AMSI VACATION RESEARCH SCHOLARSHIPS 2018-2019

Optimal partitioning of photovoltaic modules on a curved solar collector

Maria Kapsis

Supervised by Amie Albrecht & Peter Pudney
University of South Australia

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Australian Government
Department of Education and Training

Abstract

The ATN group of universities is building a solar car to participate in the 2019 World Solar Challenge. The car will be powered by 29 photovoltaic modules on its top surface. To get a useful voltage from the solar collector, the modules must be connected in series. We can minimise series mismatch losses, due to the curvature of the solar panel, by using power optimisers with small groups of modules, and connecting the outputs of the power optimisers in series.

The challenge is to partition the modules into groups so that the energy generated during a six-day trip across Australia is maximised.

This report investigates the optimal module partitioning for a given sun position and for the entire race. We modelled this problem as an integer linear programming model and solved with a cross entropy probabilistic search technique.

1 Introduction

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

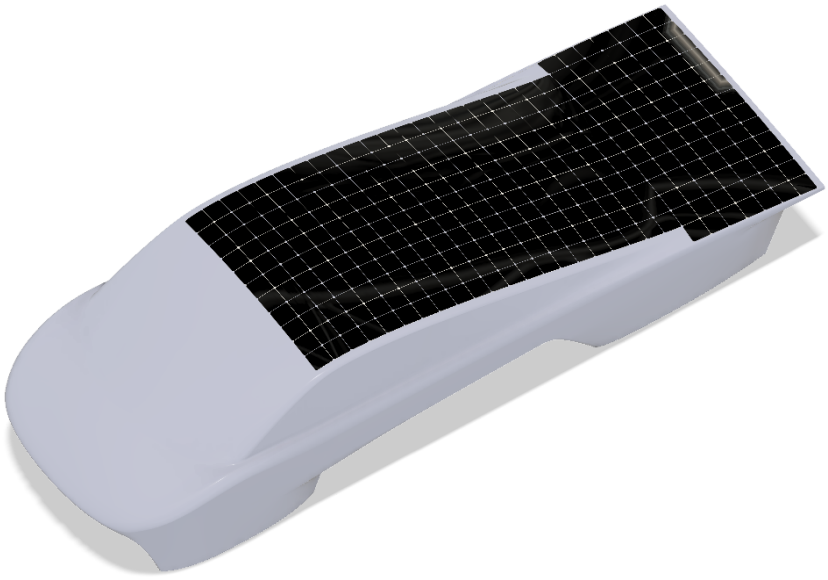


Figure 1: Layout of cells on the ATN solar car.

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 temperature of the photovolataic cells while driving will be about 45°C. At this temperature, each cell will operate at a nominal voltage of 0.605 V. The current that can be generated by a cell depends on the solar irradiance on the cell; with irradiance of 1000 Wm^{-2} the current generated will be 6.138 A. 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 cells with low irradiance.

For ease of manufacturing, the cells will be organised into 29 modules (Figure 2), with the cells in each module connected in series.

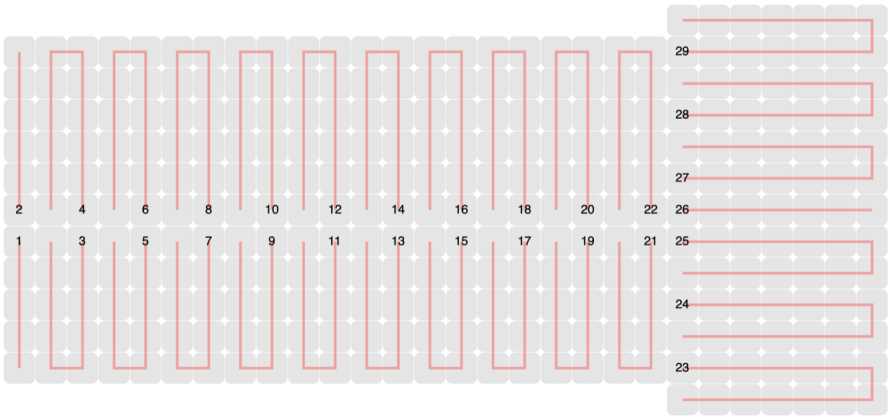


Figure 2: Layout of modules on the ATN solar car.

