23rd International Congress on Modelling and Simulation, Canberra, ACT, Australia, 1 to 6 December 2019
mssanz.org.au/modsim2019

Optimal partitioning of photovoltaic modules on a curved solar collector

M. Kapsis[a], **A.R. Albrecht**[b] **and P.J. Pudney**[b c]

[a]*School of Information Technology and Mathematical Sciences, University of South Australia, Mawson Lakes, Australia*

[b]*Scheduling and Control Group, Centre for Industrial and Applied Mathematics, University of South Australia, Mawson Lakes, Australia*

[c]*Future Industries Institute, University of South Australia, Mawson Lakes, South Australia*

Email: kapmy005@mymail.unisa.edu.au

Abstract: The Australian Technology Network (ATN) group of universities designed and built a solar car to participate in the 2019 Bridgestone World Solar Challenge. The car is powered by 29 photovoltaic modules on its top surface. To get a useful voltage from the solar collector, modules are connected in series. However, the power generated by a group of modules in series is limited by the module with the lowest irradiance. Irradiance depends on the angle between the sun and the cell normal, which is influenced by the curvature of the collector and the position of the sun relative to the car. If all of the modules were facing the same direction then each module would receive the same irradiance and there would be no "series mismatch" losses, but the solar panel is curved for aerodynamic efficiency.

Our challenge was to partition the modules into groups so that the energy generated by the solar collector is maximised during a six-day journey across Australia. We describe two methods for partitioning modules into groups. Our mixed-integer programming model provides optimal solutions for a single time instant, but can not solve the problem for the entire journey. Using a Cross Entropy Optimisation method, we were able to find solutions that were within 0.03% of optimal for a single time instant, and could also find good solutions to the six-day problem.

Keywords: *Solar energy, maximising efficiency, optimal partitioning, integer programming, cross-entropy optimisation*

1 INTRODUCTION

The Australian Technology Network (ATN) group of universities designed and built a solar car to drive 3022 km across Australia in the 2019 Bridgestone World Solar Challenge. The car collects most of its energy from 322 photovoltaic cells arranged on the curved upper surface of the car (Figure 1).

The electrical power that can be generated by a cell depends on the angle between the cell normal and the sun, and on the angle between the cell normal and vertical. Because of the curvature of the solar collector surface, cells face a variety of directions and so the power that can be generated by each cell varies between cells, and changes with the location of the car along the route, the direction of the road, and the position of the sun.

The current that can be generated by a cell depends on the solar irradiance on the cell. To generate a workable voltage, cells must be connected in series. When cells are connected in series, the voltages of the cells are summed but the current that can flow through the cells is limited by the cell with the lowest irradiance. For ease of manufacturing, the cells are organised into 29 modules (Figure 2), with cells in each module connected in series.

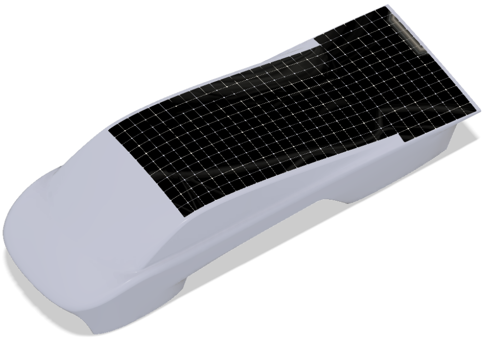

Figure 1. Layout of cells on the ATN solar car.

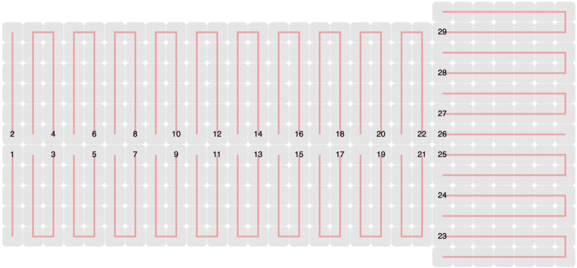

Figure 2. Layout of modules on the ATN solar car.

For the purpose of optimising the configuration of the solar collector, it is reasonable to assume that each module operates at a fixed voltage and, at any instant in time, a fixed current (Pudney, 2000). The power generated by an individual module is the product of the module voltage and the module current. When a group of modules is connected in series, however, the total voltage of the group is the sum of the module voltages, but the current flowing through the group will be the minimum of the module currents. Thus the power generated by a group of modules depends on the minimum module current within the group. We assume that there are no wiring losses.

