Area C
Total pages 19
Date October 16, 2013

Nonlinear programming with applications to production processes

graph
robot

Torsten Bosse Andreas Griewank René Henrion Dietmar Hömberg
Chantal Landry Hernan Leövey Werner Römisch

1 Introduction

Nonlinear programming is a key technology for finding optimal decisions in production processes. It applies to optimal control as well as to operations research, to deterministic as well as to stochastic models. The efficient solution of nonlinear programs requires both, a good structural understanding of the underlying optimization problems and the use of tailored algorithmic approaches mainly based on SQP methods. The present chapter provides an account of the work in three MATHEON-projects with various applications and aspects of nonlinear programming in production.

2 An optimal control problem in automotive industry

2.1 Background

Automotive industry has by now reached a high degree of automation. Complex production lines must have been created. These lines are divided into workcells, which are composed of a workpiece, several robots and some obstacles. The robots perform tasks on the workpiece before the piece is moved to the next workcell.

Efficient production lines are essential to ensure the competitiveness of automotive industry. For that purpose, the manufacturer must minimize the time taken to complete all the tasks in a workcell, that is the *makespan*. The goal of the MATHEON project "Automatic reconfiguration of robotic welding cells" is to design an algorithm which minimizes the makespan. Given the Computer Aided Design (CAD) data of the workpiece, the location of the tasks and the number of robots, the aim is to assign tasks to the different robots and to decide in which order the tasks are executed as well as how the robots move to the next task such that the makespan is minimized. We call this problem the WorkCell Problem (WCP).

As presented in [34], the (WCP) can be modeled as a graph. The nodes of the graph are the task locations and the initial location of the end effector of the robots. An arc exists for a robot if and only if the robot can move between the nodes which form the arc. Finally, a weight is associated with each arc. This weight is the traversal time used by the robot to join the endpoints of the arc. The (WCP) is an instance of vehicle routing problem and is solved with column generation and resource constraint shortest path as the pricing subproblem, see [41] for more details.

In the (WCP), the crucial information is the weight of the arcs, namely the traversal time for the robot to join the source node of the arc to its target node.

path-planning
collision
optimal control problem

These times are obtained when calculating the path-planning of the robot to join the nodes defining the arc. This motion must be as fast as possible and without collision with the obstacles of the workcell. The collision between the robots is not tested during the computation of the path-planning, but is checked during the computation of the scheduled tours, as explained in [34]. The computation of the motion of the robot and the associated traversal times is presented in the next sections.

2.2 Optimal control problem

In this section, we present a model to compute the path-planning of a robot. Let us consider a robot composed of p links, which are connected by revolute joints. Let $q = (q_1, \ldots, q_p)$ denote the vector of joint angles of the robot. Moreover, let $v = (v_1, \ldots, v_m)$ contain the joint angle velocities and let $u = (u_1, \ldots, u_m)$ describe the torques applied at the center of gravity of each link.

The robot is asked to move as fast as possible from a given position to a desire location. Its motion is given in the Lagrangian form as follows

$$q^{'}(t) = v(t) \text{ and } M(q(t))\, v^{'}(t) = G(q(t), v(t)) + F(q(t), u(t)), \tag{1}$$

where $M(q)$ is the symmetric and positive definite mass matrix, $G(q, v)$ contains the generalized Coriolis forces and $F(q, u)$ is the vector of applied joint torques and gravity forces. The function F is linear in u.

The motion of the robot must follow (1), but also be collision-free with the obstacles of the workcell. For simplicity, let us assume that only one obstacle exists. To establish the collision avoidance condition, the robot and the obstacle are approximated by a union of convex polyhedra. The approximation is denoted by P for the robot, by Q for the obstacle and are given by $P = \cup_{i=1}^{p} P^{(i)}$, with $P^{(i)} = \{x \in \mathbb{R}^3 | A^{(i)} x \leq b^{(i)}\}$ and $Q = \cup_{j=1}^{q} Q^{(j)}$, with $Q^{(j)} = \{x \in \mathbb{R}^3 | C^{(j)} x \leq d^{(j)}\}$, where $A^{(i)} \in \mathbb{R}^{p_i \times 3}$, $b^{(i)} \in \mathbb{R}^{p_i}$, $C^{(j)} \in \mathbb{R}^{q_j \times 3}$, $d^{(j)} \in \mathbb{R}^{q_j}$, and p_i and q_j are the number of faces in $P^{(i)}$ and $Q^{(j)}$, respectively.

There exist several techniques to characterize the collision avoidance between the robot and the obstacle. One natural way is to require that the distance between the objects remains bigger than a safety margin. However, the computation of the distance is complex, in particular when the objects are intersecting [13]. Moreover, the distance function is non-differentiable in general. Instead, we develop the following formulation whose derivative is simple to obtain:
The robot P and the obstacle Q do not collide if and only if for each pair of polyhedra $(P^{(i)}, Q^{(j)})$, $i = 1, \ldots, p$, $j = 1, \ldots, q$, there exists a vector $w^{(i,j)} \in R^{p_i + q_j}$ such that:

$$w^{(i,j)} \geq 0, \left(\begin{array}{c} A^{(i)} \\ C^{(j)} \end{array} \right)^T w^{(i,j)} = 0 \text{ and } \left(\begin{array}{c} b^{(i)} \\ d^{(j)} \end{array} \right)^T w^{(i,j)} < 0. \tag{2}$$

This is a direct consequence of Farkas's lemma, see [12] for more details.

The fastest trajectory of a robot is the solution of an optimal control problem where the system of ordinary differential equations (ODE) is given by (1), see [7]. If an obstacle is present in the workcell, the collision avoidance is guaranteed as soon as the vector $w^{(i,j)}$ of (2) is found at each time t and for all pairs of polyhedra.

SQP active set

However, to be written as state constraints, the strict inequality in (2) has to be relaxed. Furthermore, since the robot moves, the matrices $A^{(i)}$ and the vectors $b^{(i)}$ evolve in time. Their evolution depends explicitly on $q(t)$. A complete formulation of $A^{(i)}(q(t))$ and $b^{(i)}(q(t))$ is given in [12].

