A dynamic penalty function method for the solution of structural optimization problems

J. A. Snyman, Nielen Stander, and W. J. Roux

Structural Optimization Research Group, Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, Republic of South Africa

This paper presents an adaptation of an existing dynamic trajectory method for unconstrained minimization to handle constrained optimization problems. This is done by the application of a dynamic penalty parameter procedure to allow for the constraints. The method is applied to structural optimization problems that involve the determination of minimum weight structures of trusses and frames, subject to stress, displacement, and frequency constraints, under various prescribed load conditions. Because structural problems, in general, require detailed finite-element analyses to evaluate the constraint functions, the direct application of the trajectory method, requiring updated information at each step along the path, would be expensive. This problem is overcome by the successive application of the trajectory method to approximate quadratic subproblems that can be solved economically. The comprehensive new approach is called the DYNAMIC-Q method. The method is successfully applied to a number of truss and frame problems and is found to be both reliable and easy to use.

Keywords: constrained minimization, penalty function, structural optimization

1. Introduction

The dynamic method for unconstrained minimization as embodied in the final algorithm LFOPl(b) was originally proposed by Snyman.1,2 The method seeks the minimum of a function by considering the associated dynamic problem of the motion of a particle of unit mass in an n-dimensional conservative force field, in which the potential energy of a particle at point \( x(t) = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n \) at time \( t \) is taken to be the function \( f(x) \) to be minimized. In particular the method requires the solution of the equations of motion:

\[
\dot{x}(t) = -V(f(x(t)))
\]

subject to the initial conditions:

\[
\begin{align*}
  x(0) &= x_0 \\
  \dot{x}(0) &= v_0
\end{align*}
\]

The method computes an approximation to the associated trajectory and applies an interfering strategy that allows for a drastic reduction in the kinetic energy of the particle whenever the kinetic energy appears to decrease along the path. The consequence of such a strategy, based on an energy conservation argument, is that a systematic reduction in the potential energy \( f \) is obtained. The particle is thus forced to follow a path to a local minimum at \( x^* \). For economy the trajectory is computed by the “leap-frog” (Euler forward—Euler backward) method. The method contains some heuristic elements, for example, relating to the time step selection. Outstanding features of this method for unconstrained minimization are that it requires no line searches and that only gradient information, and no function evaluations, are required. For more details of the method the reader is referred to the original papers.1,2

In practice over many years, the method has proved to be extremely robust and in many cases as efficient as, if not better than, the conjugate gradient method. A prominent feature of the method in practice is the relatively fast convergence to the neighborhood of the solution, although in some cases the final convergence to the exact solution may be very slow. To improve the convergence rate of the algorithm near the local minimum the method has also been combined with a gradient-only version of the conjugate gradient method.3 The original method is also the basic minimization procedure in the global optimization algorithm of Snyman and Fatti.4 This algorithm has been successfully applied to the optimal design of laminated composite plates.5 More recently a mathematically more rigorously constructed trajectory method for unconstrained minimization with less heuristic characteristics has been developed.6 Although its performance is good its efficiency is definitely worse than that of the original algorithm LFOPl(b),2 which represents the most efficient and robust version of all the trajectory methods studied by Snyman to date.

In addition to the first author’s extensive personal experience in the application of LFOPl(b) to numerous
ad hoc industrial and academic problems, the method has also recently been applied to optimal control problems and to vibration isolation problems in mechanical engineering. In the case of the latter problems the objective functions are effectively non-differentiable and nonconvex. Nevertheless the trajectory method succeeds in solving these problems where traditional quasi-Newton and conjugate gradient methods have failed. Snyman’s trajectory methods have also recently been applied to a number of problems in chemistry. One of the advantages of LFOPl(b) reported by workers in this field is the tendency of the trajectory method to avoid saddle points, which frequently occur in chemical potential studies. Finally, because the method requires no function line searches and uses only gradient vector information, it is suitable for implementation on parallel computers.

In this study we consider the application of the trajectory method to the solution of constrained optimization problems through the consideration of successive approximate quadratic subproblems and the formulation of associated penalty functions. In particular structural optimization problems are tackled, which involve the determination of minimum weight structures of trusses and frames, subject to stress, displacement, and frequency constraints, under various prescribed load conditions. The proposed comprehensive method is called the dynamic quadratic (DYNAMIC-Q) method.

