# GEOMETRIC OPTIMISATION OF YIELD-LINE PATTERNS USING A DIRECT SEARCH METHOD

A.C.A. RAMSAY & D. JOHNSON

Department of Civil & Structural Engineering, The Nottingham Trent University,

Burton Street, Nottingham NG1 4BU, U.K.

[Submitted to the International Journal of Structural Optimization, December 1996]

**Keywords: automated yield-line analysis, geometric optimisation, limit analysis of slabs.**

**Abstract:** A class of problems in the geometric optimisation of yield-line patterns, for which the currently advocated *conjugate gradient* and *sequential linear programming* geometric optimisation algorithms fail is investigated. The *Hooke-Jeeves* direct search method is implemented and is demonstrated to solve such problems robustly.

## INTRODUCTION

The yield-line technique[1] is well established for the ultimate analysis of slabs. The technique requires the postulation of a kinematically admissible yield-line or fracture pattern from which the corresponding collapse load is determined through the principle of virtual displacements. By the upper-bound theorem of plasticity it can be shown that for fracture patterns other than the critical (true) one, the collapse load is greater than, or equal to the critical (true) collapse load. The approach is thus inherently unsafe. To obtain a good estimate of the critical collapse load, great care needs to be exercised to ensure that the postulated fracture pattern is sufficiently close to the critical fracture pattern. For what might be considered as 'standard' configurations of slabs, there exists a large body of experience that can be called upon to assist in the postulation of fracture patterns. Whilst, for such slabs, it is possible to predict the correct mode of fracture pattern, it is generally less easy to predict the correct geometry of this mode. The distinction between the correct mode of failure and the correct geometry of the mode is clarified with the cantilever slab example shown in Fig. 1.

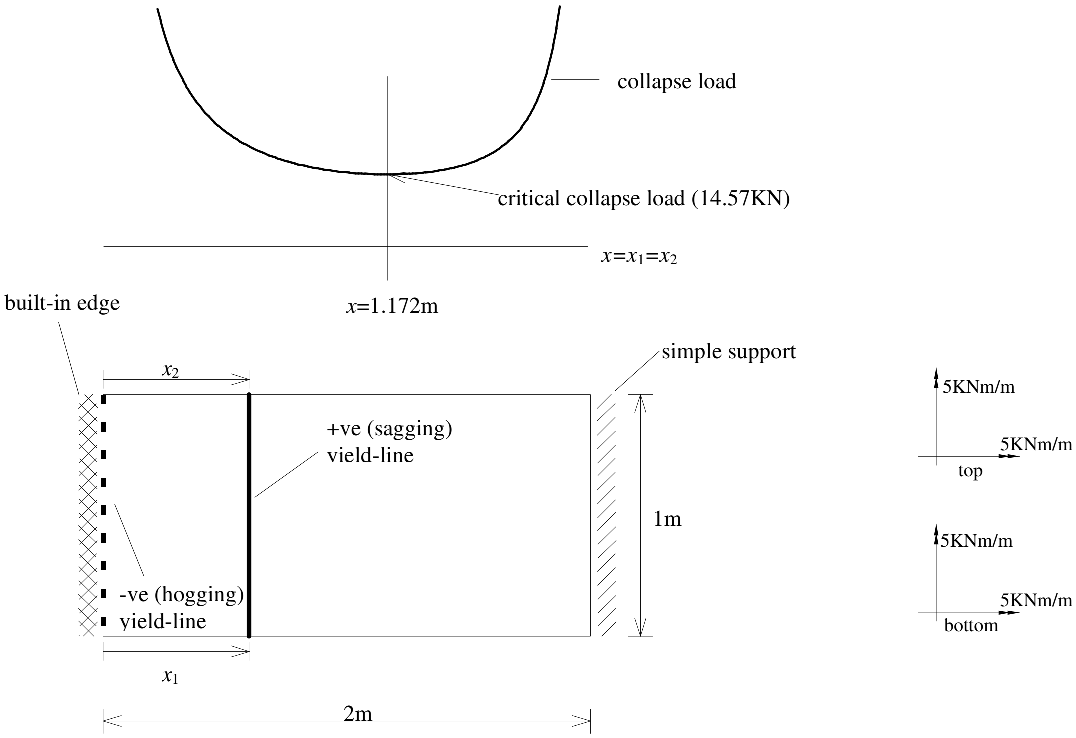



**Figure 1: C**antilever slab

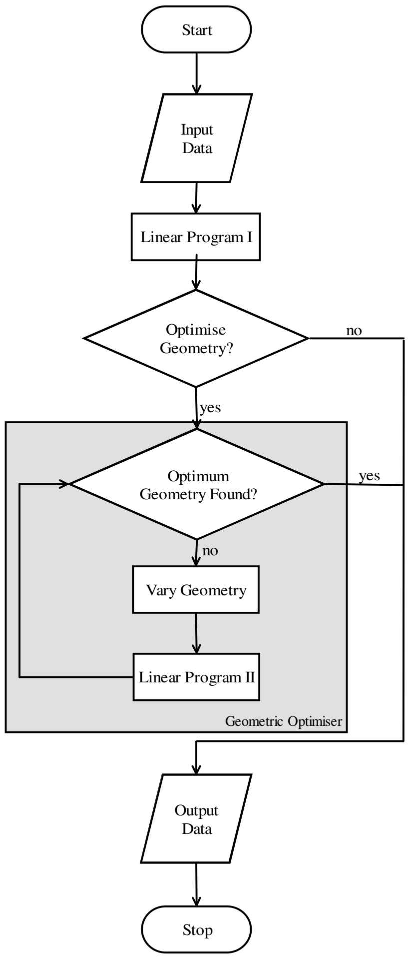
The slab, which is built-in at the left-hand edge and simply supported at the right-hand edge, has a total load *P* uniformly distributed over the entire area. Both top and bottom reinforcement are assumed to be uniform and isotropic as indicated in the figure.

