Constrained bundle methods with inexact minimization applied to the energy regulation provision problem [1]

Tomasz T. Gorecki * **Colin N. Jones** *

* *Automatic control Laboratory, École polytechnique fédérale de Lausanne, Switzerland*

Abstract: We consider a class of large scale robust optimization problems. While the robust optimization literature often relies on structural assumptions to reformulate the problem in a tractable form using duality, this method is not always applicable and can result in problems which are very large. We propose an alternative way of solving such problems by applying a constrained bundle method. The originality of the method lies in the fact that the minimization steps in the bundle method are solved approximately using the alternating direction method of multipliers. Numerical results from a power grid regulation problem are presented and support the relevance of the approach.

Keywords: Bundle methods, robust optimization, frequency regulation, smart grid, semi-infinite optimization

1. INTRODUCTION

We consider an optimization problem of the form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & l(x,z) \leq 0 \end{array} \tag{1}$$

where the decision variable is $x \in \mathbb{R}^{n_x}$ and $z \in \mathbb{R}^{n_z}$ represents the uncertainty entering the problem. Optimization problems with uncertainty are very common in operations research, management science and optimal control applications. A popular modeling paradigms to treat such problems is robust optimization (Ben-Tal, A. et al., 2009), where the uncertain parameters are assumed to lie in a set $\mathcal{Z}$ and the decision should satisfy constraint for any value of the uncertain parameter so that:

$$l(x,z) \leq 0, \quad \forall z \in \mathcal{Z} \tag{2}$$

Finding a x satisfying (2) is in general a challenging task, even when l is convex in x for all z. One way to proceed is to find a reformulation that circumvents the universal quantifier and has a tractable form. Most tractable reformulations of robust constraints rely on a convex reformulation (Ben-Tal et al., 2014). Supposing $\mathcal{Z}$ takes the form:

$$\mathcal{Z} = \{z_0 + A\zeta \,|\, \zeta \in Z \subset \mathbb{R}^m\} \tag{3}$$

where z_0 is the nominal value of the uncertainty. Under the assumption that l is concave in z for all x and additional technical assumptions, the main result of Ben-Tal et al. (2014) reads:

Theorem 1. x satisfies (2) if and only if x, v satisfying:

$$z_0^\top v + \delta^*(A^\top v|Z) - g_*(x,v) \leq 0 \tag{4}$$

where $\delta^*(.|Z)$ is the so-called support function of Z and g_* the partial concave conjugate of g defined as:

$$l_*(x,v) = \inf_{y \in dom(g)} v^T y - g(x,y)$$

In particular, when l is linear in x and z respectively and Z is defined as the intersection of finitely many convex cones, then Equation (4) is convex in x, v and can be solved efficiently.

Despite this, it may be necessary for large systems of robust inequalities to add a very large number of auxiliary variables in order to render the problem tractable. Moreover, this approach is only applicable under structural assumption on l, in particular concavity in z for all x. Considering these limitations, other methods have been proposed to solve such problems. Mutapcic and Boyd (2009) propose a "cutting-set" method where the problem is solved in an alternating fashion: the decision variable x is computed with a finite subset of the constraint; based on that decision, a pessimizing realization of the uncertainty z is computed by maximizing the constraint function on the uncertainty set. That pessimizer is then added to the surrogate uncertainty set and the procedure is repeated until convergence. It has been reported in Bertsimas et al. (2015) that the cutting set method can outperform reformulations in some cases.

The cutting set method is very related to cutting planes method and their modern versions called bundle methods (Bonnans et al., 2006). Though bundle methods were originally designed for nonsmooth unconstrained convex optimization, variants tackle constrained problems. In this work, a method is proposed to solve large scale robust optimization problems. It combines ideas from the cutting set method of Mutapcic and Boyd (2009) and of the constrained bundle method of Sagastizábal and Solodov (2005). Instead of assuming exact solutions to the min-

[1] This work has received support from the Swiss National Science Foundation under the GEMS project (grant number 200021 137985) and the European Research Council under the European Unions Seventh Framework Programme (FP/2007-2013)/ ERC Grant Agreement n. 307608 (BuildNet).

imization subproblems within the bundle method iterations, we propose to use an approximate solution to the minimization step and in particular to use the alternating direction method of multipliers (ADMM) to perform this step efficiently. We will see that obtaining low accuracy solutions to the minimization quickly allows to solve bigger problems faster.