Thus, the optimal control problem to find the fastest collision-free trajectory is: (OCP): *Find the traversal time* t_f, *the state variables* q, $v : [0, t_f] \to R^p$, *and the controls* $u : [0, t_f] \to R^p$ *and* $w^{(i,j)} : [0, t_f] \to R^{p_i+q_j}$, $i = 1, \ldots, p$, $j = 1, \ldots, q$, *such that* t_f *is minimized subject to*
1) *ODE:* $q'(t) = v(t)$ *and* $v'(t) = M(q(t))^{-1}\left(G(q(t), v(t)) + F(q(t), u(t))\right)$;
2) *state constraints for* $i = 1, \ldots, p$, $j = 1, \ldots, q$:

$$\begin{pmatrix} A^{(i)}(q(t)) \\ C^{(j)} \end{pmatrix}^T w^{(i,j)}(t) \;=\; 0 \text{ and } \begin{pmatrix} b^{(i)}(q(t)) \\ d^{(j)} \end{pmatrix}^T w^{(i,j)}(t) \;\leq\; -\varepsilon, \tag{3}$$

3) *boundary conditions:* $R(q(0)) - V_0 = 0$, $v(0) = 0$, $R(q(t_f)) - V_f = 0$ *and* $v(t_f) = 0$;
4) *box constraints:* $u_{\min} \leq u \leq u_{\max}$ *and* $0 \leq w^{(i,j)}$, $i = 1, \ldots, p$, $j = 1, \ldots, q$,
where $R(q)$ denotes the position of the end effector of the robot and V_0, V_f are the given task locations. The vectors u_{min} and u_{max} are given as well as the relaxation parameter ε, which is positive and small.

Depending on the number of state constraints (3), the problem is inherently sparse since the artificial control variables $w^{(i,j)}$ do not enter the dynamics, the boundary conditions, and the objective function of the problem, but only appear linearly in (3).

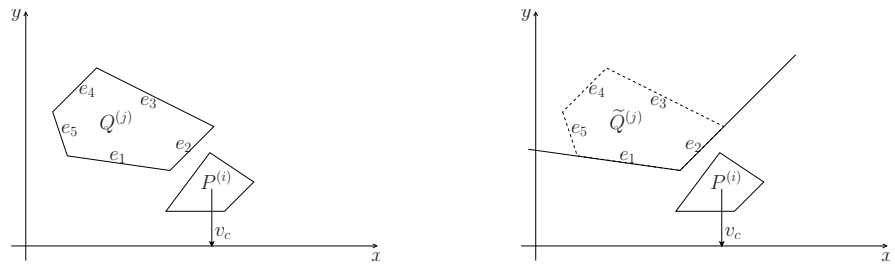
(OCP) can be easily applied with several obstacles. It suffices to define new artificial control variables and to write (3) for each obstacle.

2.3 Numerical method and results

We solve (OCP) with a reduced discretization approach. The method involves first discretizing the control problem and transforming it into a finite-dimensional nonlinear optimization problem. The control variables are approximated by B-splines, which are defined on an equidistant grid. A one-step method is used to integrate the ODE.

In a second time, the resulting nonlinear optimization problem is solved by a sequential quadratic programming (SQP) method [14]. As in [38], we use an Armijo type line-search procedure for the augmented Lagrangian function in our implementation. Moreover, we use BFGS update formula [3] instead of the exact Hessian matrix of the Lagrangian function. This formula guarantees that the Hessian matrix remains symmetric and positive definite. Thus, the quadratic subproblems in SQP are strictly convex.

The resulting optimization problem contains a lot of constraints. Indeed, at each time step of the control grid and for all pairs of polyhedra $(P^{(i)}, Q^{(j)})$, four state constraints are defined (compare (3)). To reduce the number of constraints and variables, we add an active set strategy based on the following observation: the state constraints are superfluous when the robot is far from the obstacle or moves in the opposite direction.



(a) (b)

Figure 1: (a) The polyhedron $P^{(i)}$ is moving downwards. The faces of $Q^{(j)}$ are denoted by $e_1, \ldots, e_5$. (b) The set $\tilde{Q}^{(j)}$ is generated by the faces e_1 and e_2 of $Q^{(j)}$.

If $P^{(i)}$ is far from $Q^{(j)}$ at time step t_k, then no state constraint is defined at t_k. Let us assume now that $P^{(i)}$ is close to $Q^{(j)}$ and consider the situation depicted in Figure 1: $P^{(i)}$ is moving downwards, v_c indicates the velocity of $P^{(i)}$ and $\tilde{Q}^{(j)}$ is generated by the faces e_1 and e_2 of $Q^{(j)}$. According to (2), $\tilde{Q}^{(j)}$ does not intersect $P^{(i)}$ if and only if

$$\exists \tilde{w}^{(i,j)} \geq 0, \text{ such that } \begin{pmatrix} A^{(i)} \\ C^{(j)}_{1,2} \end{pmatrix}^\top \tilde{w}^{(i,j)} = 0 \text{ and } \begin{pmatrix} b^{(i)} \\ d^{(j)}_{1,2} \end{pmatrix}^\top \tilde{w}^{(i,j)} < 0, \quad (4)$$

where $C^{(j)}_{1,2}$ is the matrix composed of the first two rows of $C^{(j)}$ and $d^{(j)}_{1,2}$ is the vector composed of the first two components of $d^{(j)}$.

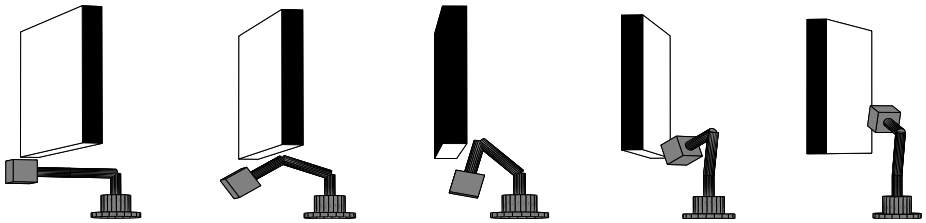
Suppose now that $\tilde{w}^{(i,j)}$ exists. By setting $w^{(i,j)} = (\tilde{w}^{(i,j)}, 0, 0, 0)$, we obtain:

$$\begin{pmatrix} A^{(i)} \\ C^{(j)} \end{pmatrix}^\top w^{(i,j)} = 0 \text{ and } \begin{pmatrix} b^{(i)} \\ d^{(j)} \end{pmatrix}^\top w^{(i,j)} < 0.$$

Then, (2) implies that $P^{(i)}$ and $Q^{(j)}$ do not intersect. In summary, if no collision occurs between $\tilde{Q}^{(j)}$ and $P^{(i)}$, then $Q^{(j)}$ and $P^{(i)}$ do not intersect. The dimension of $\tilde{w}^{(i,j)}$ is always smaller than that of $w^{(i,j)}$. Thus, the number of unknowns decrease when the state constraints are replaced by (4). The active set strategy is fully detailed in [12, 33].