The true mode of failure for this problem is well known and is indicated in Fig. 1. For simple fracture patterns such as this one, with few geometric variables, classical optimisation methods can be used to determine the optimum geometry of the chosen fracture pattern[2]. In this case the optimum geometry is $x = x_1 = x_2 = 1.172$m and the corresponding critical collapse load is $P = 14.57$KN. The way in which the collapse load varies with the coordinate $x$ is also shown in Fig. 1.

For 'non-standard' configurations, it is essential to investigate a range of possible modes of fracture pattern and the way in which the collapse load varies with the geometry of each mode. With traditional hand-calculation methods, this process is time consuming and exhaustive searches allowing for all possible configurations of failure mode and geometry are generally not possible. There is thus a distinct possibility that either the correct mode will be missed[3,4] or the optimum geometry of the correct mode will not be found[5]. The cases cited here are important reminders of the need for caution when using the yield-line technique.

One of the main drawbacks of the yield-line technique is that it is essentially a hand calculation method and the incentive to investigate large numbers of potential failure patterns and/or geometries is thus low. The computer-based, automated yield-line methods of analysis[6,7] overcome this problem by considering more than one fracture pattern simultaneously. In this method a mesh of rigid triangular elements, for which the interfaces between elements and any moment-resisting boundary edges are considered as potential yield-lines, is used to define a set of possible fracture patterns. Linear programming is used to select the fracture pattern with the lowest collapse load.

Whilst automated yield-line analysis can often determine the correct mode of fracture pattern, the method is generally unable to predict the optimum geometry of the correct mode. This is because the element edges in a particular mesh discretisation are unlikely, in general, to coincide exactly with the yield-lines of the true solution. As such, some sort of geometric optimisation is usually carried out after automated yield-line analysis has identified the critical fracture pattern. The reduction in predicted collapse load that occurs with geometric optimisation can be very significant indeed and the importance of carrying out some sort of geometric optimisation of the fracture pattern can not be over-emphasised. In reference [8], for example, a case is cited where geometric optimisation leads to a 30% reduction in the prediction of the critical collapse load. A flow chart showing the essential elements of automated yield-line analysis and geometric optimisation is given in Fig. 2.




(i) Input consists of mesh topology, boundary conditions, applied loads and material properties.

(ii) Automated Yield-Line Analysis is obtained by answering ‘no’ to the question ‘optimise geometry?’

(iii) The essential elements of the geometric optimiser are shown. The exact nature of these elements will vary with the different algorithms and implementations that are used. Linear Program II may not be the same as Linear Program I c.f. reference [10] where Linear Program II includes geometric sensitivities.

**Figure 2:** Flowchart for automated yield-line analysis and geometric optimisation

For the geometric optimisation of fracture patterns, a simplified mesh is generally used and the positions of the nodes within the mesh and on the boundaries of the model are taken as variables. The optimum nodal positions are those that minimise the collapse load for the model. The problem of geometric optimisation has been investigated by a number of workers. Jennings et al.[9], for example, perform successive line minimisations along conjugate directions using the Fletcher-Reeves implementation of the conjugate gradient method. Johnson[10], on the other hand, uses the technique of sequential linear programming in which linearised geometric sensitivities are developed analytically.

Whilst both methods have been demonstrated to perform well for a range of different problems[8,9,10,11], there are a number of surprisingly simple configurations for which both methods fail. The problems for which these methods fail tend to be those where the chosen geometric variables are such that a number of different modes of fracture pattern can occur. For example, consider again the cantilever slab of Fig. 1 discretised with four triangular elements. If, instead of the single geometric variable previously considered, the *x*-coordinate of both ends of the positive yield-line had been chosen as variables, then three possible modes of fracture pattern exist as shown in Fig. 3.

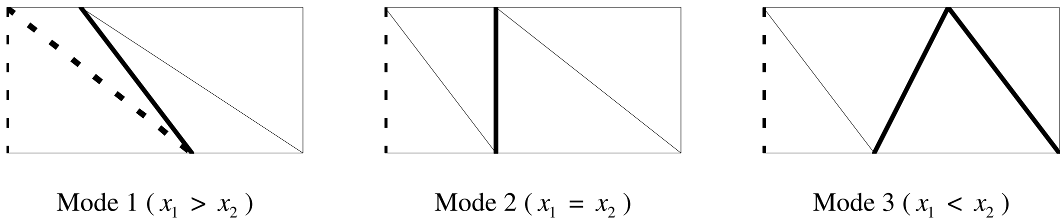



**Figure 3:** Modes of fracture pattern for cantilever slab with two geometric variables

With the initial position $x_1 = 0.2$m and $x_2 = 0.7$m, the sequential linear programming routine of reference [10] converges to the solution $x = x_1 = x_2 = 0.875$m and, although the critical mode of fracture has been correctly predicted, the method has failed to find the optimum geometry. The reason that difficulties occur for such problems seems, as suggested[9,11], to be related to the fact that the gradient of the collapse load is discontinuous at the interface of different modes of fracture pattern. However, this is not the complete answer to the question because, despite the existence of such discontinuities, for other problems, which also involve a multiplicity of fracture modes, the critical fracture mode and its optimum geometry can be found.

The main body of this paper is divided into two sections. In the first section, two examples, one which can and one which cannot be solved by currently advocated algorithms, will be considered. By comparing these two examples, it is possible to identify certain characteristics of a problem that may cause currently advocated algorithms to fail. In the second section, results from the implementation of a different optimising algorithm that is able to solve both examples robustly are presented.

## THE NATURE OF THE OBJECTIVE FUNCTION AND CONSTRAINTS

In the geometric optimisation of fracture patterns, the objective function is the collapse load and the variables are the positions of the nodes. Optimisation problems are classified as constrained or unconstrained depending on whether or not bounds are placed on the variables. In the case of geometric optimisation, bounds are imposed on the variables due to the physical considerations that the positions of the nodes should not violate the geometry of the slab and/or the topological integrity of the chosen mesh. Although geometric optimisation is, therefore, a constrained optimisation problem, it is generally found that the constraints are inactive and that the optimum solution lies within the region described by the constraints.

