A derivative-free trust-region augmented Lagrangian algorithm *

Charles Audet† Sébastien Le Digabel*
Mathilde Peyrega*

July 5, 2016

Abstract

We present a new derivative-free trust-region (DFTR) algorithm to solve general nonlinear constrained problems with the use of an augmented Lagrangian method. No derivatives are used, neither for the objective function nor for the constraints. An augmented Lagrangian method, known as an effective tool to solve equality and inequality constrained optimization problems with derivatives, is exploited to minimize the subproblems, composed of quadratic models that approximate the original objective function and constraints, within a trust region. The trust region ratio which leads the classical update rules for the trust region radius is defined by comparing the true decrease of the augmented Lagrangian merit function with the expected decrease. This mechanism allows to reuse the basic unconstrained DFTR update rules with minor modifications. Computational experiments on a set of analytical problems suggest that our approach outperforms HOPSPACK and is competitive with COBYLA. Using an augmented Lagrangian, and more generally a merit function, to design the DFTR update rules with constraints is shown to be an efficient technique.

*This work was supported in part by NSERC grant RGPIN-2015-05311.

†GERAD and Département de mathématiques et génie industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec, Canada H3C 3A7. www.gerad.ca/Charles.Audet, www.gerad.ca/Sebastien.Le.Digabel, mathilde.peyrega@polymtl.ca.

Keywords: Derivative-free optimization; Trust-region algorithms; Equality and inequality constraints; Augmented Lagrangian.

1 Introduction

We consider the general optimization problem

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{subject to} & h(x) = 0 \\ & g(x) \leq 0 \\ & l \leq x \leq u \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a single-valued function, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ correspond to equality and inequality constraints, l, $u \in (\mathbb{R} \cup \{\pm\infty\})^n$ are bounds and n, m, $p \in \mathbb{N}$ are the dimension of the problem, the number of equalities and inequalities, respectively. We are interested in a class of derivative-free optimization (DFO) problems in which the functions f, h and g are twice continuously differentiable, but their derivatives are unavailable. Such a situation may occur, for example, when a simulation is involved, for which it is difficult or costly to estimate derivatives.

Algorithms for problems of the form (1) exist. Powell [24] proposes a derivative-free trust-region (DFTR) algorithm named COBYLA in which linear models are built by interpolating on non-degenerate simplices at each iteration. A merit function based on the infinity norm of the constraints allows to compute the trust region ratio. Originally proposed for inequality constraints, the software package NLopt [16] re-implements COBYLA and replaces each equality constraint by a pair of inequalities. Recently, Sampaio and Toint [26, 27] adapt the trust-funnel method in a DFTR algorithm to solve Problem (1). The trust-funnel method treats the equality constraint with no need of a filter, a barrier or a penalty [13].

Other algorithms provide treatment for problems close to (1) in a DFO context. A SQP derivative-free trust-region algorithm is proposed by Tröltzsch [28] for equality constrained problems. The NOWPAC algorithm [5] handles nonlinear inequality constraints in a DFTR algorithm, where strict feasibility is guaranteed at each iteration. Two recent papers deploy inexact restoration schemes. In [2] a violation measure is minimized in the restoration phase with a derivative-free algorithm, and a penalty-like merit function is exploited in the optimization phase. In [11], a filter is used in the optimization phase. Both papers uses models of the function in their optimization phase. The PBTR algorithm proposed in [3] considers problems with inequalities treated with the progressive barrier of [4] and uses IPOPT [29] to solve the underlying constrained subproblems. In [7] a DFTR algorithm treats Problem (1) using the gradients of the nonlinear constraints.