A numerical example is presented in Figure 2. The robot is composed of three links. At the end of the last link, a load is fixed. Here, it is sufficient to apply the state constraints only between the load and the obstacle to have a collision-free motion. In Figure 2, the faces of the obstacle that are considered in the state constraints are white. We can observe that only three faces of the obstacle are taken into consideration. The computational time is 52 s. If we do not use the active set strategy, the computational time is about four times longer.

In conclusion, an optimal control problem was defined to find the fastest collision-free motion of an industrial robot. Farkas's lemma allowed us to state the collision avoidance as an algebraic formulation whose derivative is simple to obtain. An active set strategy was developed to speed up the SQP method. This strategy also keeps the size of the quadratic subproblems low when the robot and the obstacles have complex geometry.



electric power

Figure 2: Snapshots of the motion of the robot avoiding an obstacle. The faces of the obstacle that are considered in the state constraints are white.

3 Stochastic optimization models for electricity production in liberalized markets

3.1 An optimization model

The operation of electric power companies is often substantially influenced by a number of uncertain quantities like uncertain load, fuel and electricity spot and derivative market prices, water inflows to reservoirs or hydro units, wind speed etc. In particular, mid-term operation planning involves many risks and represents an enormous challenge. Here, we take the view of a (market) price-taking retailer owning a generation system and participating in the electricity market. This means that its operation does not influence market prices. A typical example is a municipal power company that intends to maximize revenue and whose operation system consists of thermal and/or hydro units, wind turbines and a number of contracts including long-term bilateral contracts, day ahead trading of electricity and trading of derivatives.

It is assumed that the time horizon is discretized into uniform (e.g., hourly) intervals. Let T, I, J, K and L denote the index sets of time periods, thermal units, hydro units, wind turbines and contracts, respectively. For thermal unit $i \in I$ in period t, $u_{it} \in \{0,1\}$ denotes its commitment decision (1 if on, 0 if off), x_{it} its production level and U_i the polyhedral set of all pairs (x_{it}, u_{it}) satisfying the capacity and minimum up/down-time constraints for all time periods $t \in T$ (for a detailed description of such constraints see e.g [19]). Similarly, y_{jt}, $j \in J$, and z_{kt}, $k \in K$, are the production levels of hydro and wind units, respectively, and w_{lt}, $l \in L$, the contract levels for all time periods $t \in T$. While the production decisions x_{it} and z_{kt} are always positive, the decisions y_{jt} and w_{lt} may also be negative to model pumping in case of pumped hydro units and delivery contracts, respectively.

By $\xi = (\xi_t)_{t\in T}$ we denote the stochastic input process on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, whose components may contain market prices, demands, inflows and wind speed. Typically, reliable forecasts for the inputs are available for the first t_1 time periods and, hence, the decisions at those periods are deterministic (thus, measurable with respect to the trivial σ-field $\{\emptyset, \Omega\}$.

The constraint sets of hydro units and wind turbines may then depend on ξ and, hence, are denoted by $Y_j(\xi)$ and $Z_k(\xi)$, respectively. They are assumed to be polyhedral with stochasticity appearing on right-hand side of linear constraints.

multi-stage stochastic program
scenario tree

Basic system requirements are to satisfy the electricity demand $d_t(\xi)$ and (possibly) certain reserve constraints during all time periods $t \in T$. The demand constraints are of the form

$$\sum_{i\in I} x_{it} + \sum_{j\in J} y_{jt} + \sum_{k\in K} z_{kt} + \sum_{l\in L} w_{lt} \geq d_t(\xi) \quad (t \in T)$$

and the reserve constraints are imposed to compensate sudden demand peaks or unforeseen unit outages by requiring that the totally available capacity should exceed the demand in every time period by a certain amount (e.g. a fraction of the demand).

The expected total revenue is given by the expected revenue of the contracts reduced by the expected costs of all thermal units over the whole time horizon, i.e.,

$$\mathbb{E}(R(x,u,y,z,w,\xi)) = \sum_{t\in T} \mathbb{E}\Big(\sum_{l\in L} p_{lt}(\xi) w_{lt} - \sum_{i\in I} C_{it}(x_{it}, u_i, \xi)\Big), \tag{5}$$

where we assume that the operation costs of hydro and wind units are negligible during the considered time horizon. The costs C_{it} for operating thermal unit i during period t consist of fuel and startup costs. The corresponding cost functions are assumed to be piecewise linear convex whose coefficients are possibly stochastic (see [19] for an explicit formulation of thermal cost functions).

We assume that the stochastic decisions $(x_t, u_t, y_t, z_t, w_t)$ are nonanticipative, i.e., they only depend on $(\xi_1, \ldots, \xi_t)$ and, hence, are measurable with respect to the σ-field $\mathcal{F}_t = \sigma(\xi_1, \ldots, \xi_t)$ which is contained in $\mathcal{F}$.

Then the objective consists in maximizing the expected total revenue (5) such that the decisions are nonanticipative and the operational constraints $(x_i, u_i) \in U_i$, $i \in I$, $y_j \in Y_j(\xi)$, $j \in J$, $z_k \in Z_k(\xi)$, the demand and reserve constraints and (eventually) certain linear trading constraints are satisfied. Altogether, the model represents a *multi-stage mixed-integer linear stochastic program* that is large scale in many practical situations (notice that mid-term models range from several days up to one year; hourly discretization then leads to a cardinality $|T|$ from about 100 to 8760).

