On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations

Struct Multidisc Optim (2010) 42:165–178
DOI 10.1007/s00158-010-0498-6

RESEARCH PAPER

On the conditional acceptance of iterates in SAO algorithms based on convex separable approximations

Albert A. Groenwold · L. F. P. Etman

Received: 15 July 2009 / Revised: 26 January 2010 / Accepted: 20 February 2010 / Published online: 28 March 2010

Abstract We reflect on the convergence and termination of optimization algorithms based on convex and separable approximations using two recently proposed strategies, namely a trust region with filtered acceptance of the iterates, and conservatism. We then propose a new strategy for convergence and termination, denoted *filtered conservatism*, in which the acceptance or rejection of an iterate is determined using the nonlinear acceptance filter. However, if an iterate is rejected, we increase the conservatism of every unconservative approximation, rather than reducing the trust region. Filtered conservatism aims to combine the salient features of trust region strategies with nonlinear acceptance filters on the one hand, and conservatism on the other. In filtered conservatism, the nonlinear acceptance filter is used to decide if an iterate is accepted or rejected. This allows for the acceptance of infeasible iterates, which would not be accepted in a method based on conservatism. If however an iterate is rejected, the trust region need *not* be decreased; it may be kept constant. Convergence is than effected by increasing the conservatism of *only the unconservative approximations* in the (large, constant) trust region, until the iterate becomes acceptable to the filter. Numerical results corroborate the accuracy and robustness of the method.

Keywords Sequential approximate optimization (SAO) · Diagonal quadratic approximation · Trust region method · Conservatism · Filtered conservatism · Sequential convex programming (SCP)

1 Introduction

Today, gradient-based sequential approximate optimization (SAO) methods based on convex and separable approximations represent a class of recognized and viable methods for solving large-scale optimization problems when the evaluation of the objective and/or the constraint functions are computationally demanding.

Algorithms in this class are also known as sequential convex programming (SCP) methods, e.g. see Fleury (1993), Zillober et al. (2004), and Duysinx et al. (2009). Examples of established algorithms within this class include CONLIN by Fleury and Braibant (1986) and the method of moving asymptotes (MMA) by Svanberg (1987). Algorithmic variants of MMA were presented by, for example, Borrvall and Petersson (2001), Bruyneel et al. (2002), and Zillober et al. (2004). Groenwold and Etman (2008b) presented tailorable convex separable SAO algorithms, in which convexity and separability of the approximations was used for the efficient solution of the approximate optimization subproblems. A dual approach was used in many of the references cited in the foregoing, but different yet efficient methods have also been proposed, e.g. see Zillober (2001).

SAO algorithms based on convex separable approximations often try to avoid the use of (off-diagonal) second order Hession information, because the computational and storage requirements associated with evaluating the Hessian matrix becomes prohibitively high for large scale problems. Instead, SAO algorithms typically use intervening variables

Based on the paper entitled 'Globally convergent SAO algorithms for large scale simulation-based optimization', presented at the 8th World Congress on Structural and Multidisciplinary Optimization, 1–5 June 2009, Lisbon, Portugal.

A. A. Groenwold (✉)
Department of Mechanical Engineering,
University of Stellenbosch, Matieland, South Africa
e-mail: albertg@sun.ac.za

L. F. P. Etman
Department of Mechanical Engineering,
Eindhoven University of Technology, Eindhoven, the Netherlands

to arrive at nonlinear analytical approximations to the objective and constraint functions of (hopefully) high quality. Refer to Haftka and Gürdal (1991) for an introduction into SAO methods, and to Barthelemy and Haftka (1993) for a review of various intervening variables proposed for SAO in structural optimization applications.

Both the aforementioned CONLIN and MMA are for example based on reciprocal-like intervening variables, which have proven to be very effective in structural optimization. Groenwold et al. (2010) developed an SAO algorithm based on convex diagonal quadratic approximations in which approximate second order terms are obtained by constructing quadratic approximations to the original nonlinear intervening variables based approximations; thereby creating 'approximated-approximations'.

The aforementioned algorithms are in their rudimentary forms not globally convergent. What is more, these algorithms are not even guaranteed to terminate. To overcome this, Zillober (1993) added a line search to MMA to obtain a globally convergent optimization method. Generally this requires additional function evaluations, which is less attractive for computationally demanding analysis functions such as encountered in finite element applications. To effect robust global behavior without a line search, the most obvious strategy is to cast the algorithms in a trust-region like framework, e.g. see Conn et al. (2000) and Alexandrov et al. (1998). Trust region methods have widely been used in sequential quadratic programming (SQP) algorithms, often in combination with a merit function, to decide if an iterate is accepted or rejected. Very recently, Fletcher, Leyffer and co-workers have replaced the (problematic) merit function by a nonlinear acceptance filter, e.g. see Fletcher and Leyffer (1998, 2002) and Fletcher et al. (1998, 2002b, a).

As an alternative to trust-region methods for effecting robust global behavior, Svanberg (2002) has developed the concept of *conservative, convex* and *separable* approximations (CCSA) (but only for inequality constrained problems).[1] He has implemented and demonstrated this numerically for the MMA algorithm; the resulting algorithm is known as 'a globally convergent version of MMA'.

However, enforcing termination in SAO algorithms may sometimes be very expensive. Papalambros and Wilde (2000) for example remark that 'any remedy to make MMA globally convergent makes the method arbitrarily slow'. This does not only hold for MMA; for our own algorithms, we have come to the same conclusion, e.g. see Groenwold et al. (2009).

[1] *Linear* equality constraints may in principle be added to conservative convex primal SAO algorithms, and some conservative convex dual SAO algorithms; the notion of conservatism does not exist for nonlinear equality constraints.

Even so, proof of global convergence is highly desirable, in particular if the cost associated with the termination mechanism can be kept small as compared to the unconditional acceptance of iterates. Herein, we aim to contribute to the understanding and further development of conditional acceptance mechanisms for SAO algorithms, and in particular algorithms based on convex separable approximations.

In doing so, we implement the filtered acceptance of iterates for the dual SAO algorithm based on separable diagonal quadratic approximations we have previously proposed, see Groenwold and Etman (2008b). (We have previously also cast a primal SAO algorithm in the filter framework, e.g. see Etman et al. (2006).) We compare the performance of the resulting algorithm with our conservative variant of the algorithm, see Groenwold et al. (2009), and we show that enforcing robust global behavior in both strategies may indeed be hurtful from a computational point of view.

In an attempt to address the computational expense associated with enforcing convergence, we then propose a new strategy, which we denote *filtered conservatism*. Filtered conservatism aims to combine the salient features of trust region strategies with nonlinear acceptance filters on the one hand, and conservatism on the other. In filtered conservatism methods, the nonlinear acceptance filter is used to decide if an iterate is accepted or rejected. However, if an iterate is rejected, the trust region need *not* be decreased; it may be kept constant. Convergence is than effected by increasing the conservatism of only the unconservative approximations in the (large, constant) trust region, until the iterate becomes acceptable to the filter.

Our paper is arranged as follows: In Section 2, we give the inequality constrained problem under consideration. In Section 3, we introduce sequential approximate optimization, and we present the algorithms developed herein in Section 4, without prescribing the approximations used at this stage. Numerical experiments are then presented in Section 5 using a specific approximation function, and some concluding remarks are offered in Section 6. For the sake of completeness, the test functions used are given in Appendix A, and the approximation function used in the examples (a convex spherical quadratic approximation) is given in Appendix B, but many other possibilities exist.

2 Problem statement

We consider a nonlinear inequality constrained optimization problem $\mathrm{P_{NLP}}$ of the form

$$\begin{aligned} \min \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_j(\boldsymbol{x}) \leq 0, \qquad j = 1, 2, \cdots, m, \\ & \check{x}_i \leq x_i \leq \hat{x}_i, \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{1}$$

where $f_0(\boldsymbol{x})$ is a real valued scalar objective function, and $f_j(\boldsymbol{x})$, $j = 1, 2, \cdots, m$ are m inequality constraint functions. $f_0(\boldsymbol{x})$ and $f_j(\boldsymbol{x})$ depend on the n real (design) variables $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}^T \in \mathcal{R}^n$. $\check{x}_i$ and $\hat{x}_i$ respectively indicate lower and upper bounds on variable x_i. The functions $f_j(\boldsymbol{x})$, $j = 0, 1, 2, \cdots, m$ are assumed to be (at least) once continuously differentiable.