Many DFO algorithms use the augmented Lagrangian to handle constraints. The authors of [19] highlight a DFO issue: *"as this method requires both the objective function and constraints evaluations, it can be costly when the constraints can be easily evaluated without evaluating the objective function".* This is not the case here, but it could be considered in future work. Torczon and Lewis [17] adapt the augmented Lagrangian algorithm [8] to a direct-search algorithm, without the use of derivatives. This adapted augmented Lagrangian is implemented in the direct-search HOPSPACK [18] method. The Algencan augmented Lagrangian method [6] is used in [10] to treat the difficult constraints whereas the easiest constraints are directly integrated into a subproblem solved by a DFO algorithm. An augmented Lagrangian method is also used in [30] where a DFTR algorithm is associated to a filter to solve problems with separable structure. Blackbox Optimization algorithms with surrogate models and augmented Lagrangian for inequalities and equalities are proposed in [14] and [22]. Finally in [1], augmented Lagrangian methods improve the solution of quadratic subproblems arising in the MADS direct-search algorithm.

Solving subproblems and trust region radius update are two important elements in the design of a DFTR algorithm for constrained problems. This paper proposes a new derivative-free trust-region algorithm called DFTR^{L} treating general constrained problems by using an augmented Lagrangian. Section 2 presents a short review of the DFTR framework. Section 3 presents our new DFTR augmented Lagrangian algorithm for constraints, named DFTR^{L}. Implementation details and computational results are exposed in Section 4. The proposed algorithm performs similarly to COBYLA on a set of analytical problems but outperforms COBYLA on problems with equalities and HOPSPACK on the entire chosen set of analytical problems. We conclude and evoke future work in Section 5.

2 A brief review of the DFTR framework

DFTR algorithms are inspired by the classical trust-region framework. Their originalities are the methods used to build the models and the subjacent theory guarantying similar convergence properties. As a trust-region algorithm [21, chap. 4], a DFTR algorithm solves a subproblem on a region where the original functions are replaced by models. These algorithms are efficient when the models of the original functions are good approximations within a trust region, a ball centered on the current iterate x^k, of radius Δ^k, the trust region radius at iteration k. Instead of solving the original problem, subproblems are iteratively optimized.

In classical trust-region methods with derivatives, the models are constructed using first or second order Taylor polynomials of the functions. In DFTR algorithms, the models of the functions cannot be built with the derivatives, since they are unavailable. Frequently used techniques to build models include interpolation or regression from a sample set of points around the current iterate. This sample set at iteration k is denoted by $\mathcal{Y}^k(x^k) \subset \mathbb{R}^n$.

Some properties are defined to characterized models offering similar properties than the first or second order truncated Taylor models based on derivatives. It is the case of the fully-linear models and the fully-quadratic models (see [9, chap. 6] for the formal definitions). Fully-linear models or fully-quadratic models can be guaranteed by some properties of the sample set. The well-poisedness is a geometric property characterizing a set of sample points. The theory is presented in [9, chap. 3]. If the sample set well-poisedness is satisfying, then an interpolated or regressed model computed from this sample set can be certifiably fully-linear or certifiably fully-quadratic. Some algorithms detailed in [9, chap. 6] explain how to construct such sample sets and models. From a given sample set it is also possible to improve the well-poisedness by replacing some points.

The stopping criteria is typically based on the radius Δ^k. Under certain assumptions the convergence analysis shows that the sequence of the trust region radii converges to zero, whereas in most trust-region algorithm the trust region radii diverge.

To summarize, DFTR is a trust-region algorithm with different mechanisms to build the models. Thanks to new theories characterizing the sample set, we can certify to have Taylor-like models. Updates rules for the trust-region are simply adapted from the classical trust-region methods with derivatives. For more details, a basic unconstrained DFTR algorithm is presented in [9, chap. 10].