3.2 Scenario trees and solution methods

Often historical data is available for the stochastic input process and a statistical model for ξ may be derived (for example, via time series analysis (see, e.g., [10, 40]). The next step consists in generating scenarios, i.e., possible realizations of ξ. An overview of methods for generating scenarios ranging from Monte Carlo and Quasi-Monte Carlo methods to optimal quantization and sparse grid techniques is given in [37, Section 1.5.4]. Recent developments in high-dimensional numerical integration [6] suggest that recently developed randomized Quasi-Monte Carlo methods have excellent convergence properties. Starting with scenarios obtained by one of those ways and applying stability-based scenario tree generation techniques from [25, 23] then leads to a scenario tree approximation ξ_{tr} of the input process ξ. If the number of decision variables and constraints is too large when inserting ξ_{tr}, the tree dimension may be reduced appropriately to arrive at a moderate dimension (see [24]).

A scenario tree is given by a finite set $\mathcal{N}$ of nodes with root node $n = 1$ at the first time period, a mapping t from $\mathcal{N}$ onto T and by requiring that every node $n \in \mathcal{N}$ has a unique predecessor n_- except $n = 1$. The number $t(n) - 1$ corresponds to the number of successive predecessors of n to reach the root node. If $\mathcal{N}_+(n)$ denotes the set of successors to $n \in \mathcal{N}$, we set $\mathcal{N}_T = \{n \in \mathcal{N} : \mathcal{N}_+(n) = \emptyset\}$. For every $n \in \mathcal{N}_T$, the set $\{1, \ldots, n_-, n\}$ is called *scenario* and π_n denotes its probability. A probability is assigned to each node n by setting recursively $\pi_n = \sum_{n_+ \in \mathcal{N}_+(n)} \pi_{n_+}$.

The scenario tree ξ_{tr} then consists of all ξ^n, $n \in \mathcal{N}$, where ξ^n is a realization of $\xi_{t(n)}$. The decisions in the tree formulation of the stochastic optimization model are $(x_i^n, u_i^n, y_j^n, z_k^n, w_l^n)$ and the expected revenue is of the form

$$\sum_{n \in \mathcal{N}} R(x^n, u^n, y^n, z^n, w^n, \xi^n) = \sum_{n \in \mathcal{N}} \Big(\sum_{l \in L} p_{lt}(\xi^n) w^n_{lt(n)} - \sum_{i \in I} C_{it}(x^n_{it(n)}, u_i^n, \xi^n) \Big).$$

Then the objective consists in maximizing the expected revenue subject to the operational constraints $(x_i^n, u_i^n) \in U_i$, $i \in I$, $y_j^n \in Y_j(\xi^n)$, $j \in J$, $z_k^n \in Z_k(\xi^n)$, the demand and reserve constraints and (eventually) certain linear trading constraints at every node. This tree formulation of the optimization model represents a mixed-integer linear program containing $|\mathcal{N}|(|I| + |J| + |K| + |L|)$ continuous and $|\mathcal{N}||I|$ binary variables and an extremely large number of constraints.

The numerical solution of such optimization models requires decomposition methods except in particular situations. We refer to the survey [39] for existing primal and dual decomposition approaches. Since the optimization models contains at most $2|\mathcal{N}|$ coupling constraints, Lagrangian relaxation of coupling constraints seems to be promising. In fact, as shown in [19], the dualization leads to a decomposition into unit and contract subproblems, respectively. Since there exist efficient solution algorithms for all subproblems (see e.g. [19]), dual function and subgradient evaluations are reasonable. Applying bundle subgradient methods for solving the dual then leads to an iterative coordination of the operation of all units. After finishing the bundle subgradient method the final Lagrangian solution violates in general the coupling demand and reserve constraints at some nodes $n \in \mathcal{N}$. Since the relative duality gaps are typically small for such models, simple problem-specific Lagrangian heuristics may be developed to modify the Lagrangian commitment decisions nodewise and to reach primal feasibility after finitely many steps of the heuristics. A final economic dispatch then leads to a good primal feasible solution (see also [19]).

3.3 Mean-risk objective

The revenue $R(x, u, y, z, w, \xi)$ depending on the decision (x, u, y, z, w) is a real random variable which often has a large variance if the decision is (nearly) optimal. Hence, the probability may be large that a perturbed decision leads to (much) smaller revenues than the expected revenue $\mathbb{E}(R(x, u, y, z, w, \xi))$, i.e., the optimal decision is risky. Since this effect is hardly acceptable, determining a decision based only on maximizing the expected revenue is unsuitable.

Alternatively, one might wish to measure the risk of a decision and to minimize or at least to bound the risk simultaneously when maximizing the expected

risk functional
probabilistic constraint

revenue. This idea leads to maximizing a so-called mean-risk objective of the form

$$(1-\gamma)\mathbb{E}(R(x,u,y,z,w,\xi)) - \gamma\rho(R(x,u,y,z,w,\xi))$$

where ρ is a convex risk functional (see [11]) and $\gamma \in (0,1)$ a risk parameter measuring the degree of risk aversion. Since a risk functional is always nonlinear, one might wish that the linearity structure of the optimization model is preserved. An additional aspect is that revenue represents a stochastic process that evolves over time $t \in T$. Hence, a so-called polyhedral multi-period risk functional ρ [9, 36, 20] might be an appropriate tool to be incorporated into the mean-risk objective, which then is of the form

$$(1-\gamma)\mathbb{E}(R(x,u,y,z,w,\xi)) - \gamma\rho((R_t)_{t\in T}),$$

where R_t is the portion of the revenue $R(x,u,y,z,w,\xi)$ obtained until t. In this way, risk managment is integrated into the model for maximizing the expected revenue and the scenario tree-based optimization model may be reformulated as a mixed-integer linear program as in the risk-neutral case $\gamma = 0$ (see [9, 8]).