The method is demonstrated on "'flexibility commitment problems" arising in power systems applications. Increasing attention in the literature has been devoted to how to provide balancing services, also called ancillary services (AS) to the power grid with load-side resources. Our previous work (Gorecki et al., 2015b) formalizes the computation of resource flexibility in a theoretically sound framework, relying on robust optimization. Other works (Zhang et al., 2014) also rely on robust or stochastic optimization to solve similar problems. These methods ultimately require to solve large robust optimization problems. Due to the multi-stage nature of the problems, the number of decision variables tends to grow very quickly, and even in the cases where tractable reformulations are available, the size of the problem can become prohibitive.

We show how the method proposed allows to tackle instances that could not be solved using convex reformulations. It also provides slightly suboptimal solutions on smaller instance faster and thanks to the monotonic feasibility property of the bundle method, usually provide feasible solutions.

2. THE BUNDLE METHODOLOGY

We consider a convex minimization problem of the form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \end{array} \tag{5}$$

where f, g are convex proper functions defined on $\mathbb{R}^n$. It is assumed that f is a known 'simple' convex function while g may be nonsmooth. It is assumed that an oracle is available for g giving for a query point x the value function $g(x)$ and a vector $s \in \partial g(x)$, so that:

$$\forall y \in \mathbb{R}^n,\ g(y) \geq g(x) + s^\top (y - x) \tag{6}$$

The spirit of the method proposed in Sagastizábal and Solodov (2005) is to apply a standard unconstrained bundle method on the improvement function at the current center y, defined as:

$$h_y(x) := \max(f(x) - f(y), g(x)) \tag{7}$$

relying on the fact that a point x is a minimizer of (5) if and only if it minimizes h_x on $\mathbb{R}^n$ (Theorem 2.1 in Sagastizábal and Solodov (2005))

At each iteration k, the algorithm queries a point x_k and maintains a bundle of all the information collected so far:

$$\mathcal{B}_k = \bigcup_{i<k} \{(x_i, f_i = f(x_i), g_i = g(x_i), s_i \in \partial g(x_i))\} \tag{8}$$

and the information for the current center $(y_k, f(y_k))$

This bundle of information is used to define a cutting plane model of the improvement function:

$$H_k(x) = \max(f(x) - f(y_k), G_k(x)) \tag{9}$$

where

$$G_k(x) = \max_{i<k}(g_i + s_i^\top (x - x_i)) \tag{10}$$

where each hyperplane $g_i + s_i^\top (. - x_i)$ is a lower bound on the constraint function g and consequently $G_k \leq g \quad \forall k$. Given a proximal parameter c_k, the next iterate x_k is generated by solving a proximal minimization problem at the current center:

$$\text{minimize}_x \quad \psi_k(x) := H_k(x) + \frac{1}{2c_k}\|x - y_k\|^2 \tag{11}$$

Due to strict convexity, this problem admits a unique minimizer x_k. The oracle is called at x_k to recover the values g_k and s_k. A sufficient decrease condition is used to determine if the center is updated. Defining the expected improvement

$$\delta_k := \psi_k(y_k) - \psi_k(x_k) = g^+(y_k) - H_k(x_k) - \frac{1}{2c_k}\|x_k - y_k\|^2 \tag{12}$$

where $g^+(.) = \max(g(.), 0)$. To determine if the center is updated, we check if the actual improvement $h_{y_k}(y_k) - h_{y_k}(x_k)$ achieves at least a fixed fraction of the expected improvement, *i.e.* :

$$h_{y_k}(y_k) - h_{y_k}(x_k) \geq m\delta_k \tag{13}$$

where m is a tuning parameter. This translates to the following two inequalities:

$$\begin{aligned} f(x_k) - f(y_k) &\leq g^+(y_k) - m\delta_k \\ g(x_k) &\leq g^+(y_k) - m\delta_k \end{aligned} \tag{14}$$

We can see in the first equation that if the current center is infeasible, we can sacrifice optimality in order to improve the feasibility. On the other hand, we do not allow to sacrifice feasibility in order to improve the cost. A direct consequence of that is the monotonic improvement of the infeasibility of the centers. Furthermore, once a feasible center has been identified, the center remains feasible. If the test (14) is passed, we update the center so that $y_{k+1} = x_k$ (serious step), otherwise we set $y_{k+1} = y_k$ (null step).