2. Penalty function formulation for the dynamic method

The penalty function method is the most simple and straightforward approach to constrained optimization problems. Consider the constrained problem:

$$\text{minimize } f(x), \quad x \in \mathbb{R}^n$$

such that

$$g_i(x) \leq 0, \quad i = 1, 2, \ldots, m$$

and

$$h_j(x) = 0, \quad j = 1, 2, \ldots, r$$

where \( f(x) \), \( g_i(x) \), and \( h_j(x) \) are scalar functions of \( x \).

The penalty function is then defined as

$$p(x) = f(x) + \sum_{i=1}^{m} \alpha_i g_i^2(x) + \sum_{j=1}^{r} \beta_j h_j^2(x)$$

where \( \alpha_i = 0 \) if \( g_i(x) \leq 0 \)

$$= \rho_i \text{ if } g_i(x) > 0$$

For simplicity the penalty parameters \( \rho_i, i = 1, 2, \ldots, m \) and \( \beta_j, j = 1, 2, \ldots, r \) usually take on the same positive value \( \rho_i = \beta_j = \mu \), and for high accuracy \( \mu \) is usually a very large number relative to the absolute values that the constraint functions \( g_i(x) \) and \( h_j(x) \) may assume. It can be shown that under relatively normal conditions, as \( \mu \) tends to infinity, the unconstrained minimum of \( p(x) \) yields the constrained minimum of the problem defined by expressions (3-5). Unfortunately it turns out that if high accuracy is required and a high value for \( \mu \) is prescribed the unconstrained minimization problem becomes ill-conditioned.

Because LFOPl(b) was found to do particularly well when applied to ill-conditioned problems with “steep valleys” it seemed natural to apply this method to the solution of constrained problems through the unconstrained minimization of penalty functions. Disappointingly LFOPl(b) performed poorly on penalty functions, especially when severe penalties were imposed for high accuracy. The reason for this is that the penalty function does not necessarily have steep valleys along which the particle may move toward the solution. Instead the penalty function usually has steep barriers at the boundaries where inequality constraints are violated. Applying the dynamic method results in the particle following a relatively smooth path but “bouncing back” sharply and abruptly from a penalty barrier once a constraint is violated. This “bounce” is associated with a large backward step from which a new trajectory is initiated. The consequence is that overall an erratic trajectory is obtained, which in many cases fails to converge to an acceptable solution. LFOPl(b) was therefore rarely applied to constrained problems.

Recently, however, a simple adjustment to the penalty function procedure proved to be successful in the application of the dynamic algorithm to penalty function formulations of practical constrained problems. These problems are associated with and derived from optimal control problems. In this modification to the method the penalty parameter \( \mu \) is slowly increased by a small factor at each step along the trajectory until a prescribed limit value of \( \mu \) is reached. Thereafter the remainder of the trajectory, computed with the constant limit value of \( \mu \), is followed until convergence is obtained. In practice we choose an initial value \( \mu = \mu_0 \), a magnification factor \( m \), and a limit value \( \mu_{\text{max}} \) and at each step \( k \) along the trajectory set \( \mu_k = m \mu_{k-1} \) until \( \mu_k > \mu_{\text{max}} \) after which we set \( \mu_k = \mu_{\text{max}} \) and continue until convergence.

The effect of the above dynamic penalty parameter strategy is depicted in Figures 1(a) and (b) for the simple problem:

$$\text{minimize } f(x) = -x_1 - x_2$$

such that

$$g(x) = x_1^2 + 2x_2^2 - 1 \leq 0$$

Figure 1(a) shows the behavior of the algorithm in the case of the usual implementation with a constant penalty parameter, in this case \( \mu = 100 \). The trajectory follows a straight line path from the initial point \( x_0 = (0, 0)^T \) until the 11th step it violates the constraint and bounces back sharply as shown. Hereafter it recovers to near the constraint boundary and then moves extremely slowly toward the optimum at \( x^* = (0.816; 0.408)^T \). For clarity only a few points along the trajectory are indicated. Figure 1(b) shows the trajectory obtained by the modified approach in which the penalty parameter is dynamically increased from an initial value \( \mu_0 = 1 \) by a small factor \( m = 1.1 \) at each step along the path. Clearly a much smoother trajectory is obtained, which converges much faster to the neighborhood of \( x^* \). This simple
example illustrates the potential advantage to be gained by the application of the dynamic penalty parameter (DPP) procedure to constrained optimization problems.