3.4 Probabilistic constraints

As mentioned above, many optimization problems arising from power managment are affected by random parameters. In this case, the use of probabilistic constraints makes it possible to find optimal decisions which are robust against uncertainty at a specified probability level. More precisely a probabilistically constrained optimization problem has the form

$$\min\{f(x) | \mathbb{P}\left(g(x,\xi) \geq 0\right) \geq p\}, \tag{6}$$

where f is an objective depending on a decision vector x, ξ is a random vector defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and g is a mapping defining some random inequality system with several components. The probabilistic constraint defines a decision as feasible if the associated random inequality system is satisfied at probability at least $p \in (0,1]$. From a formal point of view, probabilistic constraints are conventional inequalities restricting the domain of feasible decisions. The major difficulty in their numerical treatment consists in the absence of explicit formulae for function values and gradients. At the same time, this difficulty leads to numerous challenges in the analysis of the structure and stability for such optimization problems. Therefore, a major task of our work consisted in improving our insight into essential properties like continuity, differentiability, convexity etc. and to exploit it for algorithmic purposes. The focus was on linear probabilistic constraints, where linear relates to the random vector in the mapping $g(x,\xi)$. As a consequence, two basic models have to be distinguished:

$$\mathbb{P}\left(h(x) - A\xi \geq 0\right) \geq p \quad \text{(separated model)} \tag{7}$$

$$\mathbb{P}\left(T(x)\xi \geq b(x)\right) \geq p \quad \text{(multiplicative model)} \tag{8}$$

In the following we give a compressed account of the obtained results:

Structural Properties

In [31] we investigated continuity and differentiability properties of the probability function $\varphi(x) := \mathbb{P}(x - A\xi \geq 0)$ which is the basis for (7). It was shown that for ξ having a so-called quasi-concave distribution, Lipschitz continuity of φ is equivalent with its simple continuity and both are equivalent to the fact that none of the components ξ_i has zero variance. Combining this with a Theorem by Borell one derives that φ is Lipschitz continuous if the s-dimensional random vector ξ possesses a density f such that $f^{-1/s}$ is convex. It was also shown that in the case of ξ having a nondegenerate Gaussian distribution φ is of class $\mathcal{C}^\infty$ around some $\bar{x}$ provided that the polyhedron $\{z|Az \leq \bar{x}\}$ is nondegenerate. The latter means that the active rows of A (satisfying $A_i z = \bar{x}_i$) are linearly independent which is a substantially weaker condition than surjectivity of A.

Convexity and compactness properties of probabilistic constraints were analysed in [26, 32]. Special attention was payed to sets of feasible decisions defined by a probabilistic constraint on a linear inequality system with stochastic coefficient matrix under Gaussian distribution:

$$M := \{x|\mathbb{P}(\Xi x \leq a) \geq p\}. \tag{9}$$

Note that (9) is a special instance of (8). It could be shown that M is compact for all probability levels p larger than a critical value p^* which can be calculated explicitly from the parameters of the distribution. Similarly, under the additional assumption that the rows of Ξ are independently distributed, it follows the convexity of M for $p \geq \bar{p}$ where again $\bar{p}$ follows explicitly from the parameters of the distribution. Evidently, both properties have importance for algorithmic solution approaches to probabilistically constrained programs. It is worth mentioning that the validity of these statements for sufficiently large p only is not restrictive as in practice p is chosen close to one.

Gradient Formulae

For an efficient solution of (6) one has to be able to provide values and gradients of the probability function $x \mapsto g(x, \xi) \geq 0)$. Already on the level of function values this is a challenging task requiring sophisticated techniques of numerical integration, (Quasi-) Monte Carlo methods, variance reduction techniques etc. In general, only approximations with a certain (modest) precision can be provided. Most promising results are obtained for the special separated structure $g(x, \xi) = \xi - x$ leading to the evaluation of multivariate distribution functions. Efficient methods for approximating such distribution functions have been reported, for instance, in the case of the Gaussian, Student, Dirichlet, Gamma or Exponential distribution. If there is no explicit formula available for probability functions, much less this is true for their gradients. Given the inaccuracy of values, a finite difference scheme appears to be inappropriate for approximating gradients. Interestingly, for certain distributions (e.g., Gaussian, Student) there exists an **analytic** reduction of gradients to values of the corresponding distribution functions (with possibly modified parameters). This allows to calculate gradients by the same methodology as function values without further increasing the inaccuracy of results. The possibility to

hydro reservoir

globally control the relative precision of gradients by the precision of function values independent of the concrete argument is discussed in [27] for a special class of correlation matrices. However, this reduction traditionally requires regularity of the correlation matrix which is not given in many important applications (for instance in (7) if the matrix A has more rows than columns). Therefore, a possible extension of gradient reduction in the case of singular covariance matrices has become a central aim of our analysis. We were successful in obtaining analytic reductions of gradients to distribution function values in the case of probability functions (7) (without surjectivity of A) and (8) under Gaussian distribution [28, 2].

Application to power managment problems

The theoretical results presented above were applied to the numerical solution of several problems of power managment with data primarily provided by *Electricité de France*. For the numerical solution of the nonlinear programs (6) we employed the supporting hyperplane method – which is slow but robust and provides bounds for the optimal value – as well as an SQP solver (SNOPT). We present just one example out of the spectrum of considered applications. It concerns the optimal short-term managment of a system of 6 serially linked hydro reservoirs under stochastic level constraints (a simplified version is described in [1]). The underlying optimization problem has the structure

$$max\{p^T x | \mathbb{P}(l_* \leq c - Ax + B\xi \leq l^*) \geq p,\ x \in X\}.$$

The objective consists in maximizing the profit made by selling turbined hydroenergy on a day-ahead market for a time horizon of two days discretized in time steps of 2 hours. The vector x represents the water released from the six reservoirs at each time step and, similarly, the random vector ξ models the discrete stochastic inflow processes to two of the reservoirs. p is a price signal such that $p^T x$ yields the profit by power production. The deterministic constraints $x \in X$ reflect lower and upper operational bounds for turbining. Apart from these constraints, one has to respect lower and upper bounds l_*, l^* in the reservoir for various, technological, ecological and sometimes even economical reasons. The vector $c - Ax + B\xi$ yields the current filling levels in the reservoir at each time step (c is a vector of initial levels and A, B model the accumulation of water inflows and releases as a function of the topology of the network). Given the stochastic nature of inflows (e.g., precipitation or snow melt), the level constraints are stochastic too. On the other hand, sale on a day-ahead market has to be decided on without knowing realizations of the random inflow for the future time horizon. This motivates the introduction of a probabilistic constraint as shown above. Figure 3 (left) illustrates the (idealized) sinoidal price signal along with the optimal turbining profiles of the 6 reservoirs. It can be seen that these profiles try to follow the price signal as much as possible in order to maximize the profit. On the other hand, they have to respect the imposed constraints, in particular those for the filling level of the reservoir. The chosen probability level was $p = 0.98$. In order to make an á posteriori check of the robustness of the solution obtained, 100 inflow scenarios were generated according to the given multivariate distribution of the inflow processes. The filling levels in the reservoir resulting upon applying the computed optimal turbining profiles are