The challenge is to partition modules into groups in a way that maximises the total energy generated during some defined time interval. An obvious choice is to have one module per group, but for practical reasons each group must have a minimum voltage $V_{\min}$, which will require several modules in each group.

In this paper we formulate the problem of partitioning the modules into groups so that energy collected as the car moves from Darwin to Adelaide is maximised. Section 2 formulates the problem as a mixed-integer program (MIP). However, the MIP becomes intractable for even a small number of time instances, and we wish to solve the problem considering the power generated at each of the 3022 kilometres from Darwin to Adelaide. In Section 3 we develop a method based on Cross-Entropy Optimisation that is able to find good solutions to the full problem within a few minutes.

The method described in this paper is also applicable to any curved solar collector where we wish to maximise the energy collected over an extended period, such as an annual cycle.

2 MIXED-INTEGER PROGRAMMING MODEL

2.1 Problem Formulation

Suppose we have n photovoltaic modules on our solar car. The current generated by each module will vary as the car moves along the route and as the sun moves across the sky. We discretise the journey into r intervals of 1 km, so for the 3022 km journey from Darwin to Adelaide, $r = 3022$. Module j is assumed to generate constant current I_{jk} over the interval k, and has voltage V_j. Clear-sky solar irradiance models from Duffie and Beckman (1980), scaled to conform to historical daily irradiation from the Australian Government Bureau of Meteorology (2019), were used to estimate module current I_{jk} for each kilometre of the journey. The car will be travelling at a constant speed of 75 km/h, so each kilometre will have duration $\Delta t = 48$ seconds.

We wish to partition the modules into groups. We introduce a binary decision variable

$$x_{ij} = \begin{cases} 1 & \text{if modules } i \text{ and } j \text{ are in the same group} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i, j \in \{1, \ldots, n\}.$$

This formulation has two nice properties:

- it does not pre-suppose the number of groups
- it does not require the groups to be labelled and so avoids producing identical solutions that differ only in the order of the groups.

The voltage of a group is the sum of the voltages of the modules within the group. We require the voltage in each group to be greater than a specified minimum voltage:

$$\sum_i V_i x_{ij} \geq V_{\min}, \quad \forall\, j \in \{1, \ldots, n\}.$$

Additional constraints are required to correctly model groups. Module j is always in a group with itself:

$$x_{jj} = 1, \quad \forall\, j \in \{1, \ldots, n\}.$$

If module i is in a group with module j, then module j is in a group with module i:

$$x_{ij} = x_{ji}, \quad \forall\, i, j \in \{1, \ldots, n\}.$$

If module i is in a group with module j, and module j is in a group with module ℓ, then module i must be in a group with module ℓ. That is, we have transitivity:

$$x_{ij} + x_{j\ell} \leq x_{i\ell} + 1, \quad \forall\, i, j, \ell \in \{1, \ldots, n\}.$$

The power generated by the entire photovoltaic panel during interval k is

$$P_k = \sum_i V_i \delta_{ik}, \quad \forall k$$

where the variable δ_{ik} indicates the minimum current during interval k of the group to which module i belongs. The value of δ_{ik} is constrained by

$$\delta_{ik} \leq I_{jk}, \quad \forall\, i, j \in \{1, \ldots, n\}, k \in \{1, \ldots, r\} \text{ such that } x_{ij} = 1.$$

The objective is to maximise the total energy produced while driving, which is given by

$$E = \Delta t \sum_k P_k. \tag{1}$$

In this formulation we ignore energy collected while the car is stationary.

2.2 Results

We began by solving the mixed-integer programming model with two examples at two different times of the day. When the car is 315 km from Darwin, the sun is almost overhead. The optimal grouping is the nine-group layout shown in Figure 3. The mean power generated during the 48-second interval is 1172.4 watts. In contrast, when the car is 617 km from Darwin, the sun is low in the eastern sky. The optimal grouping generates 731.1 watts. The optimal grouping is shown in Figure 4, and is also nine groups.