Nonlinear problem $\mathrm{P_{NLP}}$ may be solved using any number of techniques; many of which rely on SAO when the objective and/or constraint functions require computationally demanding simulations. Examples include finite element (FE) and computational fluid dynamics (CFD) analyses. In structural optimization, SAO algorithms based on convex separable approximations are often used. For large-scale applications with many design variables it is important that the required gradients $\partial f_0/\partial x_i$ and $\partial f_j/\partial x_i$ can efficiently and accurately be calculated; see the review paper by van Keulen et al. (2005) for a fairly recent perspective on this issue.

3 Sequential approximate optimization

Sequential approximate optimization as a solution strategy for problem $\mathrm{P_{NLP}}$ seeks to construct successive approximate analytical subproblems $P[k]$, $k = 1, 2, 3, \cdots$ at successive approximations $\boldsymbol{x}^{\{k\}}$ to the solution $\boldsymbol{x}^*$. The solution of subproblem $P[k]$ is $\boldsymbol{x}^{\{k*\}} \in \mathcal{R}^n$, to be obtained using any suitable continuous programming method. Thereafter, $\boldsymbol{x}^{\{k+1\}} = \boldsymbol{x}^{\{k*\}}$, the minimizer of subproblem $P[k]$. This is repeatedly done, until either the sequence of iterates $\boldsymbol{x}^{\{k*\}}$, $k = 1, 2, \ldots$ converges, or until some maximum number of iterations $\hat{k}$ have passed. Herein, we will restrict ourselves to gradient-based SAO methods using convex, separable approximations.

3.1 The approximate primal subproblem for problem $\mathrm{P_{NLP}}$

A suitable approximate continuous primal subproblem $P_P[k]$, constructed at $\boldsymbol{x}^{\{k\}}$, is

Primal approximate subproblem $P_P[k]$

$$\begin{aligned} &\min \quad \tilde{f}_0(\boldsymbol{x}) \\ &\text{subject to} \quad \tilde{f}_j(\boldsymbol{x}) \le 0, \qquad j = 1, 2, \cdots, m, \\ &\qquad\qquad \check{x}_i \le x_i \le \hat{x}_i, \qquad i = 1, 2, \cdots, n, \end{aligned} \tag{2}$$

where the $\tilde{f}_j$, $j = 0, 1, 2, \cdots, m$ are convex separable analytical approximation functions to the original functions f_j. This primal problem has n unknowns, m constraints, and $2n$ side or bound constraints (when no slack or relaxation variables are introduced). Note that the bounds $\check{\boldsymbol{x}}$, $\hat{\boldsymbol{x}}$ may trivially be replaced by some move limit or trust region expressed in terms of the infinity norm.

3.2 An approximate dual subproblem for problem $\mathrm{P_{NLP}}$

Convex separable approximations provide opportunities for the use of efficient solvers for primal approximate subproblem $P_P[k]$. Very well-known are the dual approaches (see the references mentioned in Section 1), but alternatives have certainly been proposed (again see Zillober 2001). In the dual approach, $P_P[k]$ is replaced by some dual approximate subproblem $P_D[k]$, which is particularly attractive if the Falk dual (Falk 1967) can be invoked[2] and the approximations also have a relatively simple structure; the latter condition allows for *analytical relationships* between the primal and the dual variables, while the simple bounds on the primal variables $\boldsymbol{x}$ in the Falk dual become *part of the dual*. We start by defining the approximate dual function $\tilde{\gamma}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$, where $\boldsymbol{\lambda} \in \mathcal{R}^m$ represents the dual variables, and $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ the Lagrangian. Then, $\tilde{\gamma}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}} \{\tilde{f}_0(\boldsymbol{x}) + \sum_{j=1}^m \lambda_j \tilde{f}_j(\boldsymbol{x})\}$. The minimizer of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ for $\boldsymbol{\lambda}$ given will be denoted by $\boldsymbol{x}(\boldsymbol{\lambda})$, which allows for the formulation of the dual approximate subproblem as

Dual approximate subproblem $P_D[k]$

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} \left\{ \tilde{\gamma}(\boldsymbol{\lambda}) = \tilde{f}_0(\boldsymbol{x}(\boldsymbol{\lambda})) + \sum_{j=1}^m \lambda_j \tilde{f}_j(\boldsymbol{x}(\boldsymbol{\lambda})) \right\}, \\ &\text{subject to } \lambda_j \ge 0, \qquad j = 1, 2, \cdots, m. \end{aligned} \tag{3}$$

The saddle point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is then found by maximizing dual problem (3) with respect to $\boldsymbol{\lambda}$; as said, the primal variables $\boldsymbol{x}$ are obtained via analytical relationships from the dual variables $\boldsymbol{\lambda}$. This bound constrained problem requires the determination of the m unknowns λ_j only, subject to m very simple non-negativity constraints. For details, the reader is referred to Falk (1967), Fleury (1979) and Groenwold and Etman (2008b). Suffice it to say that strict convexity of the subproblems is sufficient—but not necessary, e.g. see Wood and Groenwold (2009)—to ensure that the solutions of dual

[2] Strict convexity of the subproblems is a sufficient, but not a necessary requirement, for invoking the Falk dual.

approximate subproblems $P_D[k]$ coincide with the solutions of primal approximate subproblems $P_P[k]$. Note that the bound constraints on $\check{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}$ are accounted for as part of the dual problem and do not introduce additional dual variables.

3.3 Practical considerations

We will herein not prescribe whether the primal or the dual subproblems (or even both, for that matter) are used. If the primal approximate subproblems are strictly convex, the primal and dual statements are identical. (In presenting numerical results, we will however use the dual approach, for no particular reason.)

Neither will we be prescriptive regarding the treatment of infeasible subproblems, for neither the primal nor the dual approach. Among others, suitable methods include relaxation, e.g. see Svanberg (2002), and normal-tangential approaches, e.g. see Pérez et al. (2004). These approaches are of particular importance in the primal approach. In theory, dual solvers are not affected by infeasible subproblems at all; they merely require the solution of a simple non-negativity constrained subproblem. For feasible constraints, the associated dual variables will either be zero, or relate to a turning point of the dual (if the constraint is active). For infeasible subproblems, the principle of monotonicity (Papalambros and Wilde 2000) may be invoked to show that the associated dual variables will go to infinity. Whereas this is not necessarily problematic from the primal-dual relationship point of view, it is nevertheless customary to relax the dual subproblems. As an alternative, a bounded form of dual problem (3) may be constructed, e.g. see Wood et al. (2009).

4 Algorithms

We will now develop variants of an SAO algorithm that use different termination criteria. We will do so by presenting a basic algorithmic framework, denoted algorithm SAO-θ, whereafter four variants will be developed which fit into the framework; this approach rather nicely illustrates how the various algorithmic variants are interrelated. (In presenting the four strategies, we are not trying to be exhaustive.)

1. The first variant is unconditional acceptance of each iterate. This is done in many popular SAO algorithms, such as CONLIN (Fleury and Braibant 1986) and MMA (Svanberg 1987), but also in many optimality criteria based optimization algorithms (see Groenwold and Etman 2008a, and the references mentioned therein).
2. The second variant only conditionally accepts new iterates; conservatism of the objective and all the constraint function approximations is employed to enforce global convergence. Conservative convex separable approximations (CCSA) were proposed by Svanberg (2002), and shown to render algorithms globally convergent.
3. The third variant is conditional acceptance based on a nonlinear acceptance filter. The filter was developed by Fletcher and co-workers, e.g. see Fletcher and Leyffer (1998, 2002), Fletcher et al. (2002a, b, 2005).

   Fletcher et al. (1998) present a proof of global convergence for a trust-region filter-SLP method, whereas Fletcher et al. (2005) present the same for a filter-SQP method. Herein, we replace the QP subproblem by a nonlinear but strictly convex separable subproblem for which an effective subproblem based on the dual of Falk can be derived. We then investigate whether the filter mechanism provides an efficient mechanism for conditional acceptance of iterates in sequential convex separable optimization, even though we depart slightly from the algorithm for which proof of global convergence was developed by Fletcher and his co-workers.
4. The final variant combines conditional acceptance on the basis of conservatism and the filter, in the hope of exploiting the salient features of both.

We assume convex and separable approximations (although the filter-acceptance termination concept does not require such an assumption). We specialize the approximations used in this paper by restricting ourselves to diagonal quadratic approximation (7) given in Appendix B, for which the conservatism can be increased by simply increasing the approximate curvatures $c_{2i_j}^{\{k\}}$.