Let us define $\hat{h}_k = \frac{1}{c_k}(y_k - x_k)$, the so-called aggregate subgradient and $\epsilon_k = g^+(y_k) - H_k(x_k) - c_k\|\hat{h}_k\|^2$ so that $\delta_k = \epsilon_k + \frac{1}{2c_k}\|\hat{h}_k\|^2$. It holds that

$$\hat{h}_k \in \partial_{\epsilon_k} h_k(y_k) \tag{15}$$

meaning that $h_k(.) \geq h_k(y_k) + \hat{h}_k^\top(. - y_k) - \epsilon_k$.
Termination of the algorithm is decided using the termination criterion $\delta_k \leq tol_\delta$. Indeed, it implies that both ϵ_k and $\hat{h}_k$ are small and yields the approximate optimality condition (Sagastizábal and Solodov, 2005):

$$\begin{gathered} \forall M > 0,\ \forall x \ \text{ s.t. } \|y_k - x\| \leq M, \\ h_k(x) \geq h_k(y_k) - \epsilon_k - M\|\hat{h}_k\|^2 \end{gathered} \tag{16}$$

Typically, a split termination criterion is used:

$$\epsilon_k \leq tol_\epsilon \text{ and } \|\hat{h}_k\|^2 \leq tol_g \tag{17}$$

When the center is updated, the improvement function becomes $h_{k+1} = h_{y_{k+1}}$. Since the cost at the center is allowed to grow to improve feasibility, it might be that $h_{k+1} \leq h_k$ with strict inequality for some points. Therefore the lower bounding model H_k on h_k will not in general be a lower bound on h_{k+1}, hence the need to change H_{k+1} accordingly. In our case, G_k is still a lower bound on h_{k+1} and we can simply take:

$$H_{k+1}(x) = \max(G_{k+1}(x), f(x) - f(y_{k+1})) \tag{18}$$

3. ALGORITHM DESCRIPTION

We are faced with the robust optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & l(x,z) \leq 0 \; \forall z \in \mathcal{Z} \end{array} \tag{19}$$

where $z \in \mathbb{R}^{n_z}$ is the uncertain parameter assumed to lie in a known set of dimension n_z and l is convex in x for all $z \in \mathcal{Z}$. Problem (19) can be put in the form of Problem (5) by taking:

$$g(x) = \max_{z \in \mathcal{Z}} l(x,z) \tag{20}$$

g is also convex but usually nonsmooth and expensive to evaluate. Therefore, this problem fits the assumptions of the bundle methodology.

3.1 Algorithm structure

We state here the main steps of the algorithm:

(1) **Initialization:** Define a starting point x_0. Call the oracle at x_0 to collect cutting plane $\mathcal{B}_0 \leftarrow (g_0, s_0)$ to form the first model $H_0(x) = \max(f(x) - f(x_0), G_0(x))$ with $G_0 = g_0 + s_0^\top(x - x_0)$. Define the first center $y_0 = x_0$
(2) **Minimization Step at iteration** k**:** Solve Problem (11) and yield an approximate solution x_k. We propose to use ADMM and refer the reader to subsection 3.3 for details
(3) **Termination test:** Compute δ_k per equation (12). If $\delta_k \leq tol_\delta$, terminate and return y_k as the solution.
(4) **Oracle call:** Call the oracle (subsection 3.2) at the candidate point x_k and recover g_k and s_k such that $g(.) \geq g_k + s_k^\top(. - x_k)$.
(5) **Update test:** If $h_{y_k}(y_k) - h_{y_k}(x_k) \geq m\delta_k$, serious step. Define $y_{k+1} = x_k$ and update H_{k+1} as in equation (18). Otherwise, null step and $y_{k+1} = y_k$.
(6) **Bundle management:** $\mathcal{B}_{k+1} \leftarrow \mathcal{B}_k \cup \{(x_{k+1}, f_{k+1}, g_{k+1}, s_{k+1})\}$. Some variations on that step are discussed in subsection 3.2. Update H_k by adding the new hyperplane to G_k.
(7) $k \leftarrow k + 1$ and go to step (2)

Termination of this algorithm is discussed and proved in Sagastizábal and Solodov (2005).

