PARALLEL GLOBAL OPTIMISATION BASED CONTROL OF BOUNDARY LAYER TRANSITION

G.V. Veres*, O.R.Tutty†, E.Rogers‡, P.A. Nelson**

* School of Engineering Sciences, University of Southampton, UK, gvv@soton.ac.uk,
† School of Engineering Sciences, University of Southampton, UK, o.r.tutty@soton.ac.uk,
‡ Department of Electronics and Computer Science, University of Southampton, UK, etar@ecs.soton.ac.uk,
** Institute of Sound and Vibration Research, University of Southampton, UK, pan@isvr.soton.ac.uk

Keywords: Boundary layer transition control, Flow control, Aerofoil, Non-smooth optimisation, Parallel global optimisation.

Abstract

Turbulent flow has a significantly higher drag than the corresponding laminar flow at the same flow conditions. The presence of turbulent flow over a large part of an aircraft therefore incurs a significant penalty in the form of increased fuel consumption due to the extra thrust required. One possible way of decreasing the drag is to apply surface suction to delay the transition from laminar to turbulent flow. In this paper we report further progress on the development of optimisation based control algorithms for this problem area.

1 Introduction

In recent years a number of ways have been considered for drag reduction and hence reduced operating costs for civil aircraft due to a decrease in the fuel consumption (see [5] for a summary of some recent work in this area). It is well established that suction applied to surface of a body can delay the onset of transition from laminar to turbulent flow and thereby reduce the overall drag. Consequently, surface suction is regarded as one of the most promising methods for drag reduction. However, in order to maximise the benefit from the applied suction, it is necessary to distribute the suction in an optimum, or near optimum, way.

One approach to this problem consists, in effect, of monitoring the state of the flow together with the automatic control of suction applied through the surface of the body. This program involves both algorithm development and verification and wind tunnel based experiments. In the case of the latter, an essential element has been the development of techniques for monitoring the position of transition in a boundary layer.

Previous work has demonstrated that this overall strategy is technically feasible both in simulation studies (of which computational fluid dynamics is an essential part) and by wind tunnel based experiments, see, for example, [7]. This work employed gradient based controllers, which used a local linear model constructed by standard linear system identification tools around selected operating points. For the case of flow over a flat plate these controllers performed well. In somewhat more complicated cases, however, they did not, e.g. when a non-zero pressure gradient is applied to a flat plate [8].

In [3] is was established that the gradient based approach failed when it was incompatible with the basic physics of the flow. An alternative approach, based on the application of Simulated Annealing and Genetic Algorithm based methods to solve a non-linear optimisation problem, was also successfully developed, but computationally it is very demanding. Moreover, in the case of an aerofoil, the computational load increases further. Hence there is a need to develop alternative, more efficient, solution methods. This paper details work undertaken on this problem for the model case of flow past an aerofoil where parallel optimisation techniques are developed for computationally demanding cases. There is no claim that the proposed methods are preferable to all other methods. Rather, the choice of optimisation technique will be determined by the demands of the applications considered.

2 Cost function

First, it is necessary to define an appropriate cost function to be minimised. One relatively simple function is obtained by fixing the transition position at a predetermined location, and then minimising the effort by varying the suction flow rates and the positions of the panels to achieve the best result. This approach is of practical interest as it comes directly from the design requirements for a suction control system on a nacelle.

The problem of constraining the transition position to some desired location on an aerofoil with 5 or 10 non-overlapped panels without gaps fixed at the 20% of the chord and with suction flow rates denoted by u_i can be formulated as follows:

$$\text{minimise } \phi_s = k_1 \sum_{i=1}^{N+1} u_i^2 \ : x_t - x_d = 0, \tag{1}$$

where ϕ_s is a suction effort, N is the number of panels, k_1 is a positive constant, u_i is a suction flow rate through panel i, x_t is a transition position and x_d is a desired transition position. This constrained minimisation problem can be recast as solving the unconstrained minimum problem

$$\text{minimise } \phi = \phi_s + k_2 |x_t - x_d|^l, \tag{2}$$

where k_2 is a positive constant, and l is a positive integer to be selected (in this work $l = 1$ is used).