3 A DFTR algorithm using an augmented Lagrangian method

Augmented Lagrangian methods are a class of algorithms solving constrained nonlinear problems with derivatives. They belong to the class of penalty methods and use iteratively reformulated unconstrained problems thanks to an augmented Lagrangian function, which is the Lagrangian with an additional penalty term. Different augmented Lagrangian functions and algorithms exist. We use the augmented Lagrangian function defined by Powell, Hestenes and Rockafellar [15, 23, 25], called the PHR augmented Lagrangian. It is the one used in the Algencan algorithm detailed in [6]:

$$\mathcal{L}_\rho(x;\lambda,\mu) = f(x) + \frac{\rho}{2}\left(\sum_{i=1}^{m}\left[h_i(x) + \frac{\lambda_i}{\rho}\right]^2 + \sum_{i=1}^{p}\left[\max\left(0, g_i(x) + \frac{\mu_i}{\rho}\right)\right]^2\right),$$

where $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p_+$ and $\rho \geq 0$ are penalty coefficients. The coefficient λ and μ are approximations of the Lagrange multipliers.

3.1 Solving the subproblems with an augmented Lagrangian method

As in the DFTR algorithm, our algorithm proposed in Section 3.2 solves a subproblem at iteration k within a trust region:

$$\begin{aligned} \min_{x\in B(x^k;\Delta^k)} \quad & \tilde{f}^k(x) \\ \text{subject to} \quad & \tilde{h}^k(x) = 0 \\ & \tilde{g}^k(x) \leq 0 \\ & l \leq x \leq u, \end{aligned} \tag{2}$$

where the functions $\tilde{f}^k$, $\tilde{h}^k$, and $\tilde{g}^k$ are quadratic models of f, h and g. The subproblems can be nonconvex with indefinite quadratic constraints, and are solved with an augmented Lagrangian algorithm. The augmented Lagrangian of Problem (2) is:

$$\tilde{\mathcal{L}}^k_\rho(x;\lambda,\mu) = \tilde{f}^k(x) + \frac{\rho}{2}\left(\sum_{i=1}^{m}\left[\tilde{h}^k_i(x) + \frac{\lambda_i}{\rho}\right]^2 + \sum_{i=1}^{p}\left[\max\left(0, \tilde{g}^k_i(x) + \frac{\mu_i}{\rho}\right)\right]^2\right).$$

One can observe that the augmented Lagrangian of the subproblem, $\tilde{\mathcal{L}}^k_\rho$, is also a model of the augmented Lagrangian of the Problem (1), $\mathcal{L}_\rho$.

Birgin and Martinez [6] list advantages of using an augmented Lagrangian, and propose the Algencan algorithm, from which we borrowed the augmented Lagrangian to solve our subproblem. The principles of Algencan is to minimize at each iteration the unconstrained problem obtained with the augmented Lagrangian function with a precision ε^k satisfying $\varepsilon^k \to 0$. The three penalty coefficients ρ, λ and μ are updated at the end of each iteration. For example ρ is increased when the improvement is not sufficient regarding the feasibility of the new current point. This augmented Lagrangian algorithm always manages a current point x^k satisfying the bounds constraints. In the following we denote by $\tilde{\lambda}^k$, $\tilde{\mu}^k$, and $\tilde{\rho}^k$ the values of these coefficients at the end of the subproblem solution at iteration k of our DFTR algorithm. Then the current augmented Lagrangian function at iteration k after solving the subproblem is denoted by $\mathcal{L}_{\tilde{\rho}^k}(x;\tilde{\lambda}^k,\tilde{\mu}^k)$, whereas the current augmented Lagrangian model function is denoted by $\tilde{\mathcal{L}}_{\tilde{\rho}^k}(x;\tilde{\lambda}^k,\tilde{\mu}^k)$.