Convexity is an important and desirable property both for the objective function, and for the region defined by the constraints. If the objective function is convex, then there is a unique global minimum and there are no local or false minima that could lead to convergence to the wrong solution. Convexity of the region defined by the constraints is important, because, even if an objective function is convex, regions which are not convex can lead to false minima[12,13].

In terms of the regions defined by the constraints it is clear that convexity is not a property that naturally occurs for geometric optimisation of yield-line patterns; one can envisage many problems for which the bounding edges of the slab do not form a convex region. Indeed, Example 2 of section 2 of this paper is just such a case. However, even if the boundary of the slab is not convex, it is still possible to subdivide the slab into regions which are convex and, thus, to alleviate this potential problem. Classification of the objective function, on the other hand, is more complex, since at any given point it can only be determined numerically (linear program I in Fig. 2). It may be that physical arguments can provide the necessary classification. Indeed, for a particular restricted class of slabs with no internal supports and with convex polygonal boundaries, all of which either simply supported or built-in, it is possible to prove that the objective function is convex[14]. However, for arbitrary configurations of slab, such proofs are not available and, for the examples now considered, resort needs to be made to numerical experiment.

*EXAMPLE NUMBER 1*

A square slab is simply supported on three edges with a total load *P* uniformly distributed over the entire area is shown in Fig. 4. This example and mesh are taken from reference [9].

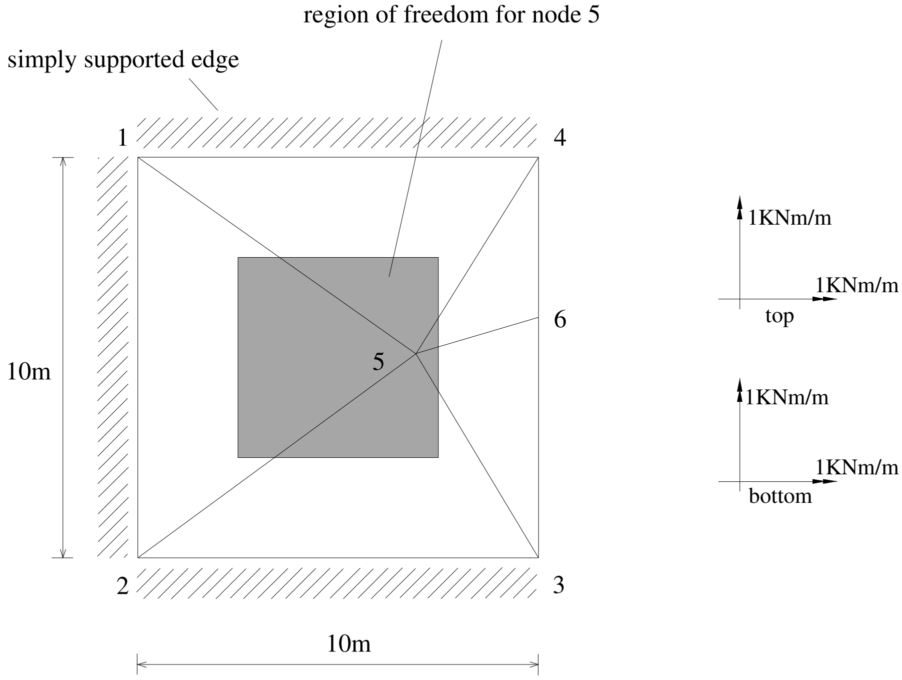



**Figure 4:** Square slab

The three geometric variables considered in this example are the *x*-coordinate of node 5 and the *y*-coordinates of nodes 5 and 6. With these variables there are three possible modes of fracture pattern depending on whether $y_6$ is greater than, equal to, or less than $y_5$. These three modes are shown in Fig. 5 which also identifies the portions of the region defined by the constraints in which each mode is active for a constant value of $x_5$ = 6.5m. The true solution for this problem is $x_5$ = 6.514m, $y_5$ = $y_6$ = 5.000m with a corresponding critical collapse load of *P* = 14.140KN. With the constraint $x_5$ = 6.500m the solution $y_5$ = $y_6$ = 5.083m is achieved which has a corresponding critical collapse load of *P* = 14.144KN.

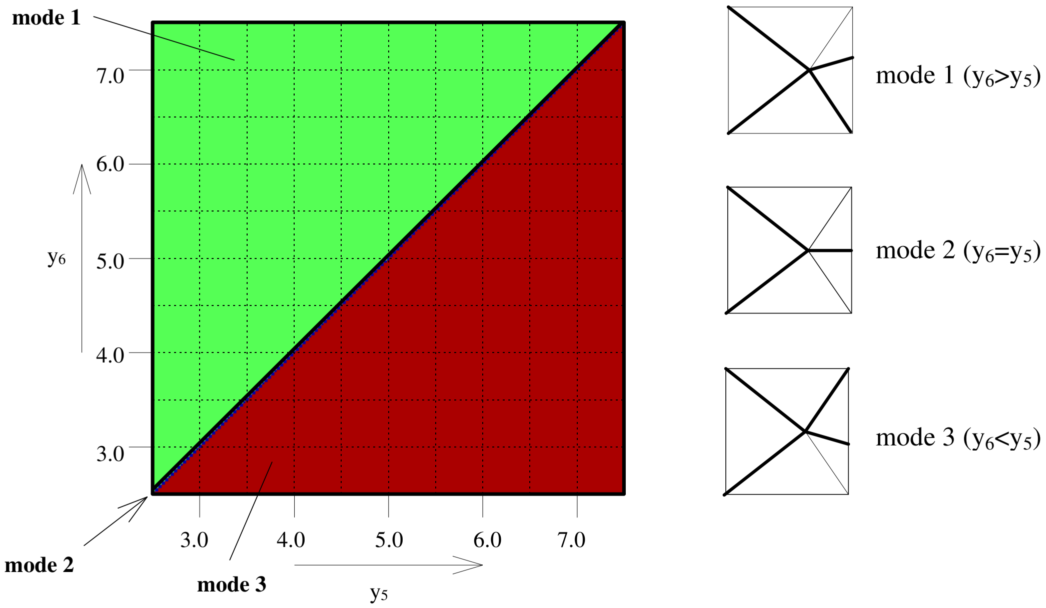



**Figure 5:** Fracture pattern as a function of $y_5$ and $y_6$ plotted for $x_5 = 6.5$m

The way in which the objective function (collapse load) varies over this region is shown in Fig. 6. The objective function is clearly convex and is anti-symmetric about the lines $y_5$ = 5m and $y_6$ = 5m.