Note that x_t is some function of the sequence $\{u^i\}$ and there is dependency between the panel positions and flow rates through an unknown (or very complicated) function. As a result, the optimisation problem (2) is nonlinear and, as experiments have confirmed, multi-modal. Increasing the number of panels and considering a realistic scenario (i.e. an aerofoil or $3D$ flow), significantly increases the computational burden to the level where parallel processing is required.

3 Random search

Random search methods directly aim to find the global minimum [1]. The random search algorithm considered here consists of the following steps.

- Choose the initial number of points N_1 and wide initial feasible bounds for flow rates. Then select N_1 uniformly distributed points within the initial feasible bounds.
- Obtain the cost function values at the initial set of points concurrently and save a point with minimum cost function value as a near optimal solution.
- Using the cost function values, reduce the feasible bounds to areas more likely to contain optimal points and choose the number of points N_2 to be selected from within the new bounds. Then select these as uniformly distributed points and calculate the cost function values at them concurrently.
- Choose the point with the minimum cost function value from the new points and old near optimal solution as a new near optimal solution.

This approach can help in narrowing the bounds on the suction flow rates, but will require a ‘sufficient large’ number of points to find an optimal or near optimal solution.

4 Parallel global optimisation method

Since the cost function to be optimised is non-smooth, non-convex and multi-modal, a modification of a direct search method is proposed to find the global minimum. Direct search methods, for example the asynchronous parallel pattern search [2], have been successfully implemented on parallel computers. The proposed algorithm here is one of the possible modifications of the deformed configuration methods [4], which belong to direct search method family. In this family, the basic idea of constructing simplex, $n+1$ vertices in n-dimensional space, and complex, $\geq n+2$ vertices in n-dimensional space, optimisation procedures is generalised by the introduction of search control, i.e. the choice of the locally optimal direction, mapping of configuration vertices and centroid, and choice of the step-size, leading reduction of the cost function values. In this work $n = N+1$.

Due to the mapping of several configuration (simplex or complex) vertices whose number is automatically adjusted from step-to-step, these methods exhibit relatively fast convergence and are less sensitive to the noise in the measurements of the objective function. The major differences between the proposed algorithm and both the asynchronous parallel pattern search [2] and the deformed configuration methods [4] are the additional preliminary procedure and the sophisticated choice of mapping directions to ensure that the global optimum of multi-modal cost function is found. All vertices are divided into three groups, i.e. mapped, reflected, and the best one when the vertices of the complex and the complex centre are compared in terms of the minimum cost function value. Mapped vertices are vertices whose cost function values are greater than cost function value in the complex centre. Reflected vertices are the vertices whose cost function values are less than or equal to the cost function values of the complex centre but greater than the cost function value at the best complex vertex, i.e. the complex vertex with the minimum cost function value in the complex.

After division of vertices, the construction of a new complex is carried out according the formulas presented below. In the proposed algorithm all complex vertices take part in the exploration of possible directions. This gives better coverage of the search space in comparison to the original algorithm, which is important when there are several local optimums as in the current application area. The final algorithm consists of three stages: initialisation, exploratory moves and modified method of deformed configurations as detailed next, where the cost function is now denoted by $F(\cdot)$.

Initialisation

1. Define feasible bounds for suction flow rates.
2. Define step-size parameters α, β, γ.
3. Select stopping tolerances tol_1 and tol_2, the maximum number of iterations N_1, and maximum number of iterations allowed without changes in minimum cost function values N_2.
4. Construct an inscribed complex (a complex in R^n is a set of $c > n+1$ vertices, where n now denotes the dimension of optimisation problem) $\mathbf{u}^{(0,i)}$, $i = 1, \ldots, c$, within the specified bounds, and compute the geometrical centre of the complex $\mathbf{u}^{(k)}$. Then evaluate the cost function values at the vertices of complex $F(\mathbf{u}^{(0,i)})$ and centre $F(\mathbf{u}^{(k)})$, concurrently.

Exploratory moves

a) Divide a line connecting each vertex of the complex and the complex centre into ten intervals.

b) Calculate cost function values at new points concurrently.

c) Take the point with minimum cost function value as a new complex point in the direction of the old complex point and centre.