3.2 Implementation of the oracle

We are looking for a subgradient of the function g defined in (20) at the current point x_k. Subgradients of g are closely related to the worst-case scenario of the uncertainty in the maximization (20). Similarly to Mutapcic and Boyd (2009), an exact oracle will need to compute a maximizer $z^\star$ in the maximization in Equation (20). A subgradient s of $l(., z^\star)$ at x_k will then be a subgradient of g at $z^\star$. Indeed we have:

$$\begin{aligned} \forall y,\, g(y) \geq l(y, z^\star) &\geq l(x_k, z^\star) + s^\top(y - x_k) \\ &= g(x_k) + s^\top(y - x_k) \end{aligned} \tag{21}$$

Remark 1. In order to have an efficient oracle, we need to able to compute the maximizer quickly. If l is concave in z and $\mathcal{Z}$ is convex, the problem will be a convex optimization problem and the maximizer will be computed efficiently. These assumptions are similar to the ones required in order to be able to reformulate the robust constraints using Theorem 1 to the robust inequality in (19). Therefore in this case, both methods are alternative ways to solve the same problem.

Remark 2. As discussed in Mutapcic and Boyd (2009), if a global maximizer is not realistically available for $z^\star$, then requirements can be relaxed. The oracle will aim at producing a scenario that strictly violates constraints. Based on that point, a cutting plane can be added to the model which will trigger improvement at the next iteration. Of course convergence to the overall optimum can no longer be guaranteed.

Oftentimes, the robust optimization problem is subject to multiple robust constraints, so that $l(x,z) = \max_i l_i(x,z)$ in (19). The following remarks are in order:

- Even if each l_i is concave in z, l will generally not be concave in z, but the maximum of l can be evaluated as follows:
  (1) For $i = 1, \ldots, r$, compute $z_i^\star = \arg\min_{z \in \mathcal{Z}} l_i(x_k, z)$ and $l_i^\star = l_i(x_k, z_i^\star)$.
  (2) Find index j such that $j = \arg\min_i l_i^\star$
  (3) Compute $s \in \partial l_j(x_k, z_j^\star)$
  (4) Return (l_j, s).
  Step (1) is most expensive and consists in solving m convex optimization problems in dimension n_z, which can be performed in parallel.
- We can add more than one lower bounding cutting plane per iteration. It is actually efficient to add multiple cuts based on the worst case violation for each robust constraint l_i. The intuition for that is that if the current iterate violates constraint i considerably for a particular scenario z_i, it is somewhat likely that z_i is also a pessimizer for the solution of the problem. In this case, the oracle returns n_s lower bounding planes, for example for the n_s most violated constraints.

3.3 Minimization step

One of the core principles of bundle methods is that the minimization problem (11) takes a relatively simple form, typically a quadratic program. For example, if f is linear, then Problem (11) can be transformed into a quadratic program (QP). Note that from one iteration to the next, the algorithm solves similar quadratic programs: during null steps, only the model of the improvement function is enriched, which adds new constraints to the minimization subproblem; during serious steps, the constraint corresponding to the cost in (22) and the linear part of the cost are updated. Strategies have been proposed to solve this succession of QPs efficiently, *e.g.* (Kiwiel, 1991). We perform an approximate minimization, keeping the following points in mind:

- The subgradients produced by the oracle may not define the "best" lower bounding planes to add, but they will not introduce errors in the lower bounding model G.
- A low accuracy solution is likely to identify the same pessimizers in the uncertainty set as the exact solution would.

We propose to use ADMM (Boyd et al., 2011) to solve the minimization problem, which allows to get low accuracy solutions quickly for large problems, exploiting favorable structure and the opportunity to warm start successive iterations. Assuming a linear cost function $f(x) = c^\top x$ and introducing slack variables σ, problem (11) becomes:

$$x_{k+1} = \text{argmin}_{x,\sigma}\{ \max_i \sigma_i + \frac{1}{2c_k}\|x - y_k\|^2\} \quad \text{subject to} \quad Sx + m = \sigma \tag{22}$$

where

$$S = \begin{pmatrix} c^\top \\ s_1^\top \\ \vdots \\ s_k^\top \end{pmatrix} \quad \text{and} \quad m = \begin{pmatrix} -c^\top y_k \\ g_1 - s_1^\top x_1 \\ \vdots \\ g_k - s_k^\top x_k \end{pmatrix}$$

The steps of the ADMM algorithm are (Boyd et al., 2011):

$$x_{t+1} = \text{argmin}_x \frac{1}{2c_k}\|x-y_k\|^2+\frac{\rho}{2}\|Sx+m-\sigma_t+\mu_t\|^2 \tag{23}$$

$$\sigma_{t+1} = \text{argmin}_\sigma \max_i \sigma_i + \frac{\rho}{2}\|Sx_{t+1} + m - \sigma_t + \mu_t\|^2 \tag{24}$$

$$\mu_{t+1} = \mu_t + Sx_{t+1} + m - \sigma_{t+1} \tag{25}$$

where μ is the scaled Lagrange multipliers associated with the equality constraints.