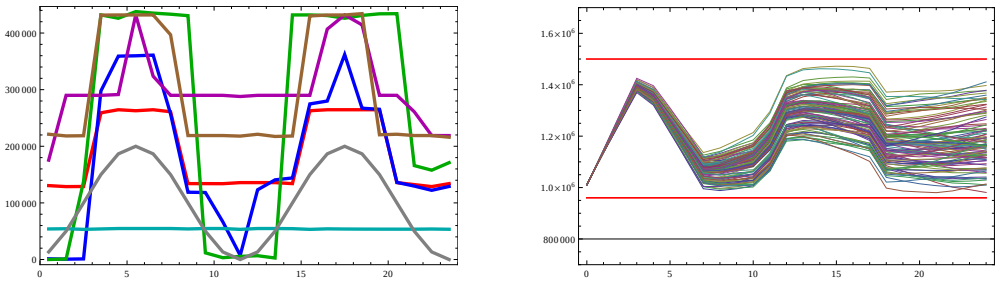


Figure 3: Illustration of the solution to a probabilistically constrained program in power managment.

plotted in Figure 3 (right). It can be seen that all of the filling level100 scenarios stay within the prescribed limits throughout the whole time horizon.

Other applications to power managment were dealing with the choice of an optimal electricity portfolio in production planning under uncertain demand and failure rates [2] and cost-minimal capacity expansion in an electricity network with uncertain demands [28].

3.5 Equilibrium problems in power spot markets

equilibrium problem with equilibrium constraints

In the model of Section 3.1 the viewpoint of a price-taking retailer was adopted. On the level of price-making companies it makes sense to model prices as outcomes of market equilibrium processes driven by decisions of competing power retailers or producers. Mathematically, this leads to so-called *Equilibrium Problems with Equilibrium Constraints* (EPECs):

$$\min_{x_i \in X_i, z \in C} \{ f_i(x_i, z) \,|\, 0 \in F(x_{-i}, x_i, z) + N_C(z) \} \quad (i = 1, \ldots, n), \tag{10}$$

Here, z is a vector of state variables (power generation by each producer, power transmission through the network), x is a decision vector (coefficients for quadratic bidding functions of each producer) and the f_i are the objectives (negative profit functions) of the producers. Note that (10) is a coupled system of optimization problems, where each producer tries to find an optimal decision x_i given the competitors decisions x_{-i}. In other words, (10) is an equilibrium problem. However, in contrast with conventional Nash equilibria, the constraints of competitors are not defined by simple convex sets but by solutions of a generalized equation. The latter models the so-called ISO-problem, in which an independent system operator (ISO) finds cost-minimal generation and transmission in the network, given the consumers demands at the nodes and given the bidding functions of producers. Stationary points for solutions to EPECs can be characterized by tools from nonsmooth and variational analysis. In [29], we provided fully explicit (in terms of the initial data) stationarity conditions for (10) by applying Mordukhovich generalized calculus. In [30], such stationarity conditions were applied to stochastic EPECs with random demands in the nodes.

Automatic Differentiation

4 Derivative based optimization

4.1 Motivation

In contrast to the situation in linear optimization, nonlinear optimization is still comparatively difficult to use, especially in an industrial setting. Well known packages like IPOPT and SNOPT have a large number of options and parameters that are not easy to select and adjust, even for someone who understands the basic mathematics of nonlinear optimization. A particularly critical point is the evaluation of first and second derivatives, which form the basis of local linear and quadratic models in nonlinear programming.

Over the last two decades there has been a concerted effort to bypass the problem through the development of derivative-free algorithms. In fact Nelder–Mead and other derivative-free algorithms dating from the middle of the last century are still rumored to be widely used, despite the danger of them getting stuck on a slope far from a stationary point. A widely accepted fallacy is that algorithms that do not explicitly use derivatives must therefore be good for the solution of *non-smooth* problems with little or no differentiability properties. In fact, all nontrivial convergence results for derivative-free algorithms have been proven under the assumption that the objectives and constraints are sufficiently smooth to be approximated by higher order interpolation [5]. In theory and practice derivative free solvers converge at best at a slow linear rate.

During the Matheon period we have attacked various problems associated with the use of derivatives in the context of optimization. By the turn of the millennium automatic differentiation tools based on operator overloading like for example ADOL-C [17] as well as source transformation tools like Tapenade [22] had reached a considerable level of maturity and were widely applied. With the notable exception of TAF/TAC developed by Fast-Opt in Hamburg all tools are public domains and the support is rather academic. The tool ADOL-C originally written by Andreas Griewank during a two week visit to ZIB in 1989 is now part of the Debian distribution and maintained in the group of Prof. Andrea Walther at the University of Paderborn.

As long as further AD tool development appeared to be mostly a matter of good software design we concentrated on the judicious use of derivatives in simulation and optimization codes. Not only the approximation of Jacobians or Hessians by divided differences, but also their evaluation by algorithmic differentiation as well as their subsequent factorization may take up the bulk of the run-time in an optimization calculation. In some large applications like aerodynamic design optimization evaluating full derivative matrices is simply out of the question. Therefore we have pursued several approaches to develop algorithms that are based on derivative vectors alone, which have provably the same complexity as the function itself.

The following specific goals were pursued by our research group during the Matheon period.