Each module can be considered as operating with a fixed voltage and, at any instant in time, a fixed current. The power generated by a module is the product of the module voltage and the module current. To get a usable voltage from the solar collector, the modules must be connected in series. But we don't want modules with low irradiance to restrict the overall current. Instead of connecting the modules in series directly, small groups of modules are connected in series and each group is connected to a 'power optimiser' that converts the input voltage V_{in} and input current I_{in} to a common output current I_{out} and output voltage satisfying $V_{\mathsf{out}}I_{\mathsf{out}} = V_{\mathsf{in}}I_{\mathsf{in}}$. The outputs of the power optimisers are then connected in series, summing the output voltages.

Figure 3 shows a simple example. The inputs to the power optimisers are:

- PO1 input: $V = 5 + 6 = 11, I = \min\{4, 3\} = 3, P = 11 \quad 3 = 33$
- PO2 input: $V = 4 + 4 = 8, I = \min\{5, 6\} = 5, P = 8 \quad 5 = 40.$

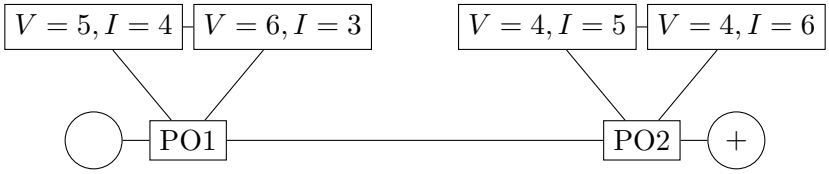


Figure 3: Groups of modules with power optimisers.

The total power generated is $33 + 40 = 73$ watts. If the power optimisers are connected across a 20 V battery then, ignoring losses in the power optimisers, the current through the optimisers will be $73/20 = 3.65$ A, and the outputs from the power optimisers will be

- PO1 output: $V = 9.04, I = 3.65, P = 33$
- PO2 output: $V = 10.96, I = 3.65 = 5, P = 40$.

This arrangement with power optimisers ensures that a low current from one group does not reduce the power from other groups.

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; that is, a power optimiser for each module. However, the power optimisers require a minimum input voltage $V_{\mathsf{min}} = 16$ V. Module voltages vary between 3 V and 9 V, and so each group must comprise two to four modules.

Maximising the energy collected over six days as the car moves from Darwin to Adelaide is a difficult problem. We will start by solving the simpler problem of how to maximise the power generated at one time instant, and then consider the more difficult problem of maximising energy collected.

2 Research overview

The first part of the research focused on developing an integer linear programming model to determine the optimal grouping for one point in time. The objective of the model is to maximise power. The power generated by a group of modules is the product of the minimum module current in the group and the sum of the module voltages in the group.The voltage and the current generated by each module for every kilometre of the route are given.

The second part of the research focused on formulating and solving the problem that maximises the energy collected over the duration of the trip. The first approach to this problem involved clustering modules that have similar orientations to minimising differences in module currents within a group.

We used a hierarchical clustering method that is inbuilt in MATLAB.

Our final results for minimising energy during the trip come from a Cross Entropy Optimisation method that was coded in MATLAB. Cross Entropy Optimisation is a probabilistic search technique that generates and improves candidate solutions. Although the method is not guaranteed to find an optimal solution, we can show that the solutions found are good by comparing them to an intuitive solution and some random solutions.

3 Problem formulations

3.1 Maximising power at a given time instant

The first problem formulated is that of finding the optimal grouping for one instant in time, with a specified number of groups.

Suppose we have n photovoltaic modules. Module j generates current I_j, and has voltage V_j. We wish to partition the modules into m groups, where $j \in G_k$ if module j is in group G_k. The current generated by group G_k will be

$$\hat{I}_k = \min_{j \in G_k} I_j.$$

The voltage generated by group G_k will be

$$\hat{V}_k = \sum_{j \in G_k} V_j.$$

The power generated by group G_k will be

$$\hat{P}_k = \hat{V}_k \hat{I}_k.$$

We want to maximise the total power

$$P = \sum_{k=1\ldots m} \hat{P}_k$$

subject to the constraints

$$\hat{V}_k \quad V_{\min}, \quad \forall\, k \in \{1, \ldots, m\}$$

where $V_{\min}$ is the minimum voltage required in a group.