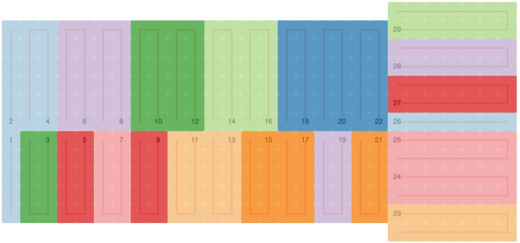

Figure 3. Optimal layout for 315 km.

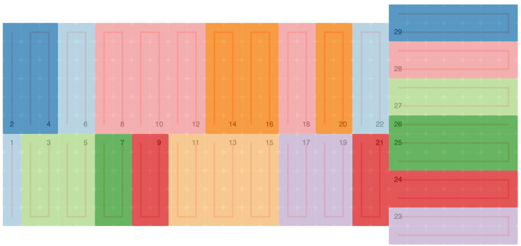

Figure 4. Optimal layout for 617 km.

To solve the mixed-integer programming model we used the MiniZinc constraint modelling language with the OSICBC 2.9/1.16 solver. The first problem (315 km) solved in around 3 minutes. The second problem (617 km) solved in around 1.5 minutes.

A key observation is that the optimal grouping is different at different times. We would expect that the optimal layout for 315 km is not necessarily appropriate for 617 km. This is the case; the first layout generates 714.9 W during the interval beginning at 617 km, compared to the optimal value of 731.3 W produced by the second layout.

We want a single layout that is optimal for the entire journey. Unfortunately the entire problem with 3022 km is too large for MiniZinc to solve, and so we turn to a probabilistic search technique. The optimal solutions obtained by MiniZinc for small problems will be useful when validating the solutions obtained by the probabilistic method.

3 CROSS-ENTROPY OPTIMISATION

Cross-Entropy Optimisation is a probabilistic search technique for finding good candidate solutions to discrete optimisation problems. The method, developed by Pieter-Tjerk de Boer and Rubinstein (2005), uses a probability distribution P to generate a set of candidate solutions. The best solutions (according to some metric) are used to update P so that future candidate solutions are better. The process iterates until a steady state is found, and is expected to produce solutions that approach optimality.

3.1 Problem formulation

For our problem, a candidate solution is represented by a permutation of the list of modules. We then use a deterministic procedure to partition the modules into groups. To generate a module grouping we go through the list in order, starting a new group when the sum of the voltages in the group exceeds the minimum voltage required for a group. Any partial group left over at the end is combined into the previous group. To give a simple example, suppose we have the following permutation of five modules with corresponding voltages listed below and $V_{\text{min}} = 10\,V$:

Modules:	2	4	3	1	5
Voltages:	4	5	6	5	6

The two groups will be $\{2, 4, 3\}$ with voltage 15 V and $\{1, 5\}$ with voltage 11 V.

The distribution used to generate candidate solutions is represented by an $n \times n$ matrix P. Element P_{ij} is the probability that module i will be placed in position j of the permutation. Initially each module has equal probability of being in any position, that is, the entries in each row of P are $1/n$.

We illustrate how to use P to generate a candidate solution with a small example where $n = 5$. We start with

$$P = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}.$$

Matrix S is an interim matrix used to generate a candidate solution. Initially $S = P$. Suppose we randomly choose to place module 1 in the fourth position. We then update S to ensure that module 1 will not be placed in any other position and that no other module will take the fourth position. We also normalise the entries in the remaining rows:

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix}.$$

We continue this process until all modules have been assigned a position. For our example, this gives

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrix S now defines a permutation of modules; row i defines the position of module i in the permutation. For our example, the permutation of the modules is $[2, 4, 3, 1, 5]$. We then use the deterministic procedure described earlier to partition the modules into two groups: $\{2, 4, 3\}$ with voltage 15 V and $\{1, 5\}$ with voltage 11 V. The energy produced over the entire journey by this grouping is calculated using Equation (1).

We repeat the procedure to generate 100 different candidate solutions. The top 10% of solutions are selected as elite candidates. We calculate the probability distribution Q of these elite candidates, then update the probability matrix with $P_{\text{new}} = \alpha Q + (1 - \alpha)P_{\text{old}}$. The smoothing parameter α is used to avoid converging too quickly to a local optimum. We used $\alpha = 0.7$. The procedure repeats until P converges, or a maximum number of iterations is reached.