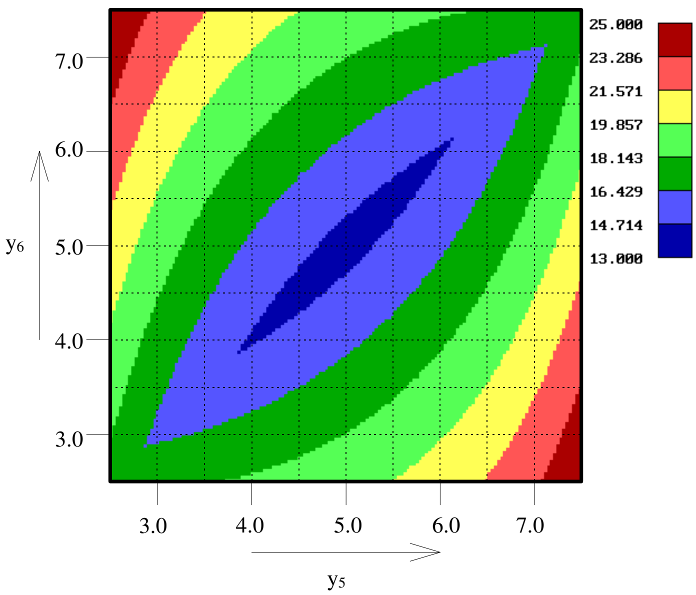



**Figure 6:** Collapse load as a function of $y_5$ and $y_6$ plotted for $x_5 = 6.5$m

As observed in reference [9], the objective function, whilst being continuous, has discontinuous gradient along the line $y_5 = y_6$. This discontinuity is illustrated in Fig. 7 which shows the way in which the collapse load varies along the (diagonal) lines $y_6 = 10 - y_5$ and $y_6 = y_5$ for which the gradient is continuous.

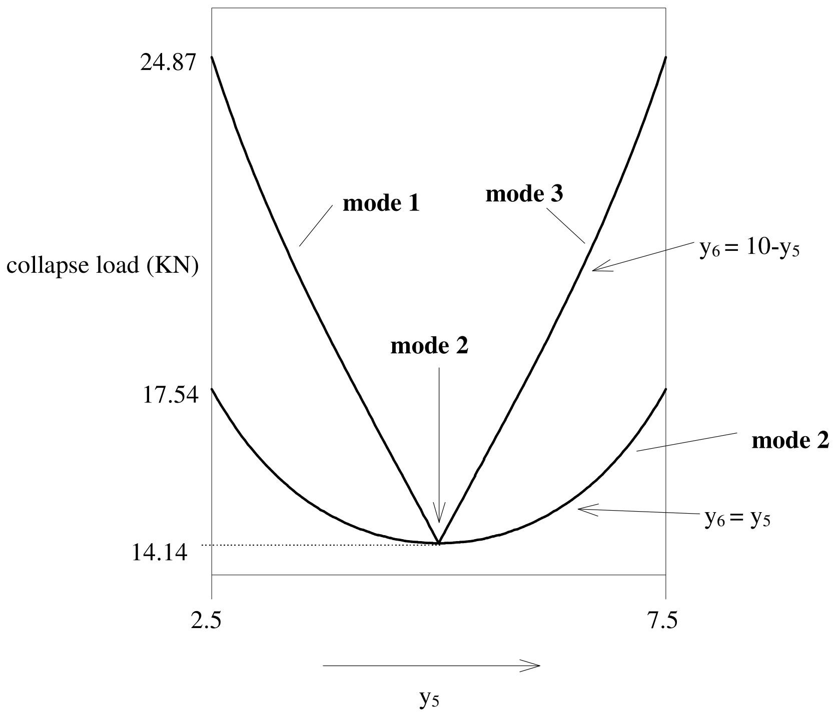



**Figure 7:** Collapse load for $x_5 = 6.5$m

For this example both the geometric optimisation algorithms detailed in references [9] and [10] fail to converge to the correct solution. One way to achieve a solution is to link the variables $y_6$ and $y_5$ with the linear equation $y_6 = y_5$. This reduces the total number of independent geometric variables from three to two and by enforcing, *a priori,* the true mode of failure ensures that the correct solution is recovered. Fig. 8 shows the collapse load as a function of $y_5$ and $x_5$ plotted for $y_6 = y_5$. This figure corresponds to Figure 5 in reference [9] and indicates the smoothness of both the function and its gradient.

It should be noted that whilst linkage of geometric variables is perfectly feasible, it does rely on a knowledge of the correct mode of fracture pattern; had a different linear linkage been defined, say $y_6 = y_5 + r$ with $r \neq 0$, then the critical solution could not have been found.

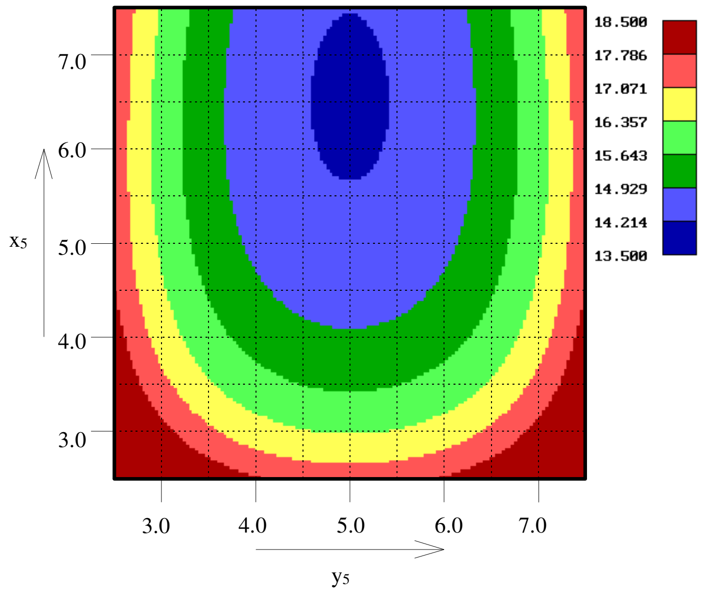



**Figure 8:** Collapse load as a function of $y_5$ and $x_5$ for $y_6 = y_5$

*EXAMPLE NUMBER 2*

An L-shaped slab simply supported on three edges, with corner column support and with a total load *P* uniformly distributed over the entire area of the slab is shown in Fig. 9. This example and mesh have been taken from reference [8].