Step (23) can be solved efficiently by precomputing a factorization of $M = \rho S^\top S + \frac{1}{c_k}\mathcal{I}_{n_x}$ and performing only forward backward solves online. The second step is a proximal step for the function $k(\sigma) = \max_i \sigma_i$. It can be performed efficiently (Parikh and Boyd, 2014) as follows: by bisection, find the solution $t^\star$ to the scalar equation:

$$\sum_{i=1}^{k} \rho(v_i - t^\star)^+$$

where $v_i = (Sx + m + \mu)_i$. Then, $\sigma_i = \min(t^\star, v_i)$. This makes both steps very computationally efficient, which combined with the ability to warm start successive iterations provide a good performance.

Remark 3. Affine equalities $Ax = b$ can easily be incorporated in the problem by adding them to Problem (11). This would result in step (23) being an equality constrained QP.

Remark 4. If the original problems contains generic conic constraints of the form $Az \leq_\mathcal{K} b$, meaning that $b - Ax \in \mathcal{K}$ with $\mathcal{K}$ a convex cone, ADMM can handle those in an almost identical fashion provided an efficient projection algorithm on the cone $\mathcal{K}$ is available.

3.4 Approximate solution to the minimization step

Supposing the minimization step is solved with a tolerance tol_h so that x_k satisfies:

$$h_k(x_k) - h_k(x_k^*) \leq tol_h \tag{26}$$

where x_k is the solution proposed by our inexact minimization and x_k^* is the true solution to Problem (11). Then, $h_k(y_k)-h_k(x_k^*) = h_k(y_k)-h_k(x_k)+h_k(x_k)-h_k(x_k^*) \leq \delta_k + tol_h$. Therefore, the termination criterion $\delta_k \leq tol_\delta$ yields $\delta_k^* \leq tol_\delta + tol_h$, where δ_k^* is the expected improvement for x_k*. This is also a valid termination criterion, but with a relaxed tolerance.

Following the presentation in Boyd et al. (2011), we introduce the primal and dual residuals within the ADMM algorithm:

$$\begin{aligned} r_1^{t+1} &= Sx_{t+1} + m - \sigma_{t+1} \\ r_2^{t+1} &= \rho S^\top(\sigma_{t+1} - \sigma_t) \end{aligned} \tag{27}$$

It can be shown that:

$$h_k(x_t) - p^\star \leq \rho\|\mu_t\|\|r_1^t\| + d\|r_2^t\| \tag{28}$$

where d is an upper bound on $\|z_t - z_k^*\|$ the distance of the current iterate to the optimal point for the current iteration. The termination criterion for the ADMM is:

$$\|r_1^t\|_2 \leq tol_{prim} \text{ and } \|r_2^t\|_2 \leq tol_{dual} \tag{29}$$

with tol_{prim} and tol_{dual} chosen so as to meet $\rho\|\mu_t\|_2\|r_1^t\| + d\|r_2^t\|_2 = tol_h$.

4. NUMERICAL RESULTS

We illustrate the method proposed and the performance of the algorithm proposed on a frequency regulation provision problem. In this section, bold letters are used to denote sequences over time, *e.g.*, $\mathbf{p} = [p_0^T, p_1^T, \ldots, p_{N-1}^T]^T$.

4.1 System description

We consider a three zones office building located in Chicago offering power consumption tracking to the grid by maniuplating its air conditionning power consumption. We assume that a linear state-space description of the building is available. We refer the reader to Gorecki et al. (2015a) for details on model construction for building systems.

$$\begin{cases} x_{i+1} &= Ax_i + B_u u_i + B_d d_i \\ y_i &= Cx_i \end{cases} \tag{30}$$

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ is the thermal input provided to each zone, d_i the disturbance affecting the system, summarizing the effect of weather and internal gains and $y_i \in \mathbb{R}^p$ is the room air temperature at time i.

