 0.0004s per reconfiguration.

VI. Conclusions

As a way to tackle shading variations over time, full reconfigurability of a PV panel is an over-design because not all modules in the panel truly needs to be reconfigured. In this work, we have shown that by exploiting the temporal evolution of environmental data it is possible to reconfigurable only a subset of the modules while sacrificing a small amount of the total extracted power of a fully reconfigurable solutions. Our method is effective for different panel topologies, panel sizes, and irradiance conditions.

References

[1] A. S. and K. Jager and O. Isabella and R. van Swaaij and M. Zeman, *Solar Energy: The physics and engineering of photovoltaic conversion, technologies and systems*. UIT Cambridge, 2016.

[2] D. La Manna, V. Li Vigni, E. Riva Sanseverino, V. Di Dio, and P. Romano, ''Reconfigurable electrical interconnection strategies for photovoltaic arrays: A review,'' *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 412 -- 426, 2014.

[3] Y. Wang, X. Lin, M. Pedram, J. Kim, and N. Chang, ''Capital cost-aware design and partial shading-aware architecture optimization of a reconfigurable photovoltaic system,'' in *Proc. of IEEE DATE*, 2013, pp. 909--912.

[4] S. Vinco, L. Bottaccioli, E. Patti, A. Acquaviva, and M. Poncino, ''A compact PV panel model for cyber-physical systems in smart cities,'' in *Proc. of IEEE ISCAS*, 2018, pp. 1--5.

[5] G. Velasco-Quesada, F. Guinjoan-Gispert, R. Pique-Lopez, M. Roman-Lumbreras, and A. Conesa-Roca, ''Electrical pv array reconfiguration strategy for energy extraction improvement in grid-connected pv systems,'' *IEEE TIE*, vol. 56, no. 11, pp. 4319--4331, 2009.

[6] D. Nguyen and B. Lehman, ''An adaptive solar photovoltaic array using model-based reconfiguration algorithm,'' *IEEE TIE*, vol. 55, no. 7, pp. 2644--2654, 2008.

[7] L. Bottaccioli, E. Patti, E. Macii, and A. Acquaviva, ''GIS-based software infrastructure to model pv generation in fine-grained spatio-temporal domain,'' *IEEE Systems Journal*, 2017.

[8] F. Brihmat and S. Mekhtoub, ''PV cell temperature/PV power output relationships Homer methodology calculation,'' in *International Journal on Scientific Research and Engineering Technology*, vol. 1, 2014.

[9] *PV-MF165EB3 Datasheet*, Mitsubishi Electric, 2004.

[10] S. Vinco, L. Bottaccioli, E. Patti, A. Acquaviva, E. Macii, and M. Poncino, ''GIS-based optimal photovoltaic panel floorplanning for residential installations,'' *Proc. of IEEE DATE*, 2018.

[11] S. Vinco, E. Macii, and M. Poncino, ''Optimal topology-aware PV panel floorplanning with hybrid orientation,'' in *Proc. of ACM GLSVLSI*, 2018, pp. 491--494.