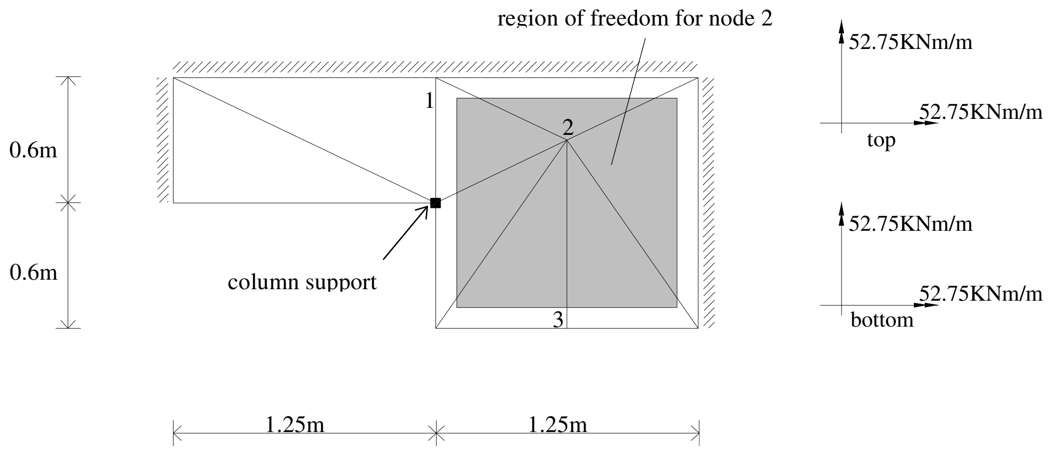



**Figure 9:** L-shaped slab

The four geometric variables considered in this example are the *x*-coordinate of nodes 1, 2 & 3 and the *y*-coordinate of node 2. With these variables, there are eight possible modes of fracture pattern. Unlike Example 1, the modes in this example are not easily classify in terms of the geometric variables. Examples of the eight modes (with the coordinates of node 2 given in parentheses) are shown in Fig. 10 which also indicates the portions of the region defined by the constraints in which each mode is active for constant values of $x_1$ =1.25m and $x_3$ =1.875m.

The true solution for this problem is $x_1$ = 1.72m, $x_2$ = 2.01m, $y_2$ = 0.51m and $x_3$ =1.87m which has a corresponding critical collapse load of $P$ = 52.47KN. With the constraints $x_1$ = 1.25m and $x_3$ =1.875m, the critical solution cannot be obtained and the optimum position of the nodes is then $x_2$ = 2.00m, $y_2$ = 0.544m which has a corresponding collapse load of $P$ = 60.25KN.