Modules can be assigned to groups by defining a binary variable

$$x_{jk} = \begin{cases} 1 & \text{if module } j \text{ is assigned to group } k \\ 0 & \text{otherwise} \end{cases}$$

with constraints to ensure that each module is assigned to exactly one group, and that each group has at least one module in it. The problem can be solved for various values of m to determine the optimum number of groups.

3.2 A more efficient formulation

A problem with the first formulation is that groups of modules are given specific group numbers. There are $m!$ ways that groups can be numbered. We can overcome this inefficiency with a modified formulation that does not number the groups.

Suppose we have n photovoltaic modules. Module j generates current I_j, and has voltage V_j. We wish to partition the modules into groups. We introduce a binary decision variable

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

At a given time instant, we want to maximise the total power

$$P = \sum_i V_i \delta_i$$

where the variable δ_i indicates the minimum current of the group to which module i belongs. The value of δ_i is imposed by the constraints

$$\delta_i \quad I_j, \quad 8\ i, j\ 2\ f1, \ldots, ng \text{ such that } x_{ij} = 1.$$

We require the voltage in each group to be greater than the minimum voltage:

$$\sum_i V_i x_{ij} \quad V_{\min}, \quad 8\ j\ 2\ f1, \ldots, ng.$$

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

$$x_{jj} = 1, \quad 8\ j\ 2\ f1, \ldots, ng.$$

If module i is in a group with module j, then module j is in a group with module i. That is, the matrix is symmetric:

$$x_{ij} = x_{ji}, \quad 8\ i, j\ 2\ f1, \ldots, ng.$$

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

$$x_{ij} + x_{jk} \quad x_{ik} + 1, \quad 8\ i, j, k\ 2\ f1, \ldots, ng.$$

3.3 Multiple time periods

The current generated by each module will vary as the car moves along the route and the sun moves across the sky. We have estimates of each module current for each of the 3020 kilometres 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. The energy collected while driving will be

$$E = \sum_k P_k \Delta t.$$

This formulation ignores energy collected while the car is stationary.

As the ultimate objective is to find the best module grouping for the whole journey we introduce a new formulation.

3.4 A formulation with multiple time periods

Suppose we have n photovoltaic modules. Module j generates current $I_{j,k}$ when the car is at location k, and has voltage V_j. We wish to partition the modules into groups. We introduce a binary decision variable

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

We require the voltage in each group to be greater than the minimum voltage:

$$\sum_i V_i x_{ij} \quad V_{\min}, \quad 8\ j\ 2\ f1,\ldots,ng.$$

As before, additional constraints are required to correctly model groups. Module j is in a group with itself:

$$x_{jj} = 1, \quad 8\ j\ 2\ f1,\ldots,ng.$$

If module i is in a group with module j, then module j is in a group with module i. That is, the matrix is symmetric:

$$x_{ij} = x_{ji}, \quad 8\ i,j\ 2\ f1,\ldots,ng.$$

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:

$$x_{ij} + x_{j\ell} \quad x_{i\ell} + 1, \quad 8\ i,j,\ell\ 2\ f1,\ldots,ng.$$

As before, we want to maximise the total energy produced while driving. The power generated at location k is

$$P_k = \sum_{i,k} V_i \delta_{i,k}$$

where the variable $\delta_{i,k}$ indicates the minimum current at location k of the group to which module i belongs. The value of $\delta_{i,k}$ is constrained by

$$\delta_{i,k} \quad I_{j,k}, \quad 8\ i,j\ 2\ f1,\ldots,ng, k\ 2\ f1,\ldots,rg \text{ such that } x_{ij} = 1.$$

The energy generated while driving is

$$E = \Delta t \sum_{k=1\ldots r} P_k.$$

4 Solving the models

We used the MiniZinc constraint modelling language to implement and solve the single instant and overall journey problems.

MiniZinc and its solvers did not work well with floating-point values. The problem was reformulated to use integers by multiplying the currents and voltages by 10^6 and 10^3 respectively. We used the inbuilt OSICBC 2.9/1.16 solver.