3.2 A DFTR algorithm based on the augmented Lagrangian

The current augmented Lagrangian function and the current augmented Lagrangian model function are used to compute the trust region ratio r^k, measuring the quality of the minimization of the original problem in comparison with the expected minimization obtained with the subproblem. We denote by $\tilde{x}^k$ the solution of the model subproblem:

$$r^k = \frac{\mathcal{L}_{\tilde{\rho}^k}(x^k;\tilde{\lambda}^k,\tilde{\mu}^k) - \mathcal{L}_{\tilde{\rho}^k}(\tilde{x}^k;\tilde{\lambda}^k,\tilde{\mu}^k)}{\tilde{\mathcal{L}}_{\tilde{\rho}^k}(x;\tilde{\lambda}^k,\tilde{\mu}^k) - \tilde{\mathcal{L}}_{\tilde{\rho}^k}(\tilde{x}^k;\tilde{\lambda}^k,\tilde{\mu}^k)}.$$

The new algorithm named DFTR^{L} is outlined in Figure 4. The algorithm parameters η_0, η_1, γ_{inc}, and γ_{dec} must respect the following conditions: $0 \leq \eta_0 < \eta_1 < 1$, $0 < \gamma_{inc} < 1 < \gamma_{dec}$. The parameters η_0 and η_1 are thresholds to quantify the quality of the ratio r^k. The parameters γ_{inc} and γ_{dec} are coefficients to increase or decrease the trust region radius Δ^k based on the quality of the ratio r^k.

Algorithm 1: Algorithm DFTR^{L}: iteration k.

1 **Model construction**
Construct the set of sample points $\mathcal{Y}^k$ around x^k and build the models $\tilde{f}^k$, $\tilde{h}^k$, $\tilde{g}^k$.

2 **Subproblem solution**
Solve Subproblem (2) within the trust region with the augmented Lagrangian algorithm. The algorithm returns $\tilde{x}^k$, $\tilde{\lambda}^k$, $\tilde{\mu}^k$ and $\tilde{\rho}^k$.

3 **Step calculation**
Evaluate f, h, and g at $\tilde{x}^k$ and compute the ratio r^k at $\tilde{x}^k$.

4 **Trust region radius update**
If $r^k \geq \eta_1$, then set $x^{k+1} = \tilde{x}^k$ and $\Delta^{k+1} = \min\left(\gamma_{inc}\Delta^k, \Delta_{max}\right)$.
If $\eta_0 \leq r^k < \eta_1$, then set $x^{k+1} = \tilde{x}^k$ and $\Delta^{k+1} = \Delta^k$.
If $r^k < \eta_0$, then set $x^{k+1} = x^k$ and $\Delta^{k+1} = \gamma_{dec}\Delta^k$.

4 Implementation details and computational results

Our algorithm DFTR^{L} is implemented in Python and this section compares it with two state-of-the-art software packages. We first describe our set of analytical problems and then the tools to analyse the results.

4.1 Computational testbed

We used a set of 83 small-scale analytical problems from the CUTEst collection [12]. This set includes the test problems used in [3]. Among them, 40 contain only inequality constraints. Their characteristics are presented in appendix. The initial point proposed in the CUTEst collection satisfies the bound constraints. A budget of $100(n+1)$ blackbox evaluations is chosen.

COBYLA is a DFTR algorithm using a l_∞ merit function and linear models, and HOPSPACK is a direct-search based method using an augmented Lagrangian to treat general constraints. We use the NLopt version of COBYLA with default settings. HOPSPACK is used with default parameters and a tolerance of 10^{-7} for each constraint. Note that HOPSPACK allows an explicit treatment of linear constraints, and as neither COBYLA nor our implementation contains this feature, it has been disabled in order to allow a fair comparison.

Data profiles and performance profiles from [20] are used to analyze performance. These graphs compare different algorithms on a given set of problems. For a tolerance parameter $\tau \in [0;1]$, fixed to 10^{-3} in this paper, data profiles present, for a particular budget of evaluations, the percentage of problems providing a solution within τ to a reference equal to the best solution found by all the algorithms. When no feasible solution has been found, no algorithm is considered to have solved this problem. A point is considered feasible when every constraint is satisfied within a tolerance of 10^{-7}.