What is more, we restrict ourselves to a spherical quadratic instance of (7); we do so *merely for the sake of brevity*. It may be advantageous to the accuracy of the approximation if diagonal quadratic approximation (7) presented in Appendix B is generalized to provide for problem-specific behavior, e.g. see Groenwold et al. (2010). This is very easily done, and merely requires that the approximate curvatures $c_{2i_j}^{\{k\}}$ are estimated using a different strategy. The approximation itself remains unchanged.

However, many other convex separable approximations may be used in the general framework we develop in the following, like MMA type approximations, etc.

4.1 Algorithm SAO-θ

Let k represent an outer iteration counter, l an inner iteration counter, and $\mathcal{T}$ some termination strategy that will generate a sequence of iterates $\boldsymbol{x}^{\{k\}}$, $k = 1, 2, \cdots$. We will elaborate on the four aforementioned possible strategies $\mathcal{T}$ in sections to come.

Now assume that suitable approximation functions $\tilde{f}_j(\boldsymbol{x})$, $j = 0, 1, 2, \cdots, m$, and a suitable termination

strategy $\mathcal{T}$ have indeed been selected. Then, given an initial point $\boldsymbol{x}^{\{k\}} = \boldsymbol{x}^{\{0\}}$, an SAO algorithm may proceed as follows (using a FORTRAN-like pseudo-language):

1. **Initialization:** Set $k := 0$, $l := 0$. Select positive constant ϵ_x, and any other constants and parameters required for the approximations $\tilde{f}_j(\boldsymbol{x})$ and the selected termination strategy $\mathcal{T}$.
2. **Simulation and sensitivity analysis:** Compute $f_j(\boldsymbol{x}^{\{0\}})$, $\nabla f_j(\boldsymbol{x}^{\{0\}})$, $j = 0, 1, 2, \cdots, m$.
3. **Construct the approximations:** Reinitialize outer-loop specific parameters, and then construct the approximate functions $\tilde{f}_j(\boldsymbol{x})$ at $\boldsymbol{x}^{\{k\}}$, $j = 0, 1, 2, \cdots m$.
4. **Approximate optimization:** Construct local approximate subproblem $P_P[k]$ or $P_D[k]$. Solve the subproblem to arrive at $(\boldsymbol{x}^{\{k*\}}, \boldsymbol{\lambda}^{\{k*\}})$.
5. **Simulation analysis:** Compute $f_j(\boldsymbol{x}^{\{k*\}})$, $j = 0, 1, 2, \cdots, m$.
6. **(Optionally), test if $\boldsymbol{x}^{\{k*\}}$ is acceptable.** GOTO Step 8 if $\boldsymbol{x}^{\{k*\}}$ is indeed acceptable.
7. **(Optionally), effect an inner loop if $\boldsymbol{x}^{\{k*\}}$ is not acceptable:**
   (a) Apply termination strategy $\mathcal{T}$.
   (b) Set $l := l + 1$ and GOTO Step 4.
8. **Move to the new iterate:** Set $\boldsymbol{x}^{\{k+1\}} := \boldsymbol{x}^{\{k*\}}$.
9. **Convergence test:** IF $\|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| \leq \epsilon_x$, STOP.
10. **Simulation sensitivity analysis:** Compute $\nabla f_j(\boldsymbol{x}^{\{k+1\}})$, $j = 0, 1, 2, \cdots, m$.
11. **Initiate an additional outer loop:** Set $k := k + 1$ and GOTO Step 3.

It is more precise to use the notation $\boldsymbol{x}^{\{k,l\}}$ rather than $\boldsymbol{x}^{\{k\}}$. The latter however is retained for the sake of brevity, and the meaning, at least, is clear. In addition, the move limit strategy mentioned in the foregoing is easily included in the algorithm outlined above.

Next, we present the different SAO implementations considered or developed herein in the following subsections.

4.2 Algorithm SAO-A (rudimentary; non-terminating)

We start with the most basic SAO sequence possible, in which each iterate is accepted unconditionally.

Accordingly, Steps 6 and 7 in algorithm SAO-θ are ignored. Viz., we have:

6. Void.
7. Void.

The resulting algorithm is not guaranteed to converge or even terminate. Nevertheless, it should be acknowledged that algorithm SAO-θ is representative of most SAO algorithms used in practice, since

1. there is no (potential) cost associated with enforcing convergence, and
2. the approximation functions $\tilde{f}_j(\boldsymbol{x})$ are normally selected to be representative of the true functions $f_j(\boldsymbol{x})$, to the extent that termination strategies are not even required.

Nevertheless, the use of algorithm SAO-A remains risky.

The SAO algorithms we present in the following subsections are all based on conditional acceptance of iterates with the aim to (ultimately) provide a framework for global convergence. At this stage only algorithm SAO-B has a proof of convergence; algorithms SAO-C and SAO-D are new variants which are investigated in an effort to seek for conditional acceptance mechanisms with minimal computational overhead associated with enforcing global convergence.

4.3 Algorithm SAO-B (conservative; terminating)

Before we present the conservative variant of algorithm SAO-θ, we briefly reiterate the machinery of a conservative SAO algorithm. However, the interested reader should also read the paper of Svanberg (2002).

In conservative methods, an outer iteration starts from the current iterate $\boldsymbol{x}^{\{k\}}$ and yields a new iterate $\boldsymbol{x}^{\{k*\}}$. If the approximate objective and constraint functions $\tilde{f}_j(\boldsymbol{x}^{\{k*\}})$, $j = 0, 1, 2, \cdots, m$ are all greater than, or equal to, the original functions $f_j(\boldsymbol{x}^{\{k*\}})$, the approximating functions are denoted *conservative*. (Conservatism does *not* imply that the feasible set of the subproblem is completely contained in the original feasible set, but it does imply that the optimal solution of the subproblem is a feasible solution of the original problem, with lower objective value than the previous iterate.)

However, rather than enforcing conservatism unconditionally as done by Svanberg, we enforce conservatism only if a feasible descent step cannot be made; we denoted this strategy 'relaxed conservatism', e.g. see Groenwold et al. (2009). (A similar strategy may be used when an algorithm is not in the feasible regime.)

If an iterate $\boldsymbol{x}^{\{k*\}}$ does not result in a *feasible descent* step, a sequence of inner iterations is initiated, until either a feasible descent step is indeed obtained, or if conservatism is obtained for all the approximating functions $\tilde{f}_j(\boldsymbol{x})$, $j = 0, 1, 2, \cdots, m$. Clearly, this is always possible if the original functions $f_j(\boldsymbol{x})$ are well behaved (and assuming that the conservatism of the approximating functions $\tilde{f}_j(\boldsymbol{x})$ is indeed variable).

Accordingly, in conservative algorithm SAO-B, Steps 6 and 7 in algorithm SAO-θ are replaced by the following steps:

6. **Test if $\boldsymbol{x}^{\{k*\}}$ is acceptable:**
   (a) **test if $\boldsymbol{x}^{\{k*\}}$ represents a feasible descent step:** IF $f_0(\boldsymbol{x}^{\{k*\}}) < f_0(\boldsymbol{x}^{\{(k-1)*\}})$ for $k > 0$, AND $\max\{f_j(\boldsymbol{x}^{\{k*\}})\} \leq 0$, $j = 1, 2, \cdots, m$, GOTO Step 8,
   (b) **test if $\boldsymbol{x}^{\{k*\}}$ represents a feasible and conservative step:** IF $\tilde{f}_0(\boldsymbol{x}^{\{k*\}}) \geq (f_0(\boldsymbol{x}^{\{k*\}}) - \epsilon_1)$, AND $\tilde{f}_j(\boldsymbol{x}^{\{k*\}}) \geq (f_j(\boldsymbol{x}^{\{k*\}}) - \epsilon_2)$, $j = 1, 2, \cdots, m$, GOTO Step 8.
7. **Effect an inner loop if $\boldsymbol{x}^{\{k*\}}$ is not acceptable:**
   (a) Set $l := l + 1$.
   (b) IF $\tilde{f}_0(\boldsymbol{x}^{\{k*\}}) < (f_0(\boldsymbol{x}^{\{k*\}}) - \epsilon_1)$, set $c_{2i_0}^{\{k\}} := \chi_1 c_{2i_0}^{\{k\}}$.
   (c) IF $\tilde{f}_j(\boldsymbol{x}^{\{k*\}}) < (f_j(\boldsymbol{x}^{\{k*\}}) - \epsilon_2)$, set $c_{2i_j}^{\{k\}} := \chi_2 c_{2i_j}^{\{k\}}$, $j = 1, 2, \cdots, m$.
   (d) GOTO Step 4.