4.1 MiniZinc results for the instantaneous problem

When the car is 315 km from Darwin, the sun is almost overhead. The optimal grouping is the eight group layout shown in Figure 4, which generated power $P = 1199.02$ W. The group comprising modules 24 and 25 has the smallest group voltage of 16.94 V.

Figure 4: Optimal solution, with eight groups, when the car is 315 km from Darwin and the sun is overhead.

When the car is 617 km from Darwin the sun is low in the eastern sky. Figure 5 shows the optimal grouping, which has nine groups. The generated power is $P = 696.8$ W. The group comprising modules 1,6 and 22 has the smallest group voltage of 17.55 V

The main difference between the two examples is that the modules on the left and right side of the car are not grouped together when the sun is directed on one side of the car. For both examples, some groups have modules that are not adjacent on the roof of the car. The key, however, is that the optimal grouping is different at different times; we want a solutions with one grouping that is optimal for the entire journey.

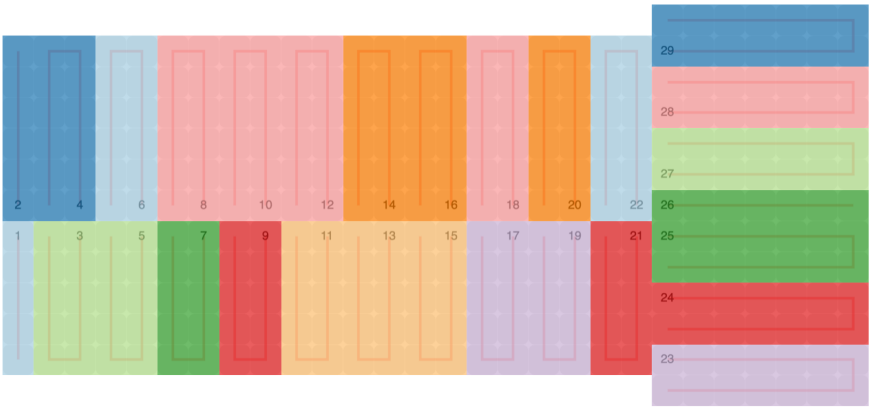


Figure 5: Optimal solution, with nine groups, when the car is 617 km from Darwin and the sun is on one side.

4.2 MiniZinc results for two instances

If we use the formulation for multiple time periods to find a single grouping that maximises the energy collected around $k = 315$ and $k = 617$ then the optimal solution is as shown in Figure 5, with nine groups that generated energy $E = 90\ 755.97$ J. The group containing modules 13,15 and 21 have the smallest group voltage of 18.15 V. Although the results produced are optimal, the computation time for a single location was less than 2.5 minutes, the computation time for two locations was more than 30 minutes, and an example with three locations did not complete within two days. As we want to calculate the optimal grouping for 3022 kilometres, this problem becomes too big for the MiniZinc constraint programming system and probably too big for other solvers as well.

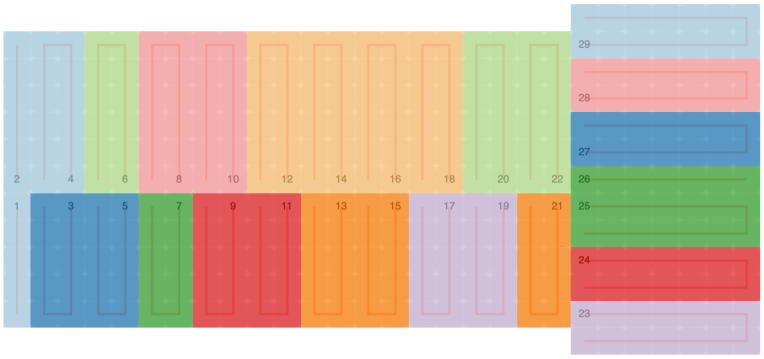


Figure 6: Optimal solution, with nine groups, for when the car is at 315 km and 617 km from Darwin.

5 Clustering modules

The integer program solver took a lot of computational time so another approach was to consider clustering modules based on their normals.