Performance profiles from [20] are also used. A performance ratio $r_{p,s}$ is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$

for Algorithm s on Problem p where S is the set of algorithms tested. If for example $r_{p,s} = 2$, Algorithm s needs twice the number of evaluations of the best algorithm to solve Problem p, within a tolerance τ. The performance profiles show for $\alpha \geq 1$ the fraction of problems solved by Algorithm s with a ratio $r \leq \alpha$. The value of a performance profile for $\alpha = 1$ indicates the proportion of problems a given algorithm solves the best (two algorithms can equally solve one problem), and a performance profile when $\alpha \to \infty$ indicates the proportion of problems efficiently solved by the algorithm.

The sample set used to build the quadratic interpolation models requires at each iteration $(n+1)(n+2)/2$ points. These points correspond to the most recent points in a ball of radius $2\Delta^k$ around the current iterate x^k. If there are not enough points, then the geometry improvement algorithm is called to select new points by keeping a well-poised geometry of the sample set.

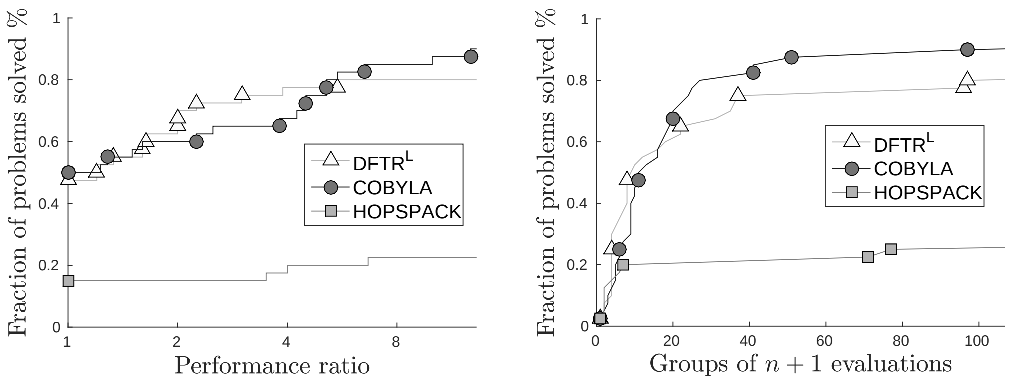
The subproblems are optimized with the Algencan algorithm implemented in the NLopt package. A limit of 5000 iterations is imposed, and the subproblem tolerance for each constraint is 10^{-8}. The original problem tolerance for each constraint is 10^{-7}.

4.2 Comparison with COBYLA and HOPSPACK

Our algorithm is compared to the two state-of-the art software packages COBYLA and HOPSPACK. The results are presented separately for constrained problems without equalities and those with at least one equality.

Inequality constrained problems. The performance profiles in Figure 1(a) show that our algorithm is competitive with COBYLA on the benchmark set of 40 inequality constrained CUTEst problems. Both $\mathrm{DFTR}^{\mathrm{L}}$ and COBYLA perform better than the direct-search HOPSPACK algorithm using augmented Lagrangian method. The performance of $\mathrm{DFTR}^{\mathrm{L}}$ is comparable to that of COBYLA even if $\mathrm{DFTR}^{\mathrm{L}}$ is slightly below. The performance profiles show that $\mathrm{DFTR}^{\mathrm{L}}$ solves 10% less of inequality constrained problems than COBYLA. The data profiles in Figure 1(b) confirm these observations: Even if $\mathrm{DFTR}^{\mathrm{L}}$ seems a bit faster when the number of function evaluations is above $20(n+1)$, COBYLA outperforms $\mathrm{DFTR}^{\mathrm{L}}$ on 10% of the tested problems with a larger number of evaluations. These results show that $\mathrm{DFTR}^{\mathrm{L}}$ is competitive with COBYLA but slightly less efficient.