Modified method of deformed configurations

1. If $k \leq N_1$, then $k = k + 1$ (k is a number of iterations), go to Step 2. Else, exit.
2. Perform correct enumeration of complex vertices, i.e. such that the following chain of inequalities holds

   $$F(\mathbf{u}^{(k,1)}) \geq F(\mathbf{u}^{(k,2)}) \geq \ldots \geq F(\mathbf{u}^{(k,c)}).$$

3. If

   $$(F(\mathbf{u}^{(k,1)}) - F(\mathbf{u}^{(k,c)}) < \varepsilon_1 \text{ and}$$
   $$\max_{2<i<c} \|\mathbf{u}^{(k,1)} - \mathbf{u}^{(k,i)}\| < \varepsilon_2)$$

   or the number of iterations without changing the minimum cost function value exceeds N_2, exit. Else, go to Step 4.
4. Divide all vertices into three groups: m mapped vertices with $F(\mathbf{u}^{(k,i)}) > F(\mathbf{u}^{(k)})$, $i = 1, \ldots, m$; l reflected vertices with $F(\mathbf{u}^{(k,i)}) \leq F(\mathbf{u}^{(k)})$ and $F(\mathbf{u}^{(k,i)}) > F(\mathbf{u}^{(k,c)})$, $i = 1, \ldots, l$; l is the vertex with minimum cost function value between the best complex vertex and the complex centre.
5. Map m vertices through the centre of the unmapped vertices

   $$\mathbf{s}^{(k,i)}(m) = \frac{1}{c-m} \sum_{j=m+1}^{c} \mathbf{u}^{(k,j)}$$

   according to

   $$\mathbf{u}^{(kr,i)} = \mathbf{s}^{(k,i)}(m) + \gamma(\mathbf{s}^{(k,i)}(m) - \mathbf{u}^{(k,i)}).$$

6. Reflect l vertices through the best complex vertex with the minimum cost function value according to

   $$\mathbf{u}^{(kr,i)} = \mathbf{u}^{(k,c)} + \gamma(\mathbf{u}^{(k,c)} - \mathbf{u}^{(k,i)}).$$

7. If $F(\mathbf{u}^{(k,c)}) > F(\mathbf{u}^{(k)})$, then reflect the best complex vertex through the complex centre:

   $$\mathbf{u}^{(kr,c)} = \mathbf{u}^{(k)} + \gamma(\mathbf{u}^{(k)} - \mathbf{u}^{(k,c)}).$$

   Else, reflect the complex centre through the best complex vertex

   $$\mathbf{u}^{(kr,c)} = \mathbf{u}^{(k,c)} + \gamma(\mathbf{u}^{(k,c)} - \mathbf{u}^{(k)}).$$

8. Evaluate $F(\mathbf{u}^{(kr,i)})$, $i = 1, \ldots, c$, concurrently.
9. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k)})$, $i = 1, \ldots, m$, then repeat mapping with a bigger step-size parameter α. Else, repeat mapping with a smaller step-size parameter β.
10. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k,c)})$, $i = 1, \ldots, l$, then repeat reflection with a bigger step-size parameter α. Else, repeat reflection with smaller step-size parameter β.
11. If a reflected vertex in group 1 has smaller cost function value than a non-reflected vertex, then repeat reflection with a bigger step-size parameter α. Else, repeat reflection with a smaller step-size parameter β.
12. Calculate the cost function values in new c vertices, concurrently.
13. Choose c vertices from the old vertices and reflect with different step-size parameters the ones with minimum cost function value.

14 Calculate a new geometrical centre of the new complex and the cost function value at it.

The vertex with minimum value of $F(\mathbf{u})$ is assumed as the estimation of the minimum point.

As it can be seen from algorithm above, parallisation occurs only at the stage of cost function value calculation. Further parallisation is possible at the stage of calculating new complex vertices. The proposed algorithm can find the global optimum quite accurately [9], however the complexity of optimisation problem will grow exponentially with dimension and in the case of 10 panels it requires weeks of computational time even on parallel processors to achieve the global minimum or a reasonable solution. A multi-start parallel global optimisation algorithm is proposed below for higher dimensions, where complexity of optimisation problem will grow linearly rather than exponentially.

5 Multi-start parallel global optimisation

To speed up the optimisation process when 10 or more panels are considered, it is proposed here to use the multi-start parallel global optimisation algorithm with pattern search approach [6]. The main idea in this approach is to first find the points from where the local search starts and then to apply the local search to selected points. The algorithm consists of three stages: initialisation, multi-start points selection, and local search, as detailed next.