In addition, it is required to specify the constants $\epsilon_1, \epsilon_2 \geq 0$, and $\chi_1, \chi_2 > 1$ in Step 1 in algorithm SAO-θ.

Svanberg (2002) has demonstrated that algorithms based on conservative convex separable approximations (CCSA) are globally convergent, but the conservatism requirements may sometimes actually be relaxed, e.g. see Groenwold et al. (2009).

4.4 Algorithm SAO-C (filtered trust-region; terminating)

Before we present SAO-C, it is necessary to briefly introduce the notion of a filter; we will do so following closely the presentation of Fletcher et al. (2002b). The interested reader should however also read the cited literature. In essence, the flter constructs a Pareto front of the competing objectives of minimizing the objective function and the infeasibility.

For the sake of brevity and consistency with the literature, let $f = f_0$, and $g_j = f_j$, $j = 1, 2, \cdots, m$. Then, a nonlinear acceptance filter consists of a list of pairs $\{(h, f)\}$, evaluated for a list of points $\{\boldsymbol{x}\}$, with $h = \max(0, g_j)$ being the maximum constraint violation, and f the objective function value for each point. The pair $(h^{\{i\}}, f^{\{i\}})$, obtained for iteration i, is said to dominate another pair $(h^{\{j\}}, f^{\{j\}})$, if and only if, both $h^{\{i\}} \leq h^{\{j\}}$ and $f^{\{i\}} \leq f^{\{j\}}$. Fletcher et al. then define the NLP filter as a list of pairs $\{(h^{\{i\}}, f^{\{i\}})\}$, such that no pair dominates any other. The list of pairs is sorted such that the $f^{\{i\}}$ are monotonically decreasing.

A point $\boldsymbol{x}$ is said to be 'acceptable for inclusion in the filter' if its pair (h, f) is not dominated by any entry in the filter, implying that either $h < h^{\{j\}}$ or $f < f^{\{j\}}$ for all elements j in the filter list. If we wish to 'include point $\boldsymbol{x}$ in the filter' it is understood that its pair (h, f) is added to the list of pairs in the filter, and all pairs in the filter that are dominated by the new pair, are removed.

To prove convergence, Fletcher et al. (2002b) define a small envelope around the current filter, which provides a mechanism to force the sequence of iterates towards feasibility. The resulting 'slanting envelope test' then becomes: a pair (h, f) is acceptable if

$$\text{either} \qquad h \leq \beta h^{\{j\}} \qquad \text{or} \qquad f + \gamma h \leq f^{\{j\}}, \tag{4}$$

for all j, where β and γ are preset parameters such that $1 > \beta > \gamma > 0$, with β close to 1 and γ close to zero. Typical values used by Fletcher, Leyffer, and Toint are $\gamma = 10^{-5}$ and $\beta = 1 - \gamma$.

Now, let $\Delta f = f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^{(k*)})$, and $\Delta q = \tilde{f}(\boldsymbol{x}^{(k)}) - \tilde{f}(\boldsymbol{x}^{(k*)})$. The solution to the subproblem yields a trial step from $\boldsymbol{x}^{\{k\}}$ to point $\boldsymbol{x}^{(k*)}$, denoted $\boldsymbol{d}$. Fletcher et al. (2002b) regard a step $\boldsymbol{d}$ that satisfies $\Delta q > 0$ to be an f-type step. If the trial step is accepted then an f-type iteration is said to have occurred. An h-type iteration is said to have occurred if the current subproblem is incompatible (the subproblem has no feasible solution) or if a step is taken for which $\Delta q \leq 0$ holds. f-type iterations have the primary aim to reduce the objective function, possibly following an increase in h. h-type iterations have the primary aim to reduce h, possibly allowing an increase in f. The filter defines what increase in either f or h is actually allowed in the current iteration stage.

The filter builds up a list of pairs corresponding to iterates from which the algorithm had to 'retreat'. We therefore include point $\boldsymbol{x}^{\{k\}}$ in the filter at the end of the iteration if, and only if, that iteration is an h-type iteration. This implies that, at the start of iteration k, the pair $(h^{\{k\}}, f^{\{k\}})$ is *not* in the current filter $\mathcal{F}^{\{k\}}$, but the pair $(h^{\{k\}}, f^{\{k\}})$ must be acceptable to the current filter $\mathcal{F}^{\{k\}}$. For this reason, we test whether $\boldsymbol{x}^{(k*)}$ is acceptable to the current filter $\mathcal{F}^{\{k\}}$ and $\boldsymbol{x}^k$, so that if $(h^{\{k\}}, f^{\{k\}})$ is subsequently entered into the filter, then $(h^{(k+1)}, f^{(k+1)})$ (which corresponded to $\boldsymbol{x}^{(k*)}$) will still be acceptable to the new filter.

A condition for a step $\boldsymbol{d}$ to give rise to an f-type iteration is that both $\Delta f \geq \sigma \Delta q$ and $\Delta q > 0$ are satisfied, see Fletcher et al. (2002b). This means that besides the required predicted decrease in the objective function, there is an additional requirement on a sufficient reduction of the

actual objective function value. This condition gives rise to the following test: if

$$\Delta f < \sigma \Delta q \text{ and } \Delta q > 0, \tag{5}$$

then the predicted improvement by $\boldsymbol{x}^{(k*)}$ is not satisfactory. Note that the 'and' statement implies that the test is passed if $\Delta q \leq 0$. Then, if $\boldsymbol{x}^{(k*)}$ passed the filter acceptance test, an h-type iteration is recognized and $\boldsymbol{x}^{\{k\}}$ is included in the filter.

An inner loop iteration arises if $\boldsymbol{x}^{\{k*\}}$ is not acceptable to the current filter $\mathcal{F}^{\{k\}}$ and $\boldsymbol{x}^k$, or if sufficient reduction criterion (5) is not met (a typical value for $\sigma = 0.1$). Then the approximate subproblem is re-solved with a reduced trust region.

Once accepted, we move to the new iterate, and the trust region Δ_∞ is updated. In principle, any strategy to reset the trust region $\Delta_\infty^{\{k+1\}}$ is allowed, as long as $\Delta_\infty > \Delta_{\infty_0}$, with $\Delta_{\infty_0} > 0$ a preset parameter. Hence any (heuristic) move limit strategy may be included here.

Indeed, it may be stated that if an infinite sequence of pairs (h, f) is accepted by the filter, then $h \to 0$, that is, convergence to a feasible point results. This is the result of Lemma 1 and its Corollary presented by Fletcher et al. (2005). Thereafter, the sufficient descent trust region condition present in the filter may be used to enforce descent and eventually, termination.

We explicitly point out that the subproblems considered in this paper differ (slightly) from the QP subproblems studied by Fletcher et al. (2005). (Since we use spherical quadratic approximations, our constraints are quadratic rather than linear, and we use only approximate Hessian information. In addition, we use a dual statement, although QP-like subproblems are also possible, e.g. see Etman et al. (2009).) Although this does *not* affect Lemma 1 and the Corollary of Fletcher et al. (2005)—which guarantees convergence to a feasible point—a formal proof for algorithm SAO-C is still to be developed. Nevertheless, algorithm SAO-C is very closely related to the filter-SQP algorithm of Fletcher et al. (2005). Indeed, given Lemma 1 and its Corollary, it merely remains to demonstrate convergence given some feasible starting point using a trust-region strategy (or conservatism, for that matter).

Accordingly, in filtered trust-region algorithm SAO-C, Steps 6 and 7 in algorithm SAO-θ are replaced by the following steps:

6. **Test if $\boldsymbol{x}^{\{k*\}}$ is acceptable:**
   (a) **test if $\boldsymbol{x}^{\{k*\}}$ is acceptable to the current filter $\mathcal{F}^{\{k\}}$ and $\boldsymbol{x}^{\{k\}}$:** IF no, GOTO Step 7,
   (b) **test if $\Delta f < \sigma \Delta q$ AND $\Delta q > 0$:** IF yes, GOTO Step 7,
   (c) **conditionally update the filter and reinitialize the outer loop:** Set $\Delta_\infty^{\{k\}} > \Delta_{\infty_0}^{\{k\}}$, and GOTO Step 8.
7. **Effect an inner loop if $\boldsymbol{x}^{\{k*\}}$ is not acceptable:**
   (a) set $\Delta_\infty^{\{k\}} = \Delta_\infty^{\{k\}} / \chi_3$
   (b) GOTO Step 4.