We define comfort constraints at the level β as $|y_i - T_{\text{ref}}| \leq \beta$. The inputs are constrained to lie in a set which describes physical limitations for the equipment $u \in \mathbb{U}$. We define the *admissible input trajectories* set as:

$$\mathcal{U}(\bar{x}, \mathbf{d}) = \left\{ \mathbf{u} \middle| \begin{array}{l} x_{i+1} = Ax_i + B_u u_i + B_d d_i \\ y_i = Cx_i \\ |y_i - T_{\text{ref}}| \leq \beta \\ u_i \in \mathbb{U} \\ x_0 = \bar{x}, \\ \forall i = 0, \ldots, N-1, \end{array} \right\} \tag{31}$$

where $\bar{x}$ is the current estimate of the state of the building, N the prediction horizon. $\mathcal{U}(\bar{x}, \mathbf{d})$ represents the set of all the input trajectories that preserve occupants comfort while respecting physical limits of the actuators.

4.2 The problem of frequency regulation provision

In order to regulate the frequency of the network, the TSO pays ancillary service providers (ASP) for "flexibility" in their power consumption. As part of grid operation, the TSO dispatches a tracking signal called the AGC signal to the ASPs, which make sure their power consumption follows their pre-purchased power consumption with the addition of the AGC signal. The tracking service procurement consists of advertising two quantities, namely a *baseline* energy consumption $\bar{\mathbf{p}}$, and a *capacity bid* γ. The latter represents the highest deviation (in absolute value) in power consumption with respect to the purchased baseline the provider is ready to absorb. If a bid of $\gamma = 1MW$ is accepted, the AGC signal dispatched to that provider will lie in the band $\pm 1MW$. The following tracking constraints are enforced:

$$|p_i - \bar{p}_i - \gamma a_i| \leq \alpha\gamma \quad (32)$$

where $\bar{p}_i$ is the baseline purchased for timestep i, a_i the normalized AGC signal and γ is the capacity bid. α represents the maximum tracking error allowed, as a percentage of the total bid submitted.

It is assumed that the power consumption is a known function of the inputs $p = h(u)$. The building receives a fixed payment in accordance to its accepted bid. For the sake of demonstration we look for the maximum flexibility the building can provide. We solve the following optimization problem:

Problem 1. (Reserve Scheduling Problem).

$$\begin{aligned}
&\text{maximize} && \gamma && \\
&\text{s.t.} && \forall \mathbf{a} \in \Xi, && (33)\\
&\text{(Building Contraints)} && \mathbf{u} \in \mathcal{U}(x_0, \mathbf{d}), && (34)\\
&\text{(Recourse policy)} && \mathbf{u} = \pi(\mathbf{a}), && (35)\\
&\text{(Power Consumption)} && \mathbf{p} = h(\mathbf{u}), && (36)\\
&\text{(Power tracking)} && \|\mathbf{p} - \bar{\mathbf{p}} - \gamma\mathbf{a}\|_\infty \leq \alpha\gamma && (37)
\end{aligned}$$

The decision variables are the capacity bid γ, the day-ahead baseline consumption $\bar{\mathbf{p}}$, and the control policy π. x_0 and $\mathbf{d}$ are data of the problem and represent the initial condition and the prediction for the disturbances. Equation (35) expresses the fact that the inputs can be chosen as a function of the uncertain parameter $\mathbf{a}$, as it becomes known. This formulation falls in the category of robust optimization problems.

We use the popular affine parametrization of the policies (Ben-Tal, A. et al., 2009) due to their nice trade-off between performance and computational properties. It results in $\mathbf{u} = \mathbf{M}\mathbf{a} + \mathbf{v}$, where $\mathbf{M}$ and $\mathbf{v}$ are the decision variables. In order to ensure that the policies are causal, appropriate constraints on $\mathbf{M}$ are imposed so that u_i can depend on a_i up to time i:

$$M_{i,j} = 0 \text{ for } j > i$$

For the design of the uncertainty set is discussed, a data-driven design is chosen. We build the set Ξ as follows:

$$\Xi = \{\mathbf{a} \,|\, \|\mathbf{a}\|_\infty \leq 1, \|\operatorname{cumsum}(\mathbf{a})\|_\infty \leq s_{\max}\} \quad (38)$$

where $\operatorname{cumsum}(\mathbf{a})$ denotes the cumulative sum of the signal $\mathbf{a}$ ($(\operatorname{cumsum}(\mathbf{a}))_i = \sum_{k=1}^{i} a_k$). $s_{\max}$ is chosen as the maximum integral on a number N_s of previously observed AGC realizations. Extensive details on the design procedure are provided in Fabietti et al. (2016). Ξ then describes the feasible input set of an ideal electric battery, therefore it allows us to abstract the building as a virtual electric storage.