Initialisation

1. Define feasible bounds for linear distributed flow rates, and the step-size parameters α, β, and γ.
2. Select the maximum number of iterations N_1 and the maximum number of iterations allowed without changes in minimum cost function value N_2.
3. Choose a number of complex vertices c for local search.
4. Select N uniformly distributed random points $\mathbf{u}_i$, $i = 1, ..., N$, within the feasible bounds to form the set of initial points S_0.
5. Evaluate the cost function values $F(\mathbf{u}_i)$ concurrently.

Multi-start points selection

a) Sort all points in ascending order with respect to cost function values, i.e. $\mathbf{u}_1$ and $\mathbf{u}_N$ will be the points with minimum and maximum cost function values respectively .

b) Normalise all cost function values according to

$$F_n(\mathbf{u}_i) = \frac{F(\mathbf{u}_i) - F(\mathbf{u}_1)}{F(\mathbf{u}_N) - F(\mathbf{u}_1)},\ i = 1, \ldots, N$$

c) Remove the points from the set S_0 with $F_n(\mathbf{u}_i) > tol_1$, where tol_1 is some threshold.

d) Do the following until there are no points left in set S_0.

e) Add the first point to the set of multi-start points S

f) Calculate the Euclidean distances from this point to all points, normalized these distances, and remove from the set S_0 points with distances less or equal to the chosen threshold tol_2 . Go to d).

Local search

1. Let $\mathbf{u}_{min}$ be a point with the minimum cost function value $F(\mathbf{u}_{min})$ from the set S, and choose the number of complex vertices c.
3. For each multi-start point from the set S do the following.
4. Construct an initial complex
5. $min \leftarrow arg_i \min_{0 \leq i \leq c-1} F(\mathbf{u}^{(0,i)})$, and swap $\mathbf{u}^{(0,min)}$ and $\mathbf{u}^{(0,0)}$
7. If $k_1 \leq N_1$ and $k_2 \leq N_2$, then $k_1 = k_1 + 1$ (k_1 is an iteration counter, k_2 is the number of non-changes in minimum cost function value), go to Step 8. Else, go to Step 14..
8. Reflect $c - 1$ vertices through the best vertex $\mathbf{u}^{(k,min)}$ according to

$$\mathbf{u}^{(kr,i)} = \mathbf{u}^{(k,min)} + \gamma(\mathbf{u}^{(k,min)} - \mathbf{u}^{(k,i)}),$$

$i = 1, \ldots, c - 1$

9. Evaluate $F(\mathbf{u}^{(kr,i)})$, $i = 1, \ldots, c - 1$, concurrently.
10. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k,min)})$, then repeat the mapping with bigger step-size parameter α. Else, repeat the mapping with smaller step-size parameter β.
11. Calculate the cost function values at the new $c - 1$ vertices concurrently.
12. Choose $c - 1$ vertices from old vertices and reflect with different step-size parameters the ones with minimum cost function value.
13. If there is no change in the minimum cost function value $k_2 = k_2 + 1$, else $k_2 = 0$
14. The vertex $\mathbf{u}^{k,min}$ with minimum value of $F(\mathbf{u})$ is assumed as the estimation of the minimum point.
15. If $F(\mathbf{u}^{k,min}) < F(\mathbf{u}_{min})$, replace $\mathbf{u}_{min}$ by $\mathbf{u}^{k,min}$, and $F(\mathbf{u}_{min})$ by $F(\mathbf{u}^{k,min})$. Go to Step 3.

This algorithm is faster than the parallel global optimsation algorithm based on the deformed configuration method, but its accuracy will be depend on the thresholds at the multi-start points selection stage and on the number of vertices in the initial complex for each multi-start point. To obtain a compromise between accuracy and speed of global optimisation, a combination of a random search procedure and multi-start parallel global optimisation is proposed, as detailed next.

If the most important criterion is accuracy, then it is possible to combine the random search procedure with multi-start parallel global optimisation. The idea is simple; wide initial bounds for flow rates are defined. Then the random search is used to narrow these bounds to regions more likely to contain the global optimum. After this, the multi-start parallel global optimisation is used within the new bounds. The steps in the proposed algorithm are as follows.