1. A derivative-vector based NLOP solver.

2. A derivative-vector based equations and least squares solver.

3. An unconstrained optimizer based on cubic overestimation.

Gauss Transposed Broyden Nonlinear Least Square

4. Non-smooth optimization via piecewise linearization.

There was also a very significant effort on one-shot optimization in aerodynamics within the DFG priority program 1259, unfortunately it fell outside the Matheon applications areas. Furthermore, there was also a collaboration with project D7 of Caren Tischendorf to analyze and utilize the structure of differential algebraic equations on the basis of their computational graph.

4.2 Transposed updating

The efforts 1) and 2) were based on the secant updating technique described in the following section. The two predominant classes of NLOP solvers, SQP and Interior Point Methods are both based on the evaluation of constraint Jacobians and Lagrangian Hessians with the latter usually being approximated by secant updates in SQP methods. Recent developments of the two methods has been benefited greatly from significant advance in sparse matrix methodology and packages. Rather than exploiting sparsity explicitly our approach was to apply low-rank updating not only to approximate the symmetric Hessian of the Lagrangian but also the rectangular Jacobian of the active constraints. The classical updates for non symmetric derivative matrices, namely the good and bad Broyden formulas [15] suffer from various short comings and have never been nearly as successful as the symmetric counterpart BFGS and its low rank variants.

For a differentiable vector function $F : \mathbb{R}^n \to \mathbb{R}^m$ an approximation $B_+ \in \mathbb{R}^{m \times n}$ may be required to satisfy direct and adjoint secant and tangent conditions of the following form

$$B_+ s = y \equiv F(x+s) - F(x), \quad B_+ s = y \equiv F'(x)s, \quad \sigma^\top B_+ = \sigma^\top F'(x) .$$

Here $s \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m$ are primal and dual steps, which arise naturally within many iterative algorithms. Using the reverse mode of algorithmic differentiation [16] one can evaluate the transposed Jacobian vector product $F'(x)^\top \sigma \in \mathbb{R}^m$ at about the same cost as the direct product $F'(x)^\top \sigma$.

Given a previous approximate Jacobian $B \in \mathbb{R}^{m \times n}$ one can apply the so-called transposed Broyden update

$$B_+ = B + \frac{\sigma\sigma^\top}{\sigma^\top\sigma}\left[F'(x) - B\right] = \left[I - \frac{\sigma\sigma^\top}{\sigma^\top\sigma}\right] B + \frac{\sigma\sigma^\top}{\sigma^\top\sigma} F'(x)$$

to satisfy not only a given transposed secant condition, but also the direct secant condition if one sets $\sigma = r \equiv y - B\,s$. The transposed Broyden formula has many attractive features, in particular it satisfies both bounded deterioration on nonlinear functions and heredity in the affine case.

4.3 Implementations

Nonlinear equations and least squares

Nonlinear optimization
Cubic Overestimation

It was shown in [18] that a nonlinear equations solver based on the transposed Broyden update always achieves the maximal super-linear convergence order $1 + \log(n)/n$ that is achievable by any method based on single rank updating per iterations. On affine problems the method reduces to GMRES and thus achieves finite termination in maximally n steps.

A quasi-Gauss–Newton method based on the transposed formula can be shown to achieve asymptotically the same Q-linear convergence rate as Gauss–Newton. More specifically, consider a stationary point x_* of the squared residual $f(x) \equiv \frac{1}{2}\|F(x)\|^2$ where the normal Hessian $W^2 \equiv [F'(x_*)^\top F'(x_*)] \in \mathbb{R}^{n\times n}$ has full rank. Then the Gauss Transposed Broyden method generates from $x_0 \approx x_*$ a sequence of iterates x_k such that

$$\limsup_{k\to\infty} \frac{\|W(x_{k+1} - x_*)\|_2}{\|W(x_k - x_*)\|_2} \;\leq\; \kappa \equiv \left\| W^{-1} \left[\sum_{i=1}^{m} F_i \nabla^2 F_i(x_*) \right] W^{-1} \right\|_2$$

provided the $\kappa \geq 0$ on the right is less than 1. This generic curvature bound also necessary for the local convergence of Gauss–Newton and implies strict minimality of f at x_*. A similar method with weaker theoretical properties has been applied extensively to geophysical data assimilation problems by Haber [21] with whom we have collaborated. The method was original developed at the HU by Claudia Kratzenstein, who works now on data assimilation problems in oceanography and climatology. A limited memory variant is expected to yield significant performance gains on these very important applications.

Constrained and unconstrained optimization

Within the NLOP solver LRAMBO the transposed updates were used to approximate the Jacobian of the active constraints. In combination with BFGS updating of the Lagrangian Hessian this yielded a null-space implementation, whose linear algebra effort grows only quadratically in the dimensions. Classical SQP methods have a cubic effort in the dimensions. For large scale applications we also developed a limited memory option and an iterative internal solver. The code has been publicly available on the NEOS server since Summer. The method was found to be competitive with standard solvers like SNOPT and IPOPT. However, on the Cuter test set and other collections of primarily academic problems, the avoidance of derivative matrix evaluations did not pay off as much as hoped since there complete Jacobians are never more than 20 times as expensive [4] to evaluate. Moreover, they can usually efficiently factorized due to their regular sparsity structures.

For unconstrained optimizations we developed a code called COUP, which is based on the cubic overestimation idea, originally proposed by Andreas Griewank in 1981. It has recently gained acceptance as an alternative to trust region stabilizations, especially through the work of Gould, Cartis, Gould et al. Our algorithmic development is specifically geared towards the scenarios where second derivatives need to be avoided and reduces the linear algebra effort to $O(n\log(n))^2$ by using fast updates of symmetric eigenvalue decompositions.