4.3 Solving the problem

One challenge of the above problem is that the problem needs to be solved over long periods of times. The problem with horizon N has $14N$ robust constraints to satisfy for a total number of $18N$ nontrivial decision variables.

- **Dual reformulation**: The problem is a robust linear program with a polytopic uncertainty set. A dual reformulation introduces $14N * 4N = 56N^2$ variables to form a linear program with $14N^2$ equality constraints, $14N$ inequality constraints and $56N^2$ non-negativity constraints. The problem obtained is a linear program, sometimes prohibitively large due to the large number of added variables. The program was formulated in Matlab, parsed with YALMIP (Löfberg, 2004) and solved using the LP solver CPLEX 12.6 with the dual simplex algorithm.
- **Bundle method with inexact minimization**: We solve the problem using the bundle method described in Section 3. The minimization step is solved using ADMM. Successive minimization are warm-started using the solution of the previous steps and the subproblems are solved to low accuracy. The oracle is implemented using Gurobi 6.5 and using a pool of 4 processors for parallel computations.

Parameter	Value	Parameter	Value
m	0.3	c_k	300
n_s	200	tol_{prim}	10^{-2}
tol_{dual}	10^{-2}	tol_δ	10^{-1}
tol_g	3.10^{-4}	$s_{\max}$	3

Table 1. Parameter values in the algorithm

Horizon	50	100	150	200	250
# variables	1020	2070	3120	4170	5220
# bundle iterations	16	29	40	46	50
# constraints added	2896	5468	7793	9400	9672
Total time (s)	54	631	1910	3885	6511
Oracle time (% of total)	12	7.5	6	12	18

Table 2. Bundle execution details

Computations are performed on a workstation with a 2.7 GHz Intel Core i7 processor 4 Gb of memory. Table 1 recaps the parameters used in the bundle algorithm.

We report computational times in Figure 1 as a function of the horizon. The computational time for the reformulation grows exponentially with the size of the problem and the problem fails to solve above horizon superior to 100 due to memory excess issues in YALMIP. With the bundle method, a solution to the problem can be found for horizons up to 250. The computational time for the bundle does not grow exponentially with the size of the problem. This cannot be claimed to be a characteristic of the method, but it suggests that the algorithm is capable of identifying the most critical constraints efficiently. This is reflected in the number of bundle iterations reported in Table 2 and the number of constraints generated, which roughly grows linearly with the size of the problem. Most of the computational time is spent in the minimization step, but we can observe that the oracle evaluations represent a significant part of the runtime as well, hence the advantage of parallelizing the computations for this step.

Remark 5. The dual formulation implementation solves the problem to a higher precision (to ensure feasibility). Also, a more efficient implementation of the dual reformulation is possible, so the comparison should be considered with caution. The key observation is that the bundle method requires a linear number of outer iterations to get a solution, in contrast to the simplex method whose solve time seem to grow faster with the horizon.

Figure 2 shows the behavior of the building when following tracking signals coming from the set described in (38). One day of tracking sampled at 15 minutes is depicted.