In addition, it is required to specify $\Delta_{\infty_0}, \gamma, \sigma, \beta > 0$, and $\chi_3 > 1$ in Step 1 in algorithm SAO-θ.

4.5 Algorithm SAO-D (filtered conservatism; terminating)

In algorithm SAO-D, we combined the salient features of algorithms SAO-B and SAO-C: the filter is used to decide if an iterate is acceptable (since the filter is more tolerant than enforcing conservatism). Then, if an iterate is unacceptable, we enforce conservatism of only those approximations that are non-conservative (since reduction of the trust region would unnecessarily hurt the conservative approximations).

Accordingly, in algorithm SAO-D, Steps 6 and 7 in algorithm SAO-θ are replaced by the following steps:

6. **Test if $\boldsymbol{x}^{\{k*\}}$ is acceptable:**
   (a) **test if $\boldsymbol{x}^{\{k*\}}$ is acceptable to the current filter $\mathcal{F}^{\{k\}}$ and $\boldsymbol{x}^{\{k\}}$:** IF no, GOTO Step 7,
   (b) **test if $\Delta f < \sigma \Delta q$ AND $\Delta q > 0$:** IF yes, GOTO Step 7,
   (c) **conditionally update the filter and reinitialize the outer loop:** Set $\Delta_\infty > \Delta_{\infty_0}$, and GOTO Step 8.
7. **Effect an inner loop if $\boldsymbol{x}^{\{k*\}}$ is not acceptable:**
   (a) Set $l := l + 1$.
   (b) IF $\tilde{f}_0(\boldsymbol{x}^{\{k*\}}) < (f_0(\boldsymbol{x}^{\{k*\}}) - \epsilon_1)$, set $c_{2i_0}^{\{k\}} := \chi_1 c_{2i_0}^{\{k\}}$.
   (c) IF $\tilde{f}_j(\boldsymbol{x}^{\{k*\}}) < (f_j(\boldsymbol{x}^{\{k*\}}) - \epsilon_2)$, set $c_{2i_j}^{\{k\}} := \chi_2 c_{2i_j}^{\{k\}}$, $j = 1, 2, \cdots, m$.
   (d) GOTO Step 4.

In addition, it is required to specify $\Delta_{\infty_0}, \gamma, \sigma, \beta, \chi_3$, $\epsilon_1, \epsilon_2 > 0$, and $\chi_1, \chi_2 > 1$ in Step 1 in algorithm SAO-θ.

Termination based on filtered-conservatism assumes that convergence follows from the (non-related) Lemma 1 and its Corollary in Fletcher et al. (2005), and conservatism as presented by Svanberg (2002). It is required to show that conservatism has the same effect as a trust region in generating points that are acceptable to the filter, which seems like a reasonable proposition.

5 Numerical experiments

We now present numerical results for the four algorithms we have implemented.

Again, we note that Algorithms SAO-C and SAO-D lack a rigorous, formal proof of convergence, and it may even be considered unfair by some to compare the computational performance of algorithms with and without proof of global convergence.

In our numerical studies (including many not reported herein), we have never encountered unpleasant surprises with Algorithms SAO-C and SAO-D (barring cases when the subproblem solvers themselves failed). In the following results, there is one instance for which unconditional acceptance of the iterates (Algorithm SAO-A) fails. The numerical results for Algorithm SAO-A are merely presented to give an impression of the expense associated with enforcing global convergence using the three conditional acceptance mechanisms.

Let k^* and l^* respectively represent the number of outer and inner iterations required for convergence, and we introduce $\bar{h} = \max(f_j)$, $j = 1, 2, ...m$ (not to be confused with $h = \max(0, f_j)$ in the filter).

Unless otherwise stated, we have set the move limit (respectively the initial trust region 'diameter') $\delta_{i_\infty}^{\{k\}} \leftarrow \Delta_\infty^{\{k\}}(\hat{x}_i - \check{x}_i)$, with $\Delta_\infty^{\{k\}} = 1.0$. This is far from optimal for algorithms A, B and D, but it does reveal the (lack of) robustness of these algorithms. (In practice, a far better choice would be to use say $\Delta_\infty^{\{k\}} = 0.2\ \forall\ k$.) Throughout, we have used $\epsilon_1 = \epsilon_2 = 10^{-7}$, and $\chi_1 = \chi_2 = 2$ in the conservative variants of the algorithms. (We have not experimented with these values to see if they are even close to being optimal.) In the trust region variants, we have used $\Delta_{\infty_0} = 1 \times 10^{-8}$, $\gamma = 1 \times 10^{-7}$, $\sigma = 1 \times 10^{-6}$, $\beta = 1 - \gamma$, and $\chi_3 = 2$. (Again, we have not experimented with these values to see if they are optimal.)

The test problems used are tabulated in Table 1, and numerical results are presented in Table 2. The test problems are also given explicitly in Appendix A, while the spherical quadratic approximation used is given in Appendix B. Unless otherwise stated, the algorithms are all stopped when $\|\boldsymbol{x}^{\{(k-1)*\}} - \boldsymbol{x}^{\{k*\}}\| \leq \epsilon_x$, with ϵ_x given in Table 1.

Even though we have used a small set of (fairly demanding) test problems, the results corroborate the potential computational expense associated with the enforcement of convergence via conservatism and a trust region strategy (when compared to our non-terminating algorithm SAO-A). For obvious reasons, this explains the popularity of algorithm SAO-A in simulation-based optimization: rather than using a costly convergence strategy, it is hoped that 'the approximations themselves will do the work', and effect termination. Nevertheless, the results presented for Problem 6 (Fleury's weight-minimization-like problem) do illustrate how dangerous this may sometimes be. What is more: for some problems, not enforcing convergence may actually be more expensive than enforcing convergence using filtered conservatism, see Problem 8 (Vanderplaats' cantilever beam), although this is not expected *if the approximations used are reasonably accurate*. (The spherical quadratic approximation we use does not exploit any advantage whatsoever that may derive from the use of intervening variables, while for Vanderplaats' cantilever beam this would be particularly advantageous. For an elaboration, see Groenwold and Etman (2009).)

It is interesting to note that the trust region strategy is superior to the conservative strategy for some problems (Problems 3, 7b and 8), and vice versa for some others (Problems 1, 4, 5, and 7a). For Problem 8 (Vanderplaats' cantilever beam) and the approximations used, conservatism is not even a viable method for termination; it is simply too expensive. For the remaining problems, there is little difference between the two methods. The filtered conservative algorithm however is on average superior to the aforementioned strategies, and in particular so for Problems 4 and 8. What is more: the computational overhead of filtered conservatism compared to the non-terminating variant (algorithm SAO-A) is very small.

Table 1 The test problems; see Appendix A for details

No.	Problem	n	m	ϵ_x	$\Delta_\infty^{\{0\}}$
1	Rosenbrock's banana valley	2	0	1×10^{-5}	1.0
2	2-Bar shape and size design problem	2	2	1×10^{-5}	1.0
3	Convex 10-bar truss	10	25	1×10^{-5}	1.0
4	12-Corner polytope	21	1	5×10^{-4}	1.0
5	Snake problem	30	41	1×10^{-5}	1.0
6	Fleury's weight-minimization-like problem	1,000	2	1×10^{-5}	0.01
7a	First large nonconvex problem	1,000	2	5×10^{-4}	1.0
7b	Second large nonconvex problem	1,000	2	5×10^{-4}	1.0
8	Vanderplaats' cantilever beam	20 and 200	21 resp. 201	1×10^{-5}	1.0

Table 2 Numerical results for the test problems—see Section 4 for a description of the algorithms