5.1 Calculating module normals

First we estimate the module normal by calculating cell normals at the corners of the modules. Cell normals were estimated by constructing a normal to the solar collector surface then measuring four angles:

α_0: the angle between the front edge of the module and the yz plane

α_1: the angle between the rear edge of the module and the yz plane

β_0: the angle between the left edge of the module and the xz plane

β_1: the angle between the right edge of the module and the xz plane

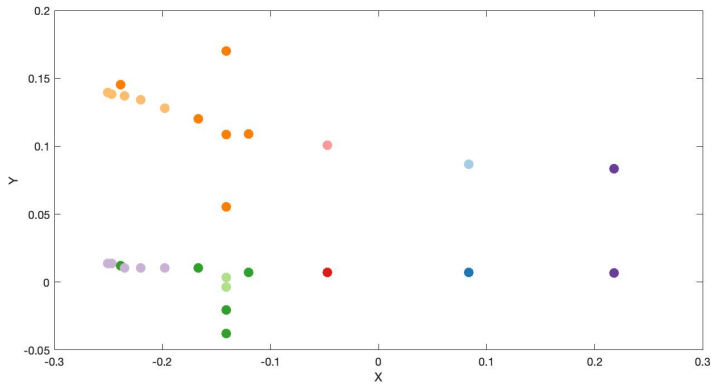
where the x axis runs from the front of the car to the back, the y axis runs from the left of the car to right side, and the z axis runs from the ground to the sky. We calculate the average angles $\alpha = \frac{\alpha_0+\alpha_1}{2}$ and $\beta = \frac{\beta_0+\beta_1}{2}$ and then form clusters that minimise the distance between module angles.

5.2 Ward clustering

Ward clustering is a hierarchical and agglomerate clustering method, (Ward, 1963). Starting with each module in its own cluster, the total number of clusters is gradually reduced by combining neighbouring clusters, until all modules are in one cluster. This creates a hierarchy of clusters; from this hierarchy we may choose the level at which there is a desirable number of clusters.

Ward is also known as a minimum variance algorithm, (Ward, 1963). The goal is to minimise the total variance inside all of the clusters, defined as the sum of squares of differences from the cluster centroid. It therefore chooses to combine two clusters so that the contribution to the variance is minimised.

MATLAB has an inbuilt function that uses the Ward method to cluster the module normals. The objective is to minimise the distance between module normals. The results are shown in Figure 7 for 10 clusters. Although this approach may have had potential, it does not satisfy the minimum voltage constraint.



(a) Clusters of module normal coordinates

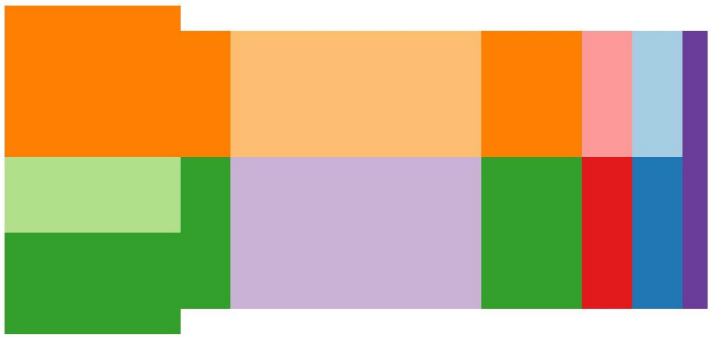


Figure 7: Cluster solution with 10 groups

6 Cross Entropy Optimisation

Cross Entropy Optimisation is a probabilistic search technique for finding good candidate solutions to discrete optimisation problems,(Pudney, 2013). This method, developed by Rubinstein et al in 1997, uses a probability distribution P to generate candidate solutions, then uses the best candidate solutions to update P so that future candidate solutions are better. This process minimises the 'distance' between the distribution P and the ideal but unknown distribution Q; The name 'cross entropy' is taken from this distance measure, (Pudney, 2013).

6.1 The algorithm

For our problem, a candidate solution is derived from a permutation of the modules by dividing the permuted modules into groups. The cross entropy method can be divided into three main steps:

1. generate random solution candidates based on some probability distribution P

(a) use P to generate a permutation of modules

(b) generate a grouping from the permutation

(c) evaluate energy

2. refine the distribution of P using the elite candidates
3. repeat until P converges.