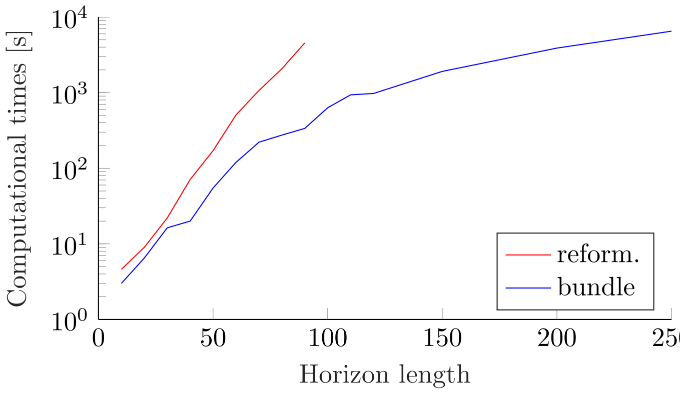

Fig. 1. Computational times as a function of the horizon

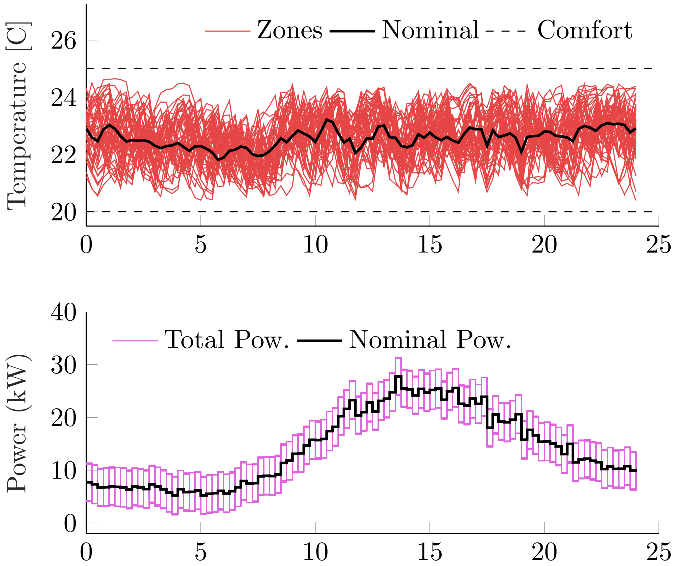

Fig. 2. Temperatures in the building [top] and total power consumption [bottom] in response to 15 random tracking signals. Nominal values in black.

15 random scenarios of the uncertainty were selected as vertices of the set and scaled by the optimal flexibility. The open-loop controller computed is applied to generate trajectories. The resulting temperature and power consumption trajectories are depicted. Due to the high tolerance used to compute the solution, the solution of the bundle is suboptimal. On instances where the exact solution could be computed through reformulation, an average suboptimality of 3% is observed. The solution is however always strictly feasible due to the monotonic nature of the constrained bundle method with respect to feasibility.

5. CONCLUSION

We propose a constrained bundle method to solve large scale robust optimization problems, relying on approximate solutions of the minimization steps. The method is demonstrated on a large multi-stage energy planning problem. Further work will include a more comprehensive study of the computational advantage of this method, and a detailed analysis of the interplay between the inner loop of ADMM and the outer bundle loop.

ACKNOWLEDGEMENTS

We are grateful to Harsh Shukla for his support.

REFERENCES

Ben-Tal, A., Hertog, D.d., and Vial, J.P. (2014). Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2), 265–299. doi:10.1007/s10107-014-0750-8.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*. Princeton University Press.

Bertsimas, D., Dunning, I., and Lubin, M. (2015). Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2), 195–217. doi:10.1007/s10287-015-0236-z.

Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., and Sagastizábal, C.A. (2006). *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1), 1–122. doi:10.1561/2200000016.

Fabietti, L., Gorecki, T., Qureshi, F., Bitlislioglu, A., Lymperopoulos, I., and Jones, C. (2016). Experimental Implementation of Frequency Regulation Services Using Commercial Buildings. *IEEE Transactions on Smart Grid*, PP(99), 1–1. doi:10.1109/TSG.2016.2597002.

Gorecki, T.T., Qureshi, F.A., and Jones, C.N. (2015a). OpenBuild : An integrated simulation environment for building control. In *2015 IEEE Conference on Control Applications (CCA)*, 1522–1527. doi:10.1109/CCA.2015.7320826.

Gorecki, T., Bitlisliolu, A., Stathopoulos, G., and Jones, C.N. (2015b). Guaranteeing input tracking for constrained systems: theory and application to demand response. In *the 2015 American Control Conference (ACC)*.

Kiwiel, K.C. (1991). A Cholesky dual method for proximal piecewise linear programming. *Numerische Mathematik*, 68(3), 325–340. doi:10.1007/s002110050065.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*. Taipei, Taiwan.

Mutapcic, A. and Boyd, S. (2009). Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods and Software*, 24(3), 381–406. doi:10.1080/10556780802712889.

Parikh, N. and Boyd, S. (2014). Proximal Algorithms. *Found. Trends Optim.*, 1(3), 127–239. doi:10.1561/2400000003.

Sagastizábal, C. and Solodov, M. (2005). An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, 16(1), 146–169.

Zhang, X., Kamgarpour, M., Goulart, P., and Lygeros, J. (2014). Selling robustness margins: A framework for optimizing reserve capacities for linear systems. In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, 6419–6424. doi:10.1109/CDC.2014.7040396.