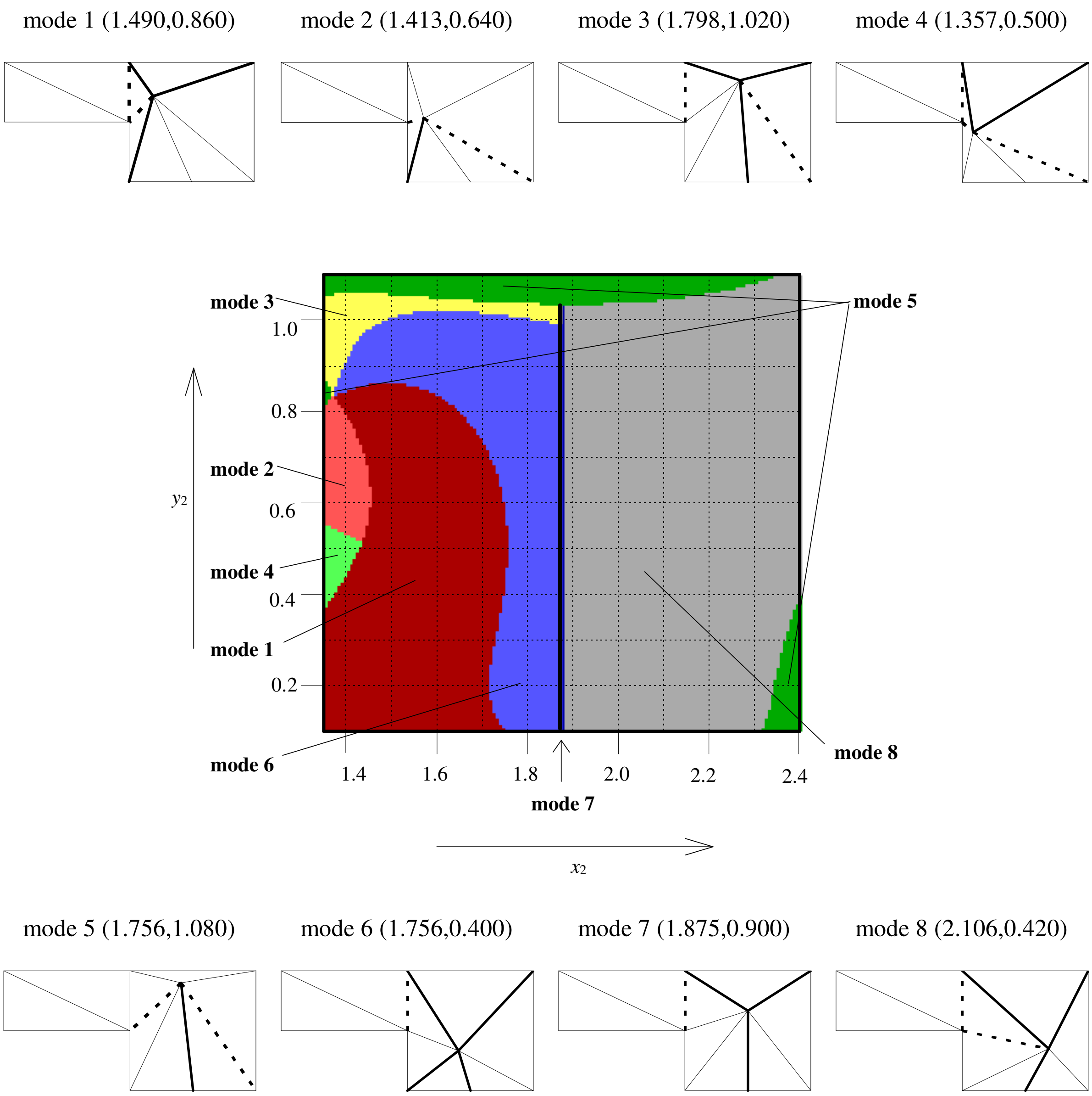



**Figure 10:** Fracture pattern as a function of $x_2$ and $y_2$ for $x_1$ = 1.25m and $x_3$ =1.875m

The way in which the objective function varies over the region is shown in Fig. 11. Similar to Example 1, the function is seen to be smooth but to have discontinuities in its gradient. The nature of these discontinuities is further illustrated in Fig. 12 which shows the way in which the objective function varies with $x_2$ along the line $y_2 = 0.45$m.

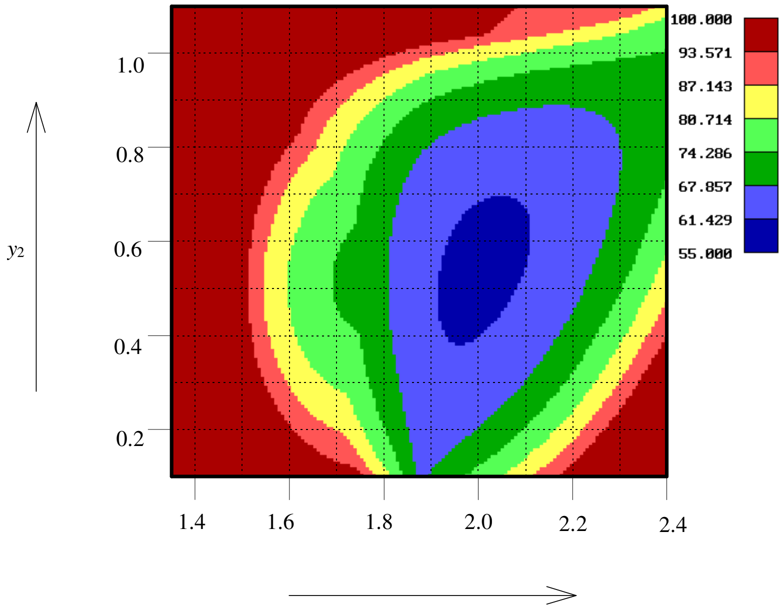



**Figure 11:** Collapse load as a function of $x_2$ and $y_2$ for $x_1 = 1.25$m and $x_3 = 1.875$m

For this example the geometric optimisation algorithm of reference [10], and by implication that of reference [9], <u>can</u> determine the critical solution.

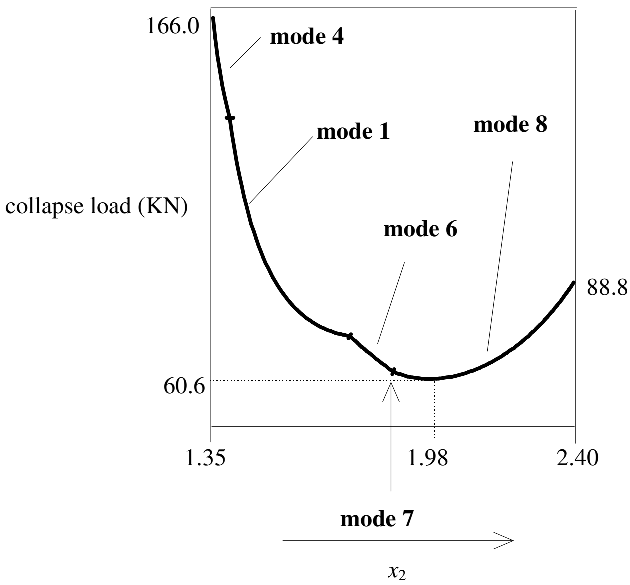



**Figure 12:** Collapse load as a function of $x_2$ for $y_2 = 0.45$m, $x_1 = 1.25$m and $x_3 = 1.875$m

From these results it would appear that the objective function for both examples is convex and that the constraints are inactive. It is also observed that the objective functions are made up of piecewise smooth functions which are continuous but which have discontinuities of slope across the interfaces of adjacent functions. In terms of the number of modes of fracture that occur, Example 2 is seemingly more complex than Example 1 with eight rather than three modes of fracture. However, whereas for Example 1 the solution lies on the intersection of three modes of failure, for Example 2 the true solution lies unambiguously within the region of a single mode (mode 8). The difference in the shapes of the two objective functions is also noteworthy. For Example 2 the objective function is bowl shaped around the minimum with similar gradients in all directions. For Example 1 the objective function is a steep-sided valley[12] with significantly different gradients in the directions of the two principal axes (see Fig. 7). Indeed, not only is the valley steep-sided but it is also sharp-edged[12]. It is also observed that for Example 1 the principal axes lie at 45 degrees to the coordinate axes.

The above mentioned characteristics exhibited by Example 1 are well known to cause difficulties for optimising algorithms[12,13]. Individually some of these characteristics seem to cause no significant problem. For example, numerical experiments carried out on slope-discontinuous ($C^0$) one-dimensional functions showed that, even when the minimum coincides with the point of slope discontinuity, the problem can always be solved robustly. It seems likely, therefore, that it is the combined effect of these characteristics that causes the currently advocated, gradient method, algorithms to fail. The characteristics which seem to cause difficulties are all related to the gradient of the function and it is probably not surprising, therefore, that methods relying on accurate gradient information fail. On the basis that direct search methods do not make use of gradient information and should, therefore, be immune from such difficulties, use of a direct search method is now advocated.