Problem no.	Algorithm	k^*	l^*	f_0^*	$\bar{h}^*$
1	Rudimentary (SAO-A)	4, 076	–	3.597×10^{-05}	–
	Conservatism (SAO-B)	45	53	2.573×10^{-06}	–
	Filtered trust region (SAO-C)	174	810	2.584×10^{-05}	–
	Filtered conservatism (SAO-D)	45	53	2.573×10^{-06}	–
2	Rudimentary (SAO-A)	9	–	1.508652	1.978×10^{-12}
	Conservatism (SAO-B)	6	5	1.508652	1.157×10^{-10}
	Filtered trust region (SAO-C)	9	0	1.508652	1.978×10^{-12}
	Filtered conservatism (SAO-D)	9	0	1.508652	1.978×10^{-12}
3	Rudimentary (SAO-A)	35	–	4,193.194	-1.165×10^{-06}
	Conservatism (SAO-B)	35	12	4,193.192	-2.044×10^{-07}
	Filtered trust region (SAO-C)	35	0	4,193.190	-1.165×10^{-06}
	Filtered conservatism (SAO-D)	35	0	4,193.189	-1.165×10^{-06}
4	Rudimentary (SAO-A)	225	–	−279.9038	1.157×10^{-07}
	Conservatism (SAO-B)	328	176	−279.9038	2.651×10^{-07}
	Filtered trust region (SAO-C)	413	642	−279.9038	-4.025×10^{-07}
	Filtered conservatism (SAO-D)	169	19	−279.9038	4.712×10^{-07}
5	Rudimentary (SAO-A)	183	–	−10.02298	2.294×10^{-06}
	Conservatism (SAO-B)	51	352	−10.02298	2.298×10^{-06}
	Filtered trust region (SAO-C)	183	0	−10.02298	2.294×10^{-06}
	Filtered conservatism (SAO-D)	183	0	−10.02298	2.294×10^{-06}
6	Rudimentary (SAO-A)	*[a]	–	0.001000	$9.500 \times 10^{+09}$
	Conservatism (SAO-B)	40	51	950.0001	-3.85×10^{-06}
	Filtered trust region (SAO-C)	25	39	950.0001	6.821×10^{-11}
	Filtered conservatism (SAO-D)	35	23	950.0001	6.544×10^{-08}
7a	Rudimentary (SAO-A)	176	–	260.8520	-3.145×10^{-08}
	Conservatism (SAO-B)	173	14	260.8520	-1.033×10^{-07}
	Filtered trust region (SAO-C)	191	208	260.8520	-2.694×10^{-07}
	Filtered conservatism (SAO-D)	173	14	260.8520	-1.033×10^{-07}
7b	Rudimentary (SAO-A)	305	–	−739.1479	-3.928×10^{-07}
	Conservatism (SAO-B)	361	102	−739.1479	-3.241×10^{-07}
	Filtered trust region (SAO-C)	305	0	−739.1479	-3.928×10^{-07}
	Filtered conservatism (SAO-D)	305	0	−739.1479	-3.928×10^{-07}
8 ($n = 20$)	Rudimentary (SAO-A)	38	–	64,244.83	1.680×10^{-06}
	Conservatism (SAO-B)	101	261	64,244.83	1.111×10^{-06}
	Filtered trust region (SAO-C)	40	94	64,244.83	3.208×10^{-06}
	Filtered conservatism (SAO-D)	29	2	64,244.83	2.414×10^{-05}
8 ($n = 200$)	Rudimentary (SAO-A)	34	–	63,678.10	1.783×10^{-06}
	Conservatism (SAO-B)	457	2, 535	63,678.10	4.896×10^{-06}
	Filtered trust region (SAO-C)	30	25	63,678.10	2.983×10^{-07}
	Filtered conservatism (SAO-D)	29	1	63,678.10	4.058×10^{-05}

[a]Did not converge; all the variables ended up on the lower bounds after 11 iterations

Finally, we reiterate that we have merely for the sake of brevity restricted ourselves to using the spherical quadratic instance of (7), again see Appendix B. In practice, we suggest that diagonal quadratic approximation (7) is generalized to provide for problem-specific behavior, e.g. see Groenwold et al. (2010); this may be expected to result in notably reduced computational effort (in that the required number of iterations is decreased). As said, this is very easily done, and merely requires that the approximate curvatures $c_{2i_j}^{\{k\}}$ are estimated using a different strategy. We emphasize that the diagonal quadratic approximation itself remains unchanged.

6 Conclusions

We have studied the convergence and termination of sequential approximate optimization (SAO) algorithms using a trust region with filtered acceptance of the iterates, and conservatism for *inequality constrained simulation-based* optimization.

We have also proposed a new strategy for convergence and termination, denoted *filtered conservatism*, in which the acceptance or rejection of an iterate is determined using a nonlinear acceptance filter. If an iterate is rejected, the conservatism of all the unconservative approximations are increased, rather than reducing the trust region. Although we have performed only limited testing, the numerical results corroborate the accuracy and robustness of the proposed method. Apparently, the advantages of the filtered conservative strategy above classical conservatism may be attributed to the fact that the former strategy requires neither strict conservatism, nor descent before an iterate may be accepted. The iterate need merely be acceptable to the filter.

In turn, the advantages of the filtered conservative strategy above the classical trust region method with nonlinear acceptance filter apparently are that the reduced trust region in the latter is hurtful not only for unconservative approximations, but also for conservative approximations, which were perfectly acceptable at the large trust region to start with.

At the same time, we have demonstrated that enforcing convergence may sometimes imply a computational penalty if classical trust-regions or conservatism are used, which possibly explains why people in structural optimization often do not bother to enforce convergence in the first place, since the function and gradient evaluations are so expensive. (Although we have not found any test problems for which filtered-conservatism was notably more expensive than the unconditional acceptance of iterates, we certainly expect that this will be true for some problems, in the spirit of a 'weak' no-free-lunch theorem.)

A note on the approximations used: the frameworks we have developed do not prescribe the approximations to be used. For generating the numerical results, we have restricted ourselves to using a spherical quadratic approximation, merely because this approximation is so simple. However, this approximation may be far from optimal; it is possible to obtain notably better results (faster convergence) using different approximations.

Finally: we have assumed that convergence follows from the (non-related) contributions in trust-region methods on the one hand, and conservatism as proposed by Svanberg on the other. The development of a unified proof of convergence, possibly requiring a modification in the implementation of filtered conservatism, seems of interest.

Appendix A: The test problems

A.1 Rosenbrock's unconstrained 2-D banana valley problem

Consider the well-known unconstrained 2-dimensional problem

$$\min f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2. \tag{6}$$

The global minimizer is $x_1^* = x_2^* = 1.0$, with $f(\boldsymbol{x}^*) = 0$. The starting point is $x_1^{\{0\}} = x_2^{\{0\}} = 0$. We use $\check{x}_1 = \check{x}_2 = -2.0$ and $\hat{x}_1 = \hat{x}_2 = 2.0$. This problem is highly coupled, and the problem is notoriously difficult for optimization methods based on separable approximations.

A.2 Simultaneous shape and sizing design of a 2-bar truss

This is a problem proposed by Svanberg (1995). It is an interesting problem in that shape and sizing design are performed simultaneously. The problem may analytically be expressed as

$$\begin{aligned}
&\min_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) = c_1 x_1 \sqrt{1 + x_2^2}, \\
&\text{subject to } f_1(\boldsymbol{x}) = c_2 \sqrt{1 + x_2^2}\left(\frac{8}{x_1} + \frac{1}{x_1 x_2}\right) - 1 \le 0, \\
&\qquad\qquad f_2(\boldsymbol{x}) = c_2 \sqrt{1 + x_2^2}\left(\frac{8}{x_1} - \frac{1}{x_1 x_2}\right) - 1 \le 0, \\
&\qquad\qquad 0.2 \le x_1 \le 4.0, \\
&\qquad\qquad 0.1 \le x_2 \le 1.6,
\end{aligned}$$

with $c_1 = 1.0$ and $c_2 = 0.124$.

A.3 The convex 10-bar truss

The reader is referred to Ringertz (1988) and many others.

A.4 The 12-corner polytope

Consider the now well-known 12-corner polytope problem in the 21 variables $(r_1, \cdots, r_{11}, v_1, \cdots, v_{10})$ proposed

by Svanberg (1995). It is required to maximize the area f_0, subject to a single constraint f_1 on the circumference. The problem was formulated by Svanberg as

$$\min_{r_i, v_i} f_0(\boldsymbol{r}, \boldsymbol{v}) = -\frac{1}{2}\sum_{i=1}^{10}\left[r_i r_{i+1}\sin(v_i)\right],$$

$$\text{subject to } f_1 = r_1 + r_{11} + \sum_{i=1}^{10}\left[r_i^2 + r_{i+1}^2 - 2r_i r_{i+1}\cos(v_i)\right]^{1/2} - 60 \le 0,$$

$$1 \le r_i \le 30, \quad i = 1, 2, \cdots, 11,$$

$$1 \le v_i \le 45, \quad i = 12, 13, \cdots, 21.$$

The starting point is taken as $r_i^0 = 11$ and $v_i^0 = 18$ for all i. The optimal solution is $f_0^* = 75(2 + \sqrt{3})$, with $r_i^* = 5\sin(15i)/\sin(15)$ and $v_i^* = 15$.