3. Successive approximate subproblems

In this study we apply the DPP procedure to typical sizing structural optimization problems of the form:

$$\begin{align*}
\text{minimize} & \quad f(x) = c^T x \\
\text{subject to} & \quad g_j(x) \leq 0, \quad j = 1, 2, \ldots, m
\end{align*}$$

subject to inequality constraints

$$g_j(x) \leq 0, \quad j = 1, 2, \ldots, m$$

Here the objective $f(x)$ represents the weight of the structure, which is a linear function of cross-sectional areas $x = (x_1, x_2, \ldots, x_n)^T$ of the structural members, and the $g_j(x)$ represent stress and displacement constraints under various prescribed loads. In general the $g_j(x)$ are nonlinear functions of $x$.

In applying the DPP procedure we require the gradient of the penalty function, which includes the gradients of the constraints at each step along the trajectory. For structural problems the evaluation of these gradients requires a detailed analysis, usually by the finite-element method, of the complete structure. It would therefore be extremely expensive to apply the trajectory method to these problems because the method, as it stands, requires updated gradient information and therefore a structural analysis at each step along the solution path. This problem can be overcome by the successive application of the method to approximate quadratic subproblems. The quadratic subproblems have simple structures so that they can be solved relatively economically.

The success of quadratic subproblems, $k = 0, 1, 2, \ldots$, is formed at successive designs $x^{(k)}$ by using the constraint approximations

$$g_j(x) = g_j(x^{(k)}) + \nabla g_j(x^{(k)})^T (x - x^{(k)})$$

$$+ c_j^{(k)} \| x - x^{(k)} \|^2; \quad j = 1, 2, \ldots, m$$

where $\nabla g_j$ denotes the gradient vector, $T$ the transpose, and $\| \|$ the euclidean norm. These approximated constraints together with the objective function $f(x)$ define the subproblem, which is solved to find the optimal point, $x^{(k)}$. Once the solution $x^{(k)}$ is obtained a new approximate subproblem is formed by analyzing the design $x^{(k)}$, and this new problem is then solved in turn to give $x^{(k+1)}$.

This succession of solutions is continued until $x^{(k)}$ converges to, hopefully, the solution $x^*$ of the original problem. A convergence criterion

$$|f(x^{(k+1)}) - f(x^{(k)})|$$

$$f(x^{(k+1)}) < \epsilon_5$$

based on the cost values of the two most recent subproblem solutions, and where $\epsilon_5$ is a specified small positive value, may be used.

An important question is the calculation of an appropriate $c_j^{(k)}$ for each constraint. In the first subproblem $c_j^{(0)}$ can be assumed to be an arbitrary number for all $j = 1, 2, \ldots, m$. A value $c_j^{(0)} = 0$ can be used, in which case the first subproblem becomes a linear programming problem. In the numerical experiments that follow, a value $c_j^{(0)} = 0.01$ was used throughout. Thereafter the coefficients $c_j^{(k+1)}$ can be determined from the approximations

$$g_j(x^{(k)}) = g_j(x^{(k+1)}) + \nabla g_j(x^{(k+1)})^T (x^{(k+1)} - x^{(k)})$$

$$+ c_j^{(k+1)} \| x^{(k+1)} - x^{(k+1)} \|^2; \quad j = 1, 2, \ldots, m$$

by using data of the two most recent consecutive design points, and the $c_j^{(k+1)}$ are then used to form the $(k + 1)$th subproblem.

The comprehensive proposed method, involving the application of the DPP procedure, which uses the trajectory method LFOP1(b) on successive quadratic
subproblems of the original problem, will hereafter for convenience be referred to as the dynamic-quadratic (DYNAMIC-Q) method.

4. Parameter selection for the DYNAMIC-Q method

The original LFOPl(b) algorithm requires, in addition to the specification of an initial starting point \( x_0 \), only three parameter settings. They are (see the original papers\(^1\) - 1): 1. the initial time step \( \Delta t \) (DT), 2. the maximum step size \( \delta \) (DELT), and 3. the convergence tolerance \( \varepsilon \) on the norm of the gradient \( \nabla \).