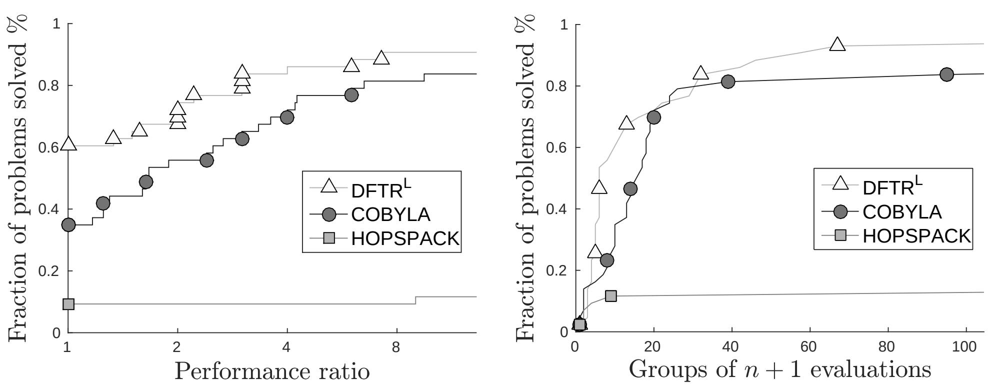
General constrained problems with at least one equality. Computational results with 43 problems containing at least one equality show that $\mathrm{DFTR}^{\mathrm{L}}$ globally outperforms COBYLA on problems with at least one equality. The performance profiles in Figure 2(a) show that our algorithm solves more than 20% of the problems faster than COBYLA, and is able to asymptotically solves almost 10% more. Both $\mathrm{DFTR}^{\mathrm{L}}$ and COBYLA dominate the direct-search HOPSPACK algorithm. The data profiles in Figure 2(b) confirm



(a) Perf. profiles at $100(n+1)$ evaluations. (b) Data profiles with $\tau = 10^{-3}$.

Figure 1: Comparison of DFTR^{L} with COBYLA and HOPSPACK on analytical CUTEst problems with only inequalities.

these observations. The performance of DFTR^{L} is comparable to that of COBYLA, and DFTR^{L} outperforms COBYLA slightly.



(a) Perf. profiles at $100(n+1)$ evaluations. (b) Data profiles with $\tau = 10^{-3}$.

Figure 2: Comparison of DFTR^{L} with COBYLA and HOPSPACK on analytical CUTEst problems with at least one equality.

5 Discussion

This work proposes a derivative-free trust-region algorithm to treat general nonlinear constraints for problems without the use of their derivatives. The augmented Lagrangian method and function are used to both solve the trust-region subproblem and simply design the update rules of the derivative-free trust-region algorithm, with few modifications to the unconstrained DFTR framework.

Computational experiments are conducted on a collection of 80 problems from the CUTEst collection with two state-of-the art algorithms: HOPSPACK, a direct-search algorithm using an augmented Lagrangian method for the constraints, and COBYLA, a DFTR algorithm. Our new algorithm, $\mathrm{DFTR}^{\mathrm{L}}$, outperforms HOPSPACK and is competitive with COBYLA on analytical problems. It is worth noting that $\mathrm{DFTR}^{\mathrm{L}}$ performs better on problems with equalities.

Future work may consider other penalty functions to solve the subproblem and design the trust region ratio. Other sample set managements could be tested to improve the performance. Finally, the progressive barrier [4] could be adapted to this new algorithm to improve the treatment of inequalities.