3.2 Results

We start by comparing the best layouts obtained from the Cross-Entropy Optimisation method with the optimal layouts from the MIP for 315 km and 617 km. The Cross-Entropy Optimisation method is quick for a single kilometre, converging to a solution in around 5 seconds and roughly 70 iterations. Consequently, we performed 50 runs of the Cross-Entropy Optimisation method and kept the best solution from each run. We report the best solution, as well as the range of the best 50 solutions, below:

- **315 km:** Range: 1171.6 W to 1172.4 W. The best solution is the same as the optimal MIP solution.
- **617 km:** Range: 727.2 W to 730.9 W. The best solution is within 0.03% of the optimal MIP solution.

The Cross-Entropy Optimisation method slows down when finding a good layout for the entire journey because the energy generated by each candidate solution must be calculated for each of the 3022 kilometres and then summed. The layout corresponding to the best solution obtained by the Cross-Entropy Optimisation method is shown in Figure 5. This layout has ten groups and generates 923.2 W on average over the entire journey.

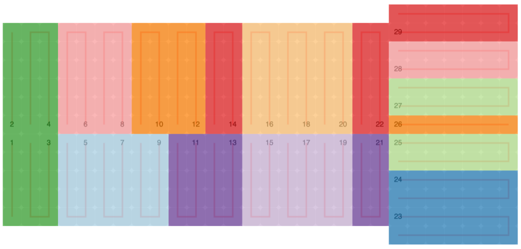

Figure 5. Optimal layout for the entire journey from Darwin to Adelaide.

We can also compare the solutions from both methods to the mean power that would be generated if no grouping is required, that is, if each module is placed in a group by itself. This is a theoretical upper bound on the optimal solution.

- **315 km:** Without grouping, the power is 1176.3 W. The optimal layout (obtained from both the MIP and from Cross-Entropy Optimisation) has a power loss of 0.3%.
- **617 km:** Without grouping, the power is 740.1 W. The optimal layout from the MIP and the best layout from Cross-Entropy Optimisation both have a power loss of 1.2%.
- **Entire journey:** Without grouping, the power is 937.8 W. The best layout from Cross-Entropy Optimisation has a power loss of 1.6%.

4 CONCLUSION AND FUTURE WORK

We have described two methods to partition the photovoltaic modules on a solar car into groups so that the energy generated during a six-day journey is maximised. The Cross-Entropy Optimisation method is fast and generates a module grouping for the entire journey that is within 1.6% of the theoretical upper bound. The same method could be used to partition modules into groups for a stationary curved solar collector in order to maximise the energy collected over an annual irradiance cycle.

There are two improvements that could still be made to the Cross-Entropy Optimisation method:

- Rather than combine any partial group left at the end with the previous group, we could check whether combining it with another group is better.
- When generating groups from a module permutation, we stop adding modules to each group as soon as the minimum voltage requirement is met. It is possible that larger groups may be better.

Pudney (2000) solved a similar problem using probabilistic search techniques and number partitioning methods, but for a problem with only 70 time intervals. The calculation duration is not reported.

An extended version of this model that takes into account power conversion losses associated with each group was used by the ATN Solar Car Team to design the wiring for their car.

ACKNOWLEDGEMENTS

This project was conducted as part of a Summer Vacation Research Scholarship sponsored by the Australian Mathematical Sciences Institute. We also thank the ATN Solar Car team for an interesting problem and for modelling advice.

REFERENCES

Australian Government Bureau of Meteorology (2019). Australian Government Bureau of Meteorology website. `http://www.bom.gov.au/climate/data-services/solar-information.shtml`. Accessed on 10-09-2019.

Duffie, J. A. and W. A. Beckman (1980). *Solar Engineering of Thermal Process* (4 ed.). New York: John Wiley and Sons.

Pieter-Tjerk de Boer, Dirk P Kroese, S. M. and R. Y. Rubinstein (2005). A tutorial on the cross-entropy method. *Annals of Operations Research 134*, 19–67.

Pudney, P. J. (2000). Wiring a curved photovoltaic array. In *Solar 2000: Proceedings of the 38th Annual Conference of the Australian and New Zealand Solar Energy Society*, pp. 381–391.