A maximum allowable number of steps per trajectory (KMAX) are usually also specified. The parameter names in brackets denote the corresponding variables in the original FORTRAN program LFOPl(b).

The efficiency of the LFOPl(b) method in the case of normal unconstrained optimization is not very sensitive to the selection of the parameters \( \Delta t \) and \( \delta \). The time step \( \Delta t \) is automatically adjusted along the trajectory to ensure efficiency so that the initial choice for \( \Delta t \) is not crucial. The default setting is \( \Delta t = 0.5 \), and this value was indeed used throughout this study. It may be useful to state here that, because the size of the first step is of the order \( \| \nabla f(x_0) \| \Delta t \), the initial time step \( \Delta t \) should in general be chosen large with reference to \( \| \nabla f(x_0) \| \) to ensure a meaningfully large initial step.

The maximum step size \( \delta \) is usually chosen to be of the same order of magnitude as the size of the region of interest. This specification prevents the trajectory from jumping far outside the region of interest. As a conservative rule we choose \( \delta \) to be of the same order of magnitude as the norm of the gradient \( \nabla f \).

The convergence tolerance \( \varepsilon \) on the norm of the gradient vector is arbitrarily chosen as \( 10^{-5} \), and allowance is made for a maximum of 3000 steps. In addition, should the step size be effectively zero, i.e., \( \| \Delta x \| < \varepsilon \), then the trajectory is also terminated.

The application of the DPP procedure introduces additional parameters \( \mu_0, m, \) and \( m_{\max} \) already discussed in Section 2 of this paper. Typical settings for these parameters for structural problems are

\[
\mu_0 = 1.0, \quad m = 1.1 \text{ to } 3.0, \quad \text{and} \quad m_{\max} = 10^7 \text{ to } 10^{10}
\]

The extremely high value for \( m_{\max} \) was necessary because smaller values resulted in significant violations of the constraints at the computed optimal point \( x^* \). This behavior must be attributed to the special nature of the structural constraint functions.

Finally the use of successive quadratic subproblems introduces further parameters. In the first place a convergence tolerance \( \varepsilon_x \) (see expression (10)) on the solution of successive subproblems must be specified. A typical choice is \( \varepsilon_x = 10^{-4} \). Second, it seems sensible to reduce the maximum stepsize \( \delta \) as the solution to the subproblem approaches \( x^* \), but not to do so until a subproblem solution is obtained that corresponds to a feasible point of the original problem. This requires a feasibility tolerance \( \varepsilon_f \) to be used in the test

\[
g(x_k) \leq \varepsilon_f \text{ for all } j
\]

where \( \varepsilon_x \) is a small positive number. The tolerance that can be satisfied here is clearly dependent on \( \mu_{\max} \). Typically we choose \( \varepsilon_f = 10^{-4} \) to \( 10^{-6} \) depending on the value of \( \mu_{\max} \).

The overall method thus starts with a \( \delta \) value chosen to be of the same order of magnitude as the norm of the expected solution. This value is kept constant from subproblem to subproblem until criterion (12) is met. Thereafter \( \delta \) is reduced by a factor \( 1/r \) whenever \( f(x^{k+1}) > f(x^k) \) by setting \( \delta := \delta/r, r > 1 \). Typically we use \( r = 1.5 \).

In summary the main parameters of DYNAMIC-Q are
1. for the LFOPl(b) algorithm: \( \Delta t, \delta, \varepsilon \), and \( \varepsilon_x \);
2. for the DPP procedure: \( \mu_0, m, \mu_{\max} \); and
3. for the sequence of quadratic subproblem solutions: \( \varepsilon_x, \varepsilon_f \), and \( r \).

A typical choice for the parameter set is
1. \( \Delta t = 0.5; \delta = 0.1 \text{ to } 1.0; \varepsilon = 10^{-3} \); and \( \varepsilon_x = 10^{-10} \);
2. \( \mu_0 = 1.0; m = 1.1 \text{ to } 3.0; \) and \( \mu_{\max} = 10^7 \text{ to } 10^{10} \); 3. \( \varepsilon_x = 10^{-4}; \varepsilon_f = 10^{-4} \text{ to } 10^{-6}; \) and \( r = 1.5 \).