6 Appendix

Name	n	p	lower bounds	upper bounds	initial point
avgasb	8	10	8	8	Feasible
b2	3	3	0	0	Infeasible
chaconn1	3	3	0	0	Infeasible
himmelp5	2	3	2	2	Infeasible
hs10	2	1	0	0	Infeasible
hs11	2	1	0	0	Infeasible
hs12	2	1	0	0	Feasible
hs15	2	2	0	1	Infeasible
hs18	2	2	2	2	Infeasible
hs19	2	2	2	2	Infeasible
hs22	2	2	0	0	Infeasible
hs23	2	5	2	2	Infeasible
hs24	2	3	2	0	Feasible
hs29	3	1	0	0	Feasible
hs30	3	1	3	3	Feasible
hs31	3	1	3	3	Feasible
hs33	3	2	3	1	Feasible
hs34	3	2	3	3	Feasible
hs35	3	1	3	0	Feasible
hs36	3	1	3	3	Feasible
hs43	4	3	0	0	Feasible
hs57	2	1	2	0	Feasible
hs64	3	1	3	0	Infeasible
hs72	4	2	4	4	Infeasible
hs76	4	3	4	0	Feasible
hs84	5	6	5	5	Feasible
hs86	5	10	5	0	Feasible
hs95	6	4	6	6	Infeasible
hs96	6	4	6	6	Infeasible
hs97	6	4	6	6	Infeasible
hs98	6	4	6	6	Infeasible
hs100	7	4	0	0	Feasible
hs101	7	6	7	7	Infeasible
hs108	9	13	1	0	Infeasible
kiwcresc	3	2	0	0	Infeasible
lootsma	3	2	0	1	Feasible
polak6	5	4	0	0	Infeasible
simpllpb	2	3	0	0	Infeasible
snake	2	2	0	0	Infeasible
spiral	3	2	0	0	Feasible

Table 1: Description of the 40 analytical problems with only inequalities ($m = 0$)

References

[1] N. Amaioua, C. Audet, A.R. Conn, and S. Le Digabel. Efficient solution of quadratically constrained quadratic subproblems within a direct-search algorithm. Technical Report G-2016-45, Les cahiers du GERAD, 2016.

[2] M.B. Arouxét, N.E. Echebest, and E.A. Pilotta. Inexact Restoration method for nonlinear optimization without derivatives. *Journal of Computational and Applied Mathematics*, 290:26–43, 2015.

[3] C. Audet, A.R. Conn, S. Le Digabel, and M. Peyrega. A progressive barrier derivative-free trust-region algorithm for constrained optimization. Technical Report G-2016-49, Les cahiers du GERAD, 2016.

[4] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, 20(1):445–472, 2009.

[5] F. Augustin and Y.M. Marzouk. NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints. Technical report, Massachusetts Institute of Technology, 2014.

[6] E. G. Birgin and J. M. Martínez. Improving Ultimate Convergence of an Augmented Lagrangian Method. *Optimization Methods Software*, 23(2):177–195, 2008.

[7] P.D. Conejo, E.W. Karas, and L.G. Pedroso. A trust-region derivative-free algorithm for constrained optimization. *Optimization Methods and Software*, 30(6):1126–1145, 2015.

[8] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[9] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[10] M.A. Diniz-Ehrhardt, J.M. Martinez, and L.G. Pedroso. Derivative-free methods for nonlinear programming with general lower-level constraints. *To appear in* Journal of Computational and Applied Mathematics, 2011.

[11] N. Echebest, M.L. Schuverdt, and R.P. Vignau. An inexact restoration derivative-free filter method for nonlinear programming, 2015. To appear in *Computational and Applied Mathematics*.