- Choose the initial number of points N_1 and wide initial feasible bounds for flow rates. Select N_1 uniformly distributed points within the initial feasible bounds
- Obtain the cost function values for the initial set of points concurrently
- Taking into consideration cost function values, reduce the feasible bounds to areas more likely to contain optimal points.
- Select N uniformly distributed random points $\mathbf{u}_i$, $i = 1, ..., N$, within the feasible bounds. These points will be form the set of initial points S_0.
- Evaluate the cost function values $F(\mathbf{u}_i)$ concurrently.
- Perform multi-start point selection and the local search stages of the multi-start parallel global optimisation algorithm subject to the initial bounds on the flow rates.

6 Results

In this section different approaches are compared for 5 and 10 non-overlapping panels fixed at 20% of the chord with linear continuous suction flow rate distributions for $2D$ flow. The three approaches proposed above are compared with the random search approach. For both numbers of panels, the parameters α, β and γ were chosen as 1.9, 0.29 and 1.1 respectively, and thresholds for the multi-start point selection were set at $tol_1 = 0.15$ and $tol_2 = 0.5$. Initial feasible sets for flow rates were specified as in the range $[-3.7e - 4, 0]$.

6.1 5 panels case

Random search (RS), parallel global optimisation based on modified deformed configuration method (PGOMDC), multi-start parallel global optimisation with pattern search for local search (MPGOPS) and a combination of random search and multi-start parallel global optimisation (CRSMPO) are compared here. For RS, $N_1 = 8190$ and $N_2 = 3567$ and S_0 contained 2900 initial points, S_1 contained 10 multi-start points for MPGOPS. For CRSMPO, $N_1 = 3567$, S_0 contained 2900 points and S_1 contained 13 multi-start points. Tables 1 and 2 give the results.

As can be seen from the Table 2, PGOMDC is the most accurate approach, but it requires almost twice as much time as the RS, MPGOPS and CRSMPO methods to achieve its solution. Moreover, the complexity of PGOMDC method, and time required to obtain a solution, grows exponentially with increasing problem dimension.

The MPGOPS method appears as a good compromise between accuracy and computational load. Note that the time given in Table 2 is pure calculation time and does not take into consideration waiting time. Here the CRSMPO method is worse both in time and in accuracy in comparison with MPGOPS. However, the performances of two algorithms MPGOPS and CRSMPO depend on turning parameters, and the type and dimension of the optimisation problem being solved. It is expected that the CRSMPO method will out-perform the MPOPS approach at least in accuracy terms.

6.2 10 panels case

The RS, MPGOPS, and a combination of random search and multi-start parallel global optimisation (CRSMPO) are compared for a case with 10 panels, where parallel optimisation based on the modified deformed configuration method (PGOMDC) is too slow. For RS, $N_1 = 8190$, $N_2 = 8961$, S_0 contains 9984 initial points, and S_1 contains 13 multi-start points in MPGOPS. For CRSMPO $N_1 = 8190$, S_0 contains 9984 points, and S_1 contains 20 multi-start points. The results of comparing the two approaches are given in Tables 3, 4, 5. It can be seen from Table 5 that the CRSMPO method is the most accurate. However, it requires the most computational time in comparison with other two methods, i.e. the preferences of users or/and the type of the application problems will decide which one of the proposed approaches is chosen for a given global optimisation problem.

7 Conclusions

Further progress on the development of algorithms for the control of boundary layer transition has been reported in this paper. In case of an aerofoil and linear continuous suction distributions over the front part of the aerofoil the computational load increases significantly, and in some cases it is not possible to continue the investigation using a single processor. Therefore several parallel global optimisation algorithms have proposed here. For the case of 5 panels, simulation studies showed that a parallel global optimisation method based on modified deformed configuration method (PGOMDC) is the most accurate. This method does, however, require 50% more computational time than the random search method (RS) and multi-start parallel global optimisation method with pattern search for local search (MPGOPS). Moreover, the computational complexity of PGOMDC will increase exponentially with the dimensionality of the problem. Therefore for case of 10 panels we consider only RS, MPGOPS and combination of random search and multi-start parallel global optimisation (CRSMPO). Simulations showed that, as expected, the CRSMPO method is the most accurate of the three approaches, but it requires more computational time to achieve its solution. Therefore there is a trade off between finding the global minimum and the computational load and choice of the global optimisation approach will depend on preferences of user and/or the particular application problem under consideration.