As will be shown DYNAMIC-Q is relatively insensitive to the parameters and converges to an acceptable solution (at least from an engineering point of view) for any reasonable choice, provided \( \delta \) is chosen according to the guidelines set out above. This makes the method both robust and easy to use.

5. Test problems and results

To illustrate the effectiveness of the DYNAMIC-Q method for structural optimization problems it was applied to five truss problems and five rigidly framed structures. For each of these structures the cross-sectional areas of the uniform bars or beams are taken as the design variables. The objective is to minimize the weight of each structure subject to certain prescribed constraints on stress and displacement, and in the case of the frames on the natural frequency of the structure as well. The structural analyses required for the optimization were done by means of the EDSAP\(^20\) implementation of the finite-element method.

The five truss problems (problems 1-5) considered here include four standard test problems selected from the literature. These problems, varying in size from 3 to 200 bars and from 3 to 29 design variables, are well documented in the literature\(^{21-23}\). The basic information for these problems is tabulated in Table 1(a). For further information regarding these problems the reader is

<table>
<thead>
<tr>
<th>Problem number</th>
<th>Reference</th>
<th>n</th>
<th>m</th>
<th>Bars</th>
<th>( SF ) in ( \text{in}^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21, 22</td>
<td>3</td>
<td>33</td>
<td>3</td>
<td>10, 5, 5</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>10</td>
<td>46</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>7</td>
<td>179</td>
<td>25</td>
<td>2.22</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>29</td>
<td>2129</td>
<td>200</td>
<td>Ref. 21</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>7</td>
<td>14700</td>
<td>379</td>
<td>100 mm²</td>
</tr>
</tbody>
</table>
referred to the indicated references. The fifth truss structure is a transmission tower of local interest, consisting of 379 bars with seven design variables. As an example the 200-bar truss problem, with 29 variables, is depicted in Figure 2(a).

The five example frame problems (problems 6–10) are a simple portal frame, (a), subjected to stress constraints only and, (b), simultaneously to stress, displacement, and frequency constraints; a one-bay two-story frame, (c), subjected to stress constraints only and, (d), simultaneously to stress, displacement, and frequency constraints; and finally, (e), a two-bay six-story frame subjected to stress constraints only. These problems for which the basic information is given in Table 1(b), are also well documented in the literature. As an example the two-bay six-story frame with 30 members and 30 variables is shown in Figure 2(b).

The performance of the DYNAMIC-Q algorithm on each of the test problems is demonstrated by depicting the respective convergence histories in Figures 3–12. In these graphs the objective function value is plotted against the number of structural analyses. The starting points $x_0$ are as given in Table 1 and for comparison the behavior of the well-established sequential quadratic programming algorithm.

<table>
<thead>
<tr>
<th>Problem number</th>
<th>Reference</th>
<th>$n$</th>
<th>$m$</th>
<th>Members</th>
<th>$x^2 m^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>25, 26</td>
<td>4</td>
<td>44</td>
<td>4</td>
<td>0.015</td>
</tr>
<tr>
<td>7</td>
<td>25, 26</td>
<td>4</td>
<td>57</td>
<td>4</td>
<td>0.015</td>
</tr>
<tr>
<td>8</td>
<td>25, 26</td>
<td>8</td>
<td>88</td>
<td>8</td>
<td>0.03</td>
</tr>
<tr>
<td>9</td>
<td>25, 26</td>
<td>8</td>
<td>113</td>
<td>8</td>
<td>0.03</td>
</tr>
<tr>
<td>10</td>
<td>25, 26</td>
<td>30</td>
<td>420</td>
<td>30</td>
<td>Ref. 26</td>
</tr>
</tbody>
</table>

Figure 2(a). The 200-bar truss problem with 29 variables.

Figure 2(b). The two-bay six-story frame with 30 members and 30 variables.

Figure 3. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the three-bar truss (problem 1).
Dynamic penalty function method for structural optimization: J. A. Snyman et al.

Figure 4. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the 10-bar truss (problem 2).

Figure 5. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the 25-bar truss (problem 3).