A.5 The snake problem

Next, consider the so-called 'snake problem', proposed by Svanberg (2007) for 'anyone who wants to test a new method for nonlinear optimization'.

Let d be a given positive integer, and let δ_s be a given 'small' positive real number. For $i = 1, 2, \cdots, d$, let

$$\psi_i = \frac{(3i - 2d)\pi}{6d}, \quad g_i(\boldsymbol{x}) = \frac{x_i^2 + x_{d+i}^2 - 1}{\delta_s},$$

and

$$h_i(\boldsymbol{x}) = \frac{x_{2d+i} - 2x_i x_{d+i}}{\delta_s}.$$

Then consider the following problem in the variables $\boldsymbol{x} = (x_1, \cdots, x_{3d})^T$:

$$\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) = \sum_{i=1}^{d}(x_i\cos\psi_i + x_{d+i}\sin\psi_i - 0.1x_{2d+1}),$$

$$\text{subject to } \sum_{i=1}^{d}\left(x_i^2 + x_{2d+i}^2\right) \le d,$$

$$-2 \le g_i(\boldsymbol{x}) + g_i(\boldsymbol{x})^7 \le 2, \ i = 1, 2, \cdots, d,$$

$$-2 \le h_i(\boldsymbol{x}) + h_i(\boldsymbol{x})^7 \le 2, \ i = 1, 2, \cdots, d,$$

$$-2 \le x_j \le 2, \ j = 1, 2, \cdots, 3d.$$

For a short discussion of the problem, see Svanberg (2007), who considers the problem 'rather difficult to solve' if the following feasible, but far from optimal, starting point $\boldsymbol{x}^{\{0\}}$ is chosen:

$$x_i^{\{0\}} = \cos\left(\psi_i + \frac{\pi}{12}\right), \ x_{d+i}^{\{0\}} = \sin\left(\psi_i + \frac{\pi}{12}\right),$$

and

$$x_{2d+i}^{\{0\}} = \sin\left(2\psi_i + \frac{\pi}{6}\right), \ i = 1, 2, \cdots, d.$$

We will present results for $d = 10$ (and hence, $n = 30$ and $m = 41$) using $\delta_s = 0.1$.

A.6 Fleury's weight-minimization-like problem

Consider the following test problem by Fleury (1979):

$$\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) = \sum_{i=1}^{1,000} x_i,$$

$$\text{subject to } \sum_{i=1}^{950}\frac{1}{x_i} + 10^{-6}\sum_{i=951}^{1,000}\frac{1}{x_i} - 1,000 \le 0,$$

$$\sum_{i=1}^{950}\frac{1}{x_i} - 10^{-6}\sum_{i=951}^{1,000}\frac{1}{x_i} - 900 \le 0,$$

$$10^{-6} \le x_i \le 10^6.$$

The starting point is $x_i^{\{0\}} = 10^{-5}$. The optimum is given by $x_i^* = 1$ for $i = 1, 2, \cdots, 950$ and $x_i^* = 10^{-6}$ for $i = 951, 952, \cdots, 1,000$, with $f_0(\boldsymbol{x}^*) = 950.0005$. Since the design domain is so huge, we use $\Delta_\infty^{\{0\}} = 0.01$.

A.7 Two nonconvex programming problems of high dimensionality

Now consider two non-convex problems, again proposed by Svanberg (2002). Both are expressed in terms of the symmetric, fully populated $n \times n$ matrices $\boldsymbol{S}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$, with elements given by

$$s_{ij} = \frac{2 + \sin(4\pi\vartheta_{ij})}{(1 + |i - j|)\ln(n)}, \quad p_{ij} = \frac{1 + 2\vartheta_{ij}}{(1 + |i - j|)\ln(n)}$$

and

$$q_{ij} = \frac{3 - 2\vartheta_{ij}}{(1 + |i - j|)\ln(n)},$$

where

$$\vartheta_{ij} = \frac{i + j - 2}{2n - 2} \in [0, 1] \ \forall \ i, j,$$

and $n > 1$.

The first nonconvex problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \ & f_0(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{S}\boldsymbol{x}, \\ \text{subject to } & f_1(\boldsymbol{x}) = \frac{n}{2} - \boldsymbol{x}^T \boldsymbol{P}\boldsymbol{x} \leq 0, \\ & f_2(\boldsymbol{x}) = \frac{n}{2} - \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \leq 0, \\ & -1 \leq x_i \leq 1, \end{aligned}$$

with starting point $\boldsymbol{x}^0 = (0.5, 0.5, \cdots, 0.5)^T$. The objective function $f_0(\boldsymbol{x})$ is strictly convex, but the nonlinear constraint functions $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$ are strictly concave.

The second nonconvex problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \ & f_0(\boldsymbol{x}) = -\boldsymbol{x}^T \boldsymbol{S}\boldsymbol{x}, \\ \text{subject to } & f_1(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{P}\boldsymbol{x} - \frac{n}{2} \leq 0, \\ & f_2(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} - \frac{n}{2} \leq 0, \\ & -1 \leq x_i \leq 1, \end{aligned}$$

with starting point $\boldsymbol{x}^0 = (0.25, 0.25, \cdots, 0.25)^T$. This time, the objective function $f_0(\boldsymbol{x})$ is strictly concave, while the nonlinear constraint functions $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$ are strictly convex.

A.8 Vanderplaats' cantilever beam

We consider the optimal sizing design of the tip-loaded multi-segmented cantilever beam proposed by Vanderplaats (2001)—see Example 5-1 on page 185 therein. The beam is of fixed length l, is divided into k segments, and is subject to geometric, stress and a single displacement constraint. The geometry has been chosen such that a very large number of the constraints are active or 'near-active' at the optimum. The objective function is formulated in terms of the design variables b_i and h_i as

$$\min \ f_0(\boldsymbol{b}, \boldsymbol{h}) = \sum_{i=1}^{k} b_i h_i l_i,$$

with l_i constant for given k. We enforce the bound constraints $1.0 \leq b_i \leq 80$, and $5.0 \leq h_i \leq 80$. The stress constraints are

$$\frac{\sigma(\boldsymbol{b}, \boldsymbol{h})}{\bar{\sigma}} - 1 \leq 0, \qquad i = 1, 2, \ldots k,$$

while the linear geometric constraints are written as

$$h_i - 20 b_i \leq 0, \qquad i = 1, 2, \ldots k.$$

The tip displacement constraint is

$$\frac{u_{\text{tip}}(\boldsymbol{b}, \boldsymbol{h})}{\bar{u}} - 1 \leq 0.$$

The constraints are rather easily written in terms of the design variables $\boldsymbol{b}$ and $\boldsymbol{h}$, e.g. see Vanderplaats (2001). Using consistent units, the geometric and problem data are as follows: we use a tip load of $P = 50,000$, a modulus of elasticity $E = 2 \times 10^7$, a beam length $l = 500$, while $\bar{\sigma} = 14,000$, and $\bar{u} = 2.5$. The starting point is $b_i = 5.0$ and $h_i = 60$ forall i. The problem is expressed in terms of $n = 2k$ design variables, and $m = 2k + 1$ constraints.

Appendix B: The approximations used in the numerical experiments

In principle, any (strictly) convex approximation may be used in (2), (3). Herein, we will restrict ourselves to the diagonal quadratic approximation that derives from the incomplete series expansion we have previously presented, see Groenwold et al. (2007).

What is more: we will herein restrict ourselves to a *spherical* variant of the diagonal quadratic approximation. Note that this is far from optimal for, in particular, the structural optimization examples, which are known to exhibit strong monotonicities. However, the reciprocal-like diagonal quadratic approximations we have proposed elsewhere, see Groenwold et al. (2010), Groenwold and Etman (2009), may be expected to result in a notably reduced number of outer and inner iterations (since the accuracy of the approximations used are higher).

We choose to use the approximations in dual approximate subproblem $P_D[k]$, rather than primal approximate subproblem $P_P[k]$. The reader may refer to our previous efforts, see Groenwold and Etman (2008b), Etman et al. (2009), for details.