References

[1] R. Horst, P.M. Pardalos (Eds.) *Handbook of Global Optimization*, Kluwer Academic Publichers, Dordrecht/ Boston/ London, 1995.

[2] T.G.Kolda, P.D. Hough and V.J. Torczon. Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, 23:134–156, 2001.

[3] W. MacCormack, O.R. Tutty, E. Rogers and P.A. Nelson. Stochastic optimisation based control of boundary layer transition. *Control Engineering Practice*, 10:243-260, 2002.

[4] A. Rykov. Construction principles of deformed configurations methods. *Preprints of the summer school course on Identification and Optimization oriented for use in adaptive control, Prague, Czech Republic*, pages 65–80, 1995.

[5] P. Thiede (Ed.). *Aerodynamic Drag Reduction Technologies: Proceedings of the CEAS/DragNet European Drag Reduction Conference, June 2000, Potsdam, Germany*, Springer, 2001.

[6] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991.

[7] Tutty, O.R., Hackenberg, P. and Nelson, P.A. Numerical optimisation of the suction distribution for laminar flow control. *AIAA J.*:38, 370-372, 2000.

[8] Tutty, O.R., Hackenberg, P. and Nelson, P.A. Gradient projection based control and optimisation of the suction distribution for laminar flow control. *Proc. I. Mech. E. Part I: Systems and Control Engineering*:214, 347-359, 2000.

[9] G.V. Veres, O.R. Tutty, E. Rogers and P. A. Nelson. On design optimisation based control methods for distributed suctions for boundary layer transition control *Proc. American Control Conference*, Denver, June 2003.

Method	u_1	u_2	u_3	u_4	u_5	u_6
RS	0	$-6.66254e-5$	$-3.4446661e-4$	$-3.1349e-04$	$-8.161222e-5$	$-2.872856e-5$
PGOMDC	0	$-1.5e-5$	$-1.1e-5$	$-9e-5$	$-3.6999e-4$	$-1.9682e-4$
MPGOPS	0	0	$-4.32388e-5$	0	$-2.67273e-4$	$-1.98027e-4$
CRSMPO	0	$-1.580755e-4$	$-4.168359e-5$	$9.0e-5$	$-3.442298e-4$	$-3.065454e-4$

Table 1: The near optimal flow rates for 5 panels

Method	x_t	ϕ_s	ϕ	Time
RS	0.79783	$2.288502e-3$	$4.458831e-3$	19.15 hours
PGOMDC	0.8	$1.453423e-3$	$1.453421e-3$	30.73 hours
MPGOPS	0.80003	$1.777773e-3$	$2.061330e-3$	19.47 hours
CRSMPO	$0.0.7999722$	$2.472896e-3$	$2.750586e-3$	19.81 hours

Table 2: The minimum overall cost functions obtained by optimisation approaches for 5 panels

Method	u_1	u_2	u_3	u_4	u_5	u_6
RS	0	$-4.517587e-5$	$-6.48977e-5$	$-6.626079e-05$	$-4.143352e-5$	$-8.269108e-5$
MPGOPS	0	$-4.517458e-4$	$-8.147645e-5$	$-2.293512e-4$	$-1.702706e-4$	$-6.660711e-5$
CRSMPO	0	$-1.136449e-4$	0	0	$-3.991312e-5$	0

Table 3: The near optimal flow rates for 10 panels

Method	u_7	u_8	u_9	u_{10}	u_{11}
RS	$-4.846197e-5$	$-1.02092e-4$	$-1.354573e-4$	$-3.570386e-04$	$-2.767380e-4$
MPGOPS	$-3.352883e-4$	$-3.423307e-4$	0	$-7.614603e-5$	0
CRSMPO	$-3.623112e-4$	$-2.624833ee-4$	0	0	0

Table 4: The near optimal flow rates for 10 panels

Method	x_t	ϕ_s	ϕ	Time
RS	0.8000698	$2.543782e-3$	$3.241703e-3$	16.80 hours
MPGOPS	0.7999865	$3.301165e-3$	$3.436260e-3$	18.20 hours
CRSMPO	0.8000892	$2.146751e-3$	$3.038975e-3$	31.53 hours

Table 5: The minimum overall cost functions obtained by optimisation approaches for 10 panels