Figure 6. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the 200-bar truss (problem 4).

Figure 7. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the 379-bar truss (problem 5).

Figure 8. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the simple portal frame subjected to stress constraints only (problem 6).

Figure 9. Convergence histories of DYNAMIC-Q (DYN) and E04UCF (NAG) for the simple portal frame subjected to stress, displacement, and frequency constraints (problem 7).

(SQP) method, as implemented in the NAG subroutine E04UCF, is also shown except for the two-bay six-story frame (problem 10). The figures indicate that both algorithms obtain the same optimal function value and that for some problems (Figures 4 and 5) DYNAMIC-Q solved the problem using fewer structural analyses than E04UCF but in some problems it required more (Figures 3, 6, 8, 9, 10, and 11). In fairness it should, however, be added that for most of the latter problems, corresponding to Figures 8–11, difficulties were experienced with E04UCF in selecting a starting point and parameter settings that gave convergence. The results in the figures represent the successful convergence history for E04UCF only after, through experiment, a suitable choice for the starting point and parameter values were determined. Indeed for the final problem extreme difficulties were experienced in selecting suitable parameters and...
starting point for the NAG routine, and no satisfactory convergence behavior could be obtained. In contrast identical parameter settings (see Section 4) were used for all truss and all frame problems respectively in the application of DYNAMIC-Q and, indeed, parameter studies\textsuperscript{26} have shown that the convergence of this method is relatively insensitive to variations in the parameter values and starting point selection. The insensitivity to the choice of starting point is illustrated in Figure 13, which depicts the convergence behavior for the 10-bar truss using a variety of both feasible and infeasible starting points.

In solving the approximate subproblem an active set strategy may be adopted, in which only those constraints for which $g_j(x^{(k)}) > -\varepsilon_c$ for some $\varepsilon_c > 0$ are considered. This may considerably speed up the solution of the subproblem. In these tests this strategy was only applied to the solution of the truss problems for which a value of $\varepsilon_c = 0.1$ was used.

6. Conclusion

Snyman's proven robust dynamical method for unconstrained optimization has successfully been extended in two respects. First a modification, the DPP procedure, has been proposed that enables the economic and accurate handling of constraints through the straightforward penalty function formulation. Second, to allow for the economic solution of complex structural optimization problems, where the evaluation of constraint functions and sensitivities requires complete and expensive structural analysis, the DPP procedure is applied to a sequence of quadratic approximations of the original problem.

The implementation of the resultant algorithm DYNAMIC-Q, based on the above modifications, to structural problems proved to be straightforward. Practical experience with the algorithm confirms that the modified method is extremely robust, giving relatively rapid convergence, in terms of the required structural analyses, to an acceptably accurate solution for almost any starting point and over a wide range of algorithm parameter settings. In this respect DYNAMIC-Q appears to be more robust than the NAG implementation of SQP. Indeed experience indicates that, although E04UCF may be fast once a suitable starting point and parameter selection has been made, it generally requires fine tuning over a range of parameters to solve a problem successfully. This is in contrast to DYNAMIC-Q, which
readily yields good results from a “cold start” with standard settings.

The generally robust behavior of DYNAMIC-Q is encouraging as it is generally agreed that, for problems requiring extensive computing for the evaluation of gradient information, robustness should be a prime concern. We believe that the proposed method may successfully be applied to the optimization of other systems that require expensive function evaluations. In particular it may be worthwhile to apply the method to the optimization of mechanical systems where the function evaluations are the outcome of expensive time simulations of the behavior of the system.

References

3 Snyman, J. A. Unconstrained minimization by combining the dynamic and conjugate gradient methods. Quaestiones Mathematicae 1985, 8, 33–42
7 Frangos, C. and Snyman, J. A. The application of parameter optimization techniques for linear optimal control system design. Automatica, 1992, 28, 153–157

13 Smith, C. M. How to find a saddle point. Int. J. Quant. Chem. 1990, 37, 773
15 Smith, C. M. Molecular electrostatic force, solution electrostatic energy and solubility of polar and non-polar molecules in water. J. Mol. Struct. (Theochem) 1990, 209, 273–287
20 Wilson, E. L. ED-SAP—An educational version of a three dimensional structural analysis program. University of California, Berkeley, CA, 1988