4.4 Application and numerical results

Sigmoidal Regression

The first application was a highly non-linear regression problem coming from a cooperation with a German energy provider who was interested in a simple model for the daily consumption of gas based on empirical data that were recorded over the last years to predict future developments. Therefore, a sigmoidal model was used to link the daily gas consumption rate with the temperature of the previous days at one exit point of the gas network. The model itself was given by

$$f(\alpha,\beta,\gamma,\delta,\hat{t}_i(w),X) = E(X)\left(\frac{\alpha}{1+\left(\frac{\beta}{\hat{t}_i(\omega)-40}\right)^{\gamma}}+\delta\right),$$

depending on the expectation $E(X)$ of the N considered gas-flow consumption measurements $X = (X_1,\dots X_N) \in \mathbb{R}^N$, some parameters $(\alpha,\beta,\gamma,\delta) \in \mathbb{R}^4$ and a weighted temperature average $\hat{t}_i(\omega) = \omega_1 t_i + \omega_2 t_{i-1} + \omega_3 t_{i-2} + \omega_4 t_{i-3}$ over the last 4 days of the measured temperatures $t = (t_{-3}, t_{-2}, \dots, t_N) \in \mathbb{R}^{N+3}$ with weighting factors $\omega = (\omega_1,\omega_2,\omega_3,\omega_4) \in \mathbb{R}^4$. The resulting non-linear least square problem

$$\min_{(\alpha,\beta,\gamma,\delta,\omega_1,\omega_2,\omega_3,\omega_4)} \sum_{i=1}^{N} (f(\alpha,\beta,\gamma,\delta,t_i,X) - X_i)^2$$

and several extensions of it were successfully solved by various of our methods (compare Figure 4), and represented a further qualitative improvement to the results mentioned in [35]. We considered above minimization problem including the additional convex-combination constraints $\sum_{j=1}^{4} \omega_j = 1$ and $0 \le \omega_i \le 1$.

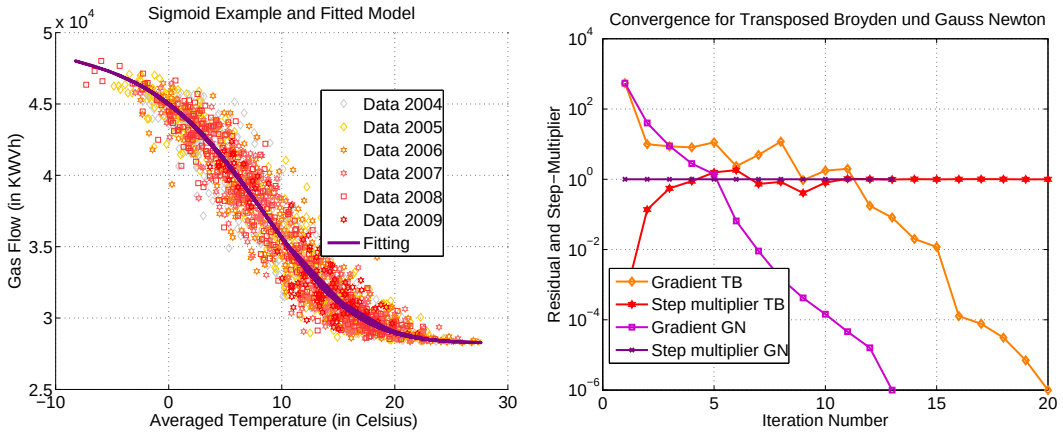


Figure 4: Recorded gas consumptions X_i w.r.t. to the temperature t_i at one exit point and the fitting of the sigmoid model (left); Convergence history for transposed Broyden TN and Gauss Newton GN (right).

Simulated Moving Bed Chromatographical Process

Simulated Moving Bed Processes

In the second application we considered the optimization of a Simulated Moving Bed (*SMB*) process for continuous multi-column chromatography. The problem was used to verify the robustness and performance of our non-linear optimization solver LRAMBO since the periodic adsorption process based on fluid-solid interactions, never reaches steady state, but a cyclic steady state, which leads to dense Jacobians, whose computation dominates the overall cost of the optimization strategy. Therefore, we used a simplified model [42] for SMB with a nonlinear adsorption isotherm consisting of six chromatographic columns, packed with solid adsorbent and arranged in four zones to determine a high purity separation of two components, which was solved by backward Euler method.

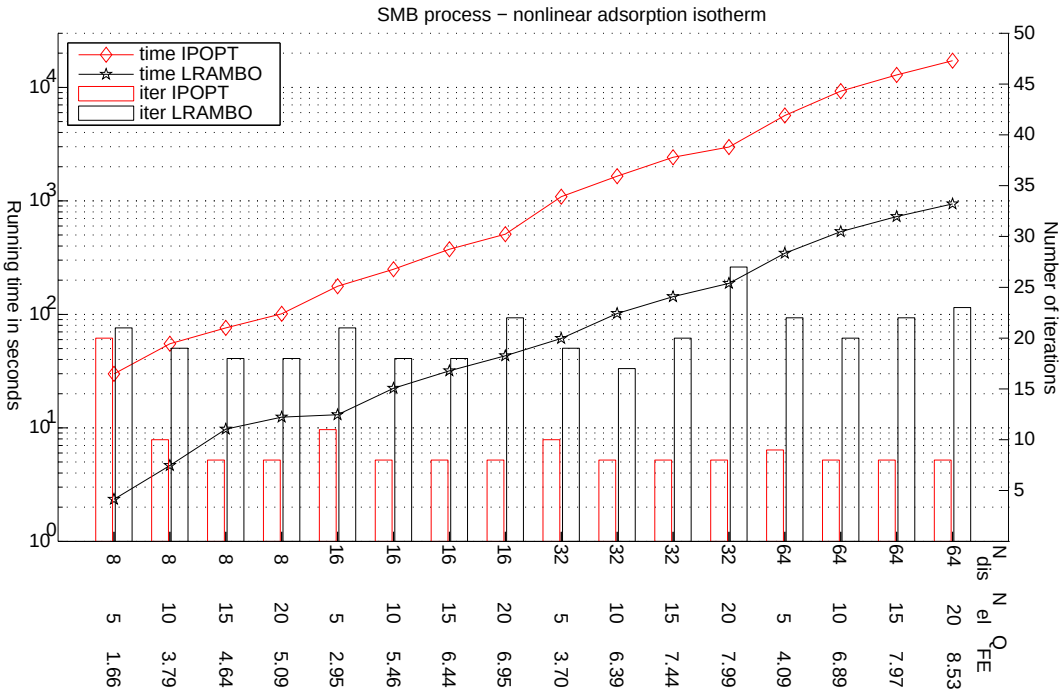


Figure 5: Comparison results for LRAMBO and IPOPT applied to nonlinear SMB.

As decision variables we choose the extract, raffinate, desorbent