6.2 Example

This example illustrates the process of the Cross Entropy Optimisation method for five modules.

Step 1 (a): Generate permutation

Element p_{ij} of the probability matrix P is the probability that module i will be placed in position j of the permutation. Matrix S is an interim matrix used when generating permutations. We start with

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

Suppose that module 1 is randomly placed in the fourth position. We then ensure that module 1 will not be placed in any other position and no other module will take the fourth position by updating S:

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

We continue this process until all modules have been randomly assigned a position and a permutation

is constructed.

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

The sequence of modules from this matrix is $(2, 4, 3, 1, 5)$.

Step 1 (b): Generate grouping from permutation

From the resulting permutation, the modules are then grouped going through the list, in order, and starting a new group when the sum of the voltages in the group exceeds the minimum voltage, $V_{\mathsf{min}} = 10\,\mathrm{V}$. Any partial group left over at the end is added to the last group.

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

Step 1 (c): Evaluate energy

From the resulting grouping the energy produced for the journey from Darwin to Adelaide is stored and used to compare the candidate solutions.

Step 2: Update P using elite candidates

Step 1 is repeated to generate 100 candidate solutions. We then select the two 'elite' candidates that produce the most energy.

$$S_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad S_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The elite candidates are then combined to give a new probability matrix

$$Q = \frac{1}{2}(S_1 + S_2) = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

The new P matrix formed from the old P and from Q using a smoothing parameter $\alpha = 0.7$.

$$P = \alpha P + (1 \quad \alpha)Q$$

$$P = \begin{bmatrix} 0.14 & 0.14 & 0.14 & 0.29 & 0.29 \\ 0.44 & 0.14 & 0.14 & 0.14 & 0.14 \\ 0.14 & 0.29 & 0.29 & 0.14 & 0.14 \\ 0.14 & 0.29 & 0.29 & 0.14 & 0.14 \\ 0.14 & 0.14 & 0.14 & 0.29 & 0.29 \end{bmatrix}$$

Step 3: Repeat until P converges

Steps 1 and 2 are repeated until Q no longer changes, or else until a predetermined number of iterations have elapsed.

6.3 Result

After 60 iterations we were able to get a reasonable 10 group result that only had a 1.27 % energy loss compared to the energy that would have been generated if no grouping was required.

We do not know how close to optimal this solution is. However, 100 random solutions from the first generation had a mean energy loss of 2.5%.

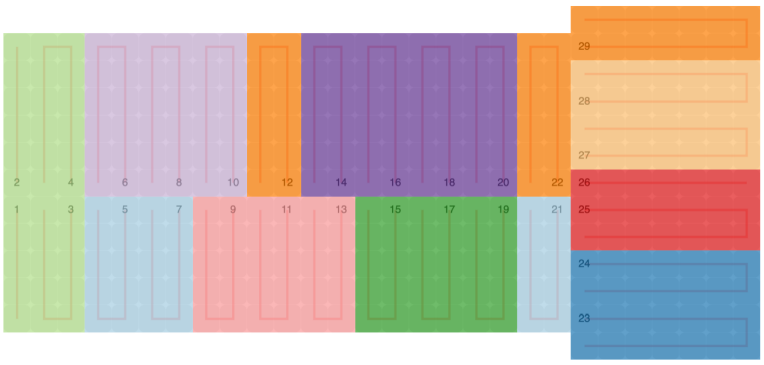


Figure 8: Solution with ten group layout for the race from Darwin to Adelaide.

7 Conclusion

This problem proved to be computationally expensive which meant that we could not prove that our final result was optimal. Further research on the clustering method may ensure that constraints are meant. Although we can not prove optimum for this problem the Cross Entropy Optimisation algorithm was able to partition the 29 modules into 3 groups of 2, 5 groups of 3 and 2 groups of 4. The total energy generated during the trip is within 1.27% of the energy that would be generated if each module had a separate power optimiser.

8 References

Pudney, P 2013, 'Cross Entropy Optimisation', pp.1,11.

Ward, J.H 1963, 'Hierarchical grouping to optimize an objective function', *Journal of the American Statistical Association*, vol.58, no.301, pp. 236-244.