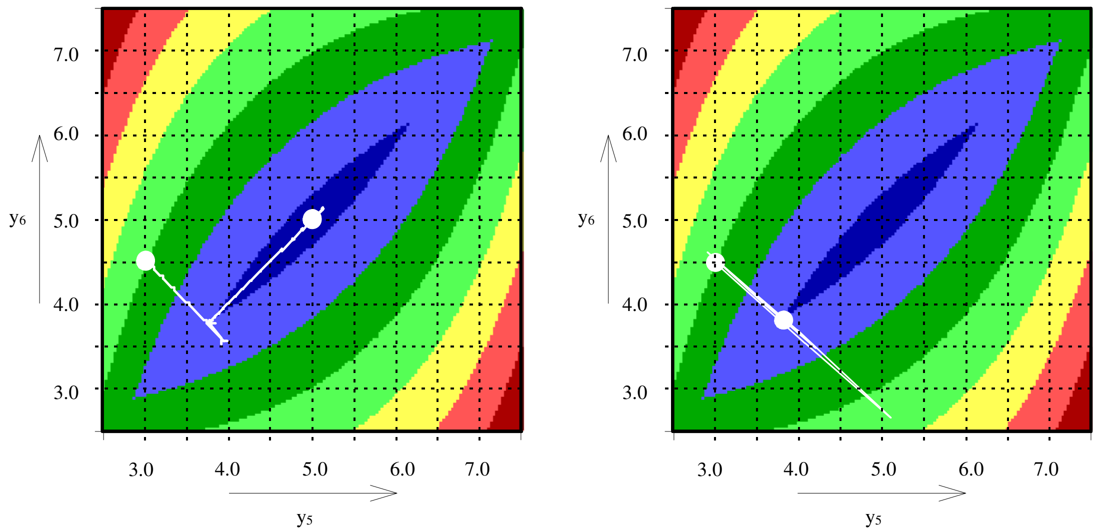
## A DIRECT SEARCH STRATEGY

As an alternative to the gradient methods used in the currently advocated algorithms, a simple direct method is now proposed and evaluated. The direct search algorithm that the current authors have used is the Hooke-Jeeves method[15]. This popular algorithm is recommended[13] as being '...a very efficient and ingenious procedure' and has the added advantage of being extremely simple to code.

The Hooke-Jeeves method progresses towards the optimum solution in a cyclic process consisting of two stages. The first stage of the cycle involves a series of *local explorations* about the current *base point*. These explorations are carried out in the positive and negative directions of the coordinate axes with a distance equal to the current step length. All coordinate axes are considered in turn and explorations which lead to a reduction in the objective function are immediately implemented. In this way the current base point is progressively updated until all the coordinates have been considered. The second stage of the cycle is the *pattern move* which defines the new base point for the next series of local explorations. This move is based upon an extrapolation of the progress made during the previous series of local explorations. If the base point remains unchanged after a series of local explorations then the step length is reduced and the series of local explorations is repeated. When the step length is reduced below some predefined minimum step length, the cycle is terminated with convergence being deemed to have occurred.

The implementation of the Hooke-Jeeves algorithm detailed in reference [12] was used in which it is necessary to define the initial base point, the initial step length (assumed constant for all variables), the step length reduction factor, and the minimum step length

Although the solution to the optimisation problem is independent of these parameters, they cannot be chosen arbitrarily. For example, poor choice of initial base point in conjunction with an initial step length which is too large can result in a pattern move to a base point which is outside the region defined by the constraints. This situation needs to be avoided since it will generally mean that the geometry of the slab and/or the topological integrity of the mesh is violated. For the geometric optimisation problem, in which constraints on geometric variables are generally linear, a good initial base point might be defined as the centroid of a set of unit masses placed at the vertices of the constrained region. Again, an intelligent choice of initial step length might be to take a fraction, say one tenth, of the minimum distance between the vertices of the constrained region. The value of the step length reduction factor is clearly bounded as $0 < \rho < 1$ with a value of 1/10 being found satisfactory in practice[13].




(a) Hooke-Jeeves algorithm (b) Sequential linear programming

**Figure 13:** Solution trajectories plotted over the objective function for Example 1

As an illustration of the ability of the Hooke-Jeeves method to solve problems which are not soluble with other currently advocated algorithms, the solution trajectories for the Hooke-Jeeves method and the sequential linear programming approach of reference [10] are compared for Example 1 in Fig. 13. The initial base point is $y_5$ = 3.0m and $y_6$ = 4.5m. This point does not lie at the centroid of a set of unit masses placed at the vertices of the constrained region but has been chosen so that the trajectories can be clearly visible on the figure.

The Hooke-Jeeves method is seen to progress down the sides of the valley until the floor is reached and then to proceed along the valley floor towards the true solution. In contrast to this, the sequential linear program, whilst able to reach the valley floor appears unable to progress along it.