B.1 The diagonal quadratic approximation

The approximation is given as

$$\begin{aligned} \tilde{f}_j(\boldsymbol{x}) = & f_j(\boldsymbol{x}^{\{k\}}) + \sum_{i=1}^{n} \left(\frac{\partial f_j}{\partial x_i} \right)^{\{k\}} \left(x_i - x_i^{\{k\}} \right) \\ & + \frac{1}{2} \sum_{i=1}^{n} c_{2i_j}^{\{k\}} \left(x_i - x_i^{\{k\}} \right)^2. \end{aligned} \tag{7}$$

For the sake of notational brevity, it is understood that

$$\left(\frac{\partial f}{\partial x_i} \right)^{\{k\}} = \frac{\partial f}{\partial x_i}(\boldsymbol{x}^{\{k\}}),$$

being the partial derivative of f with respect to x_i at the point $\boldsymbol{x}^{\{k\}}$. Approximation (7) is convex if $c_{2i_j}^{\{k\}} \geq 0 \; \forall \; i$, while the approximation is strictly convex if the inequality holds for all i.

Some strategies for estimating the $c_{2i_j}^{\{k\}}$ may be found in Groenwold et al. (2007). Herein, we will for the sake of brevity consider only the simplest, which results in the *spherical* quadratic approximations used by Snyman and Hay (2002): Select $c_{2i_j}^{\{k\}} \equiv c_{2_j}^{\{k\}} \forall \; i$, which requires the determination of the single unknown $c_{2_j}^{\{k\}}$, to be obtained by enforcing the condition

$$\tilde{f}_j\left(\boldsymbol{x}^{\{k-1\}}\right) = f_j\left(\boldsymbol{x}^{\{k-1\}}\right). \tag{8}$$

This implies that

$$c_{2_j}^{\{k\}} = \frac{2[f_j(\boldsymbol{x}^{\{k-1\}}) - f_j(\boldsymbol{x}^{\{k\}}) - \nabla^T f_j(\boldsymbol{x}^{\{k\}})(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})]}{\|\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}\|_2^2}. \tag{9}$$

To obtain strictly convex dual subproblems, we enforce $c_{2_j}^{\{k\}} = \max(\epsilon_n > 0, c_{2_j}^{\{k\}}) \; \forall \; i$ if $j = 0$, and $c_{2_j}^{\{k\}} = \max(0, c_{2_j}^{\{k\}}) \; \forall \; i$ if $j > 0$, with ϵ_n selected rather arbitrarily as 10^{-6}.

References

Alexandrov NM, Dennis JE, Lewis RM, Torczon V (1998) A trust region framework for managing the use of approximation models in optimization. Struct Optim 15:16–23

Barthelemy JFM, Haftka RT (1993) Approximation concepts for optimum structural design—a review. Struct Optim 5:129–144

Borrvall T, Petersson J (2001) Large scale topology optimization in 3D using parallel computing. Comput Methods Appl Mech Eng 190:6201–6229

Bruyneel M, Duysinx P, Fleury C (2002) A family of MMA approximations for structural optimization. Struct Multidisc Optim 24:263–276

Conn AR, Gould NIM, Toint PL (2000) Trust-region methods. MPS/SIAM Series on Optimization. SIAM, Philadelphia

Duysinx P, Bruyneel M, Fleury C (2009) Solution of large scale optimization problems with sequential convex programming. Tech. rep., LTAS—Automotive Engineering, Institute of Mechanics and Civil Engineering, University of Liège

Etman LFP, Groenwold AA, Rooda JE (2006) Sequential approximate optimization in an NLP filter framework. In: Proc. 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, Portsmouth, Virginia, paper no. AIAA-2006-7078

Etman LFP, Groenwold AA, Rooda JE (2009) On diagonal QP subproblems for sequential approximate optimization. In: Proc. eighth world congress on structural and multidisciplinary optimization, Lisboa, Portugal, paper 1065

Falk JE (1967) Lagrange multipliers and nonlinear programming. J Math Anal Appl 19:141–159

Fletcher R, Leyffer S (1998) User manual for filterSQP. Numerical Analysis Report NA\181. Department of Mathematics, University of Dundee, Dundee

Fletcher R, Leyffer S (2002) Nonlinear programming without a penalty function. Math Program 91:239–269

Fletcher R, Leyffer S, Toint PL (1998) On the global convergence of an SLP-filter algorithm. Technical Report 00/15, Department of Mathematics, University of Namur, Namur

Fletcher R, Gould NIM, Leyffer S, Toint PL, Wächter A (2002a) Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. SIAM J Optim 13:635–659

Fletcher R, Leyffer S, Toint PL (2002b) On the global convergence of a filter-SQP algorithm. SIAM J Optim 13:44–59

Fletcher R, Leyffer S, Toint P (2005) On the global convergence of a filter-SQP algorithm. Numerical analysis report NA\197, Department of Mathematics, University of Dundee, Dundee

Fleury C (1979) Structural weight optimization by dual methods of convex programming. Int J Numer Methods Eng 14:1761–1783

Fleury C (1993) Mathematical programming methods for constrained optimization: dual methods. In: Structural optimization: status and promise, AIAA, Progress in astronautics and aeronautics, chapter 7, pp 123–150

Fleury C, Braibant V (1986) Structural optimization: a new dual method using mixed variables. Int J Numer Methods Eng 23:409–428

Groenwold AA, Etman LFP (2008a) On the equivalence of optimality criterion methods and sequential approximate optimization in the classical topology layout problem. Int J Numer Methods Eng 73:297–316

Groenwold AA, Etman LFP (2008b) Sequential approximate optimization using dual subproblems based on incomplete series expansions. Struct Multidisc Optim 36:547–570

Groenwold AA, Etman LFP (2009) On the supremacy of reciprocal-like approximations in SAO—a case for quadratic approximations. In: Proc. eighth world congress on structural and multidisciplinary optimization, Lisboa, Portugal, paper 1062

Groenwold AA, Etman LFP, Snyman JA, Rooda JE (2007) Incomplete series expansion for function approximation. Struct Multidisc Optim 34:21–40

Groenwold AA, Wood DW, Etman LFP, Tosserams S (2009) Globally convergent optimization algorithm using conservative convex separable diagonal quadratic approximations. AIAA J 47:2649–2657

Groenwold AA, Etman LFP, Wood DW (2010) Approximated approximations for SAO. Struct Multidisc Optim 41:39–56

Haftka RT, Gürdal Z (1991) Elements of structural optimization, solid mechanics and its applications, vol 11, 3rd edn. Kluwer, Dordrecht

van Keulen F, Haftka RT, Kim NH (2005) Review of options for structural design sensitivity analysis. Comput Methods Appl Mech Eng 194:3213–3243

Papalambros PY, Wilde DJ (2000) Principles of optimal design: modeling and computation, 2nd edn. Cambridge University Press, New York

Pérez VM, Eldred MS, Renaud JE (2004) Solving the infeasible trust-region problem using approximations. In: Proc. 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY, paper no. AIAA 2004-4312

Ringertz UT (1988) On methods for discrete structural optimization. Eng Optim 13:47–64

Snyman JA, Hay AM (2002) The Dynamic-Q optimization method: an alternative to SQP? Comput Methods Appl 44:1589–1598

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24:359–373

Svanberg K (1995) A globally convergent version of MMA without linesearch. In: Rozvany GIN, Olhoff N (eds) Proc. First World Congress on Structural and Multidisciplinary Optimization, Goslar, Germany, pp 9–16

Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12:555–573

Svanberg K (2007) On a globally convergent version of MMA. In: Proc. seventh world congress on structural and multidisciplinary optimization, Seoul, Korea, paper no. A0052

Vanderplaats GN (2001) Numerical optimization techniques for engineering design. Vanderplaats R&D, Collorado Springs

Wood DW, Groenwold AA (2009) Non-convex dual forms based on exponential intervening variables, with application to weight minimization. Int J Numer Methods Eng 80:1544–1572

Wood DW, Groenwold AA, Etman LFP (2009) A bounded dual to circumvent the requirement of relaxation in SAO. Optim Eng (submitted) (OPTE-S-09-00053)

Zillober C (1993) A globally convergent version of the method of moving asymptotes. Struct Multidisc Optim 6:166–174

Zillober C (2001) A combined convex approximation—interior point approach for large scale nonlinear programming. Optim Eng 2:51–73

Zillober C, Schittkowski K, Moritzen K (2004) Very large scale optimization by sequential convex programming. Optim Methods Softw 19:103–120