[12] N.I.M. Gould, D. Orban, and Ph.L. Toint. CUTEst: a Constrained and

Name	n	m	p	lower bounds	upper bounds	initial point
booth	2	2	0	0	0	Infeasible
bt4	3	2	0	0	0	Infeasible
bt5	3	2	0	0	0	Infeasible
bt8	5	2	0	0	0	Infeasible
bt13	5	1	0	0	1	Infeasible
byrdsphr	3	2	0	0	0	Infeasible
cluster	2	2	0	0	0	Infeasible
dixchlng	10	5	0	0	0	Infeasible
extrasim	2	1	0	0	2	Infeasible
gottfr	2	2	0	0	0	Infeasible
hs006	2	1	0	0	0	Infeasible
hs007	2	1	0	0	0	Infeasible
hs008	2	2	0	0	0	infeasible
hs014	2	1	1	0	0	Infeasible
hs027	3	1	0	0	0	Infeasible
hs028	3	1	0	0	0	Feasible
hs032	3	1	1	3	0	Feasible
hs039	4	2	0	0	0	Infeasible
hs040	4	3	0	0	0	Infeasible
hs042	4	0	0	0	0	Infeasible
hs048	5	2	0	0	0	Feasible
hs052	5	3	0	0	0	Infeasible
hs053	5	3	5	5	5	Infeasible
hs054	6	1	0	6	6	Infeasible
hs055	6	6	0	6	2	Infeasible
hs060	3	1	0	3	3	Infeasible
hs061	3	2	0	0	0	Infeasible
hs062	3	1	0	3	3	Feasible
hs063	3	2	0	3	0	Infeasible
hs071	4	1	1	4	40	Feasible
hs073	4	1	2	4	0	Infeasible
hs078	5	3	0	0	0	Infeasible
hs080	5	3	0	5	5	Infeasible
hs111	10	3	0	10	10	Infeasible
hs112	10	3	0	10	0	Infeasible
hs114	10	3	8	0	0	Infeasible
hypcir	2	2	0	0	0	easible
maratos	2	1	0	0	0	easible
odfits	10	6	0	10	0	easible
portfl1	12	1	0	12	12	easible
supersim	2	2	0	2	0	easible
tame	2	1	0	2	0	easible
zangwil3	3	3	0	0	0	easible

Table 2: Description of the 40 analytical problems with at least one equality constraint

Unconstrained Testing Environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015. Code available at `https://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki`.

[13] N.I.M. Gould and Ph.L. Toint. Nonlinear programming without a penalty function or a filter. *Mathematical Programming*, 122(1):155–196, 2010.

[14] R.B. Gramacy, G.A. Gray, S. Le Digabel, H.K.H. Lee, P. Ranjan, G. Wells, and S.M. Wild. Modeling an Augmented Lagrangian for Blackbox Constrained Optimization. *Technometrics*, 58(1):1–11, 2016.

[15] M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[16] S.G. Johnson. The NLopt nonlinear-optimization package. `http://ab-initio.mit.edu/wiki/index.php/NLopt`.

[17] R.M. Lewis and V. Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002.

[18] R.M. Lewis and V. Torczon. A direct search approach to nonlinear programming problems using an augmented lagrangian method with explicit treatment of linear constraints. Technical report, College of William & Mary, 2010.

[19] J.M. Martínez and F.N.C Sobral. Constrained derivative-free optimization on thin domains. *Journal of Global Optimization*, 56(3):1217–1232, 2013.

[20] J.J. Moré and S.M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[21] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, 1999.

[22] V. Picheny, R.B. Gramacy, S.M. Wild, and S. Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. Technical Report G-2016-43, Les cahiers du GERAD, 2016.

[23] M.J.D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1969.

[24] M.J.D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In S. Gomez and J.-P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, volume

275 of *Mathematics and Its Applications*, pages 51–67. Springer Netherlands, 1994.

[25] R.T. Rockafellar. Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming. *SIAM Journal on Control*, 12(2):268–285, 1974.

[26] Ph.R. Sampaio and Ph.L. Toint. A derivative-free trust-funnel method for equality-constrained nonlinear optimization. *Computational Optimization and Applications*, 61(1):25–49, 2015.

[27] Ph.R. Sampaio and Ph.L. Toint. Numerical experience with a derivative-free trust-funnel method for nonlinear optimization problems with general nonlinear constraints. *Optimization Methods and Software*, 31(3):511–534, 2016.

[28] A. Tröltzsch. A sequential quadratic programming algorithm for equality-constrained optimization without derivatives. *Optimization Letters*, 10(2):383–399, 2016.

[29] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[30] D. Xue and W. Sun. On convergence analysis of a derivative-free trust region algorithm for constrained optimization with separable structure. *Science China Mathematics*, 57(6):1287–1302, 2014.