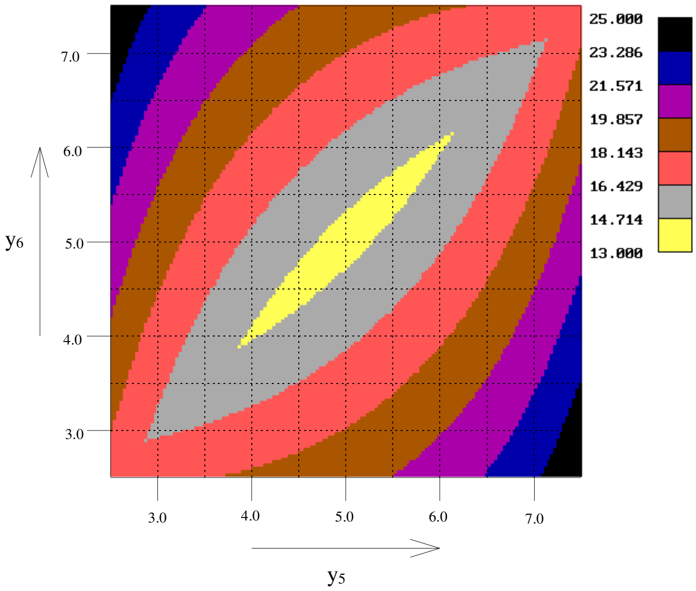
It should be observed that for objective functions similar to that occurring in Example 1 which has a long valley orientated at 45 degrees to the coordinate axes, the Hooke-Jeeves method can fail if the initial base point, or indeed during solution an intermediate base point, lies on the line $y_6 = y_5$. The reason for this is that local exploration about a base point on the line $y_6 = y_5$, which takes place along the coordinate axes, reveals no new base point and the method continues to take futile exploratory steps along the coordinate axes until the step length is reduced below the value which flags termination. There are a number of solutions to this potential difficulty. One way is to re-define the variables in terms of some other, alternative, set of axes. For Example 1 a suitable alternative set of axes could be obtained by rotating the existing set through 45 degrees. Alternatively, and more simply, one could just try restarting from an different base point.

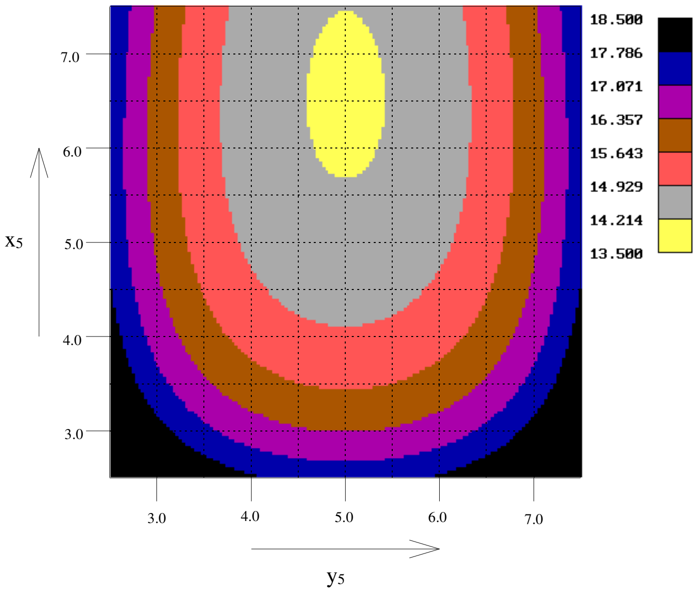
## CONCLUSIONS

In the context of automated yield-line analysis, many perfectly feasible slab configurations result in objective functions which appear not to be amenable to solution with gradient methods. Robustness of solution is an important property of any geometric optimisation algorithm and in the pursuit of this, currently advocated gradient based algorithms have been abandoned in favour of a more simple, direct search method. The Hooke-Jeeves algorithm reported in this paper has been demonstrated to be robust and to be able to solve problems which are, at present, intractable to algorithms based on gradient methods.

## REFERENCES

1. K.W. Johansen, *Yield Line Theory*, Cement and Concrete Association, London, 1962.
2. A. Ghali and A.M. Neville, *Structural Analysis: A Unified Classical and Matrix Approach*, 3rd edn, E & F.N. Spon, 1996.
3. W.L. Shoemaker, 'Computerized Yield Line Analysis of Rectangular Slabs', *Concrete International*, August, 62-65, (1989).
4. A. Hillerborg, 'Yield Line Analysis', *Concrete International*, May, 9, (1991).
5. D. Johnson, 'Is Yield-Line Analysis Safe?', *proceedings of the Third Canadian Conference on Computing in Civil & Building Engineering*, 494-503, (1996).
6. J. Munro, and A.M.A. da Fonseca, 'Yield Line Method by Finite Elements and Linear Programming', *Structural Engineer*, **56b**(2), 37-44, (1978).
7. K.V. Balasubramanyam and V. Kalyanaraman, 'Yield-Line Analysis by Linear Programming', *J. Struct. Engineering*, **114**(6), (1988).
8. A.C.A. Ramsay and D. Johnson, 'Automated Yield-Line Analysis and Geometric Optimisation of Practical Slab Configurations', *submitted to Int. J. Engineering Structures*, December (1996).
9. A. Jennings, A. Thavalingam, J.J. McKeown and D. Sloan, *On the Optimisation of Yield Line Patterns*, Developments in Computational Engineering Mechanics, 209-214, B.H.V. Topping, (ed.) Civil-Comp Press, Edinburgh, 1993.
10. D. Johnson, 'Yield-Line Analysis by Sequential Linear Programming', *Int. J. Solids Structures*, **32**(10), 1395-1404, (1994).
11. J.J. McKeown, A. Jennings, A. Thavalingam and D. Sloan, *Optimisation Techniques for Generating Yield-Line Patterns*, Advances in Structural Optimisation, 161-169, B.H.V. Topping, and M Papadrakakis (eds.), Civil-Comp Press, Edinburgh, 1994.
12. A.A. Smith, E. Hinton and R.W. Lewis, *Civil Engineering Systems: Analysis & Design*, Wiley, 1983.
13. B.D. Bunday, *Basic Optimisation Methods*, Edward Arnold, 1985.
14. A. Jennings, 'On the Identification of Yield-Line Collapse Mechanisms', *Int. J. Eng. Struct.* **18**(4), 332-337, (1996).
15. R. Hooke and T.A. Jeeves, 'Direct Search Solution of Numerical and Statistical Problems', *J. Applied Computer Methods*, **8**, 212, (1961).


25.000
23.286
21.571
19.857
18.143
16.429
14.714
13.000
7.0
6.0
5.0
4.0
3.0
y6
3.0
4.0
5.0
6.0
7.0
y5


18.500
17.786
17.071
16.357
15.643
14.929
14.214
13.500
7.0
6.0
5.0
4.0
3.0
x5
3.0
4.0
5.0
6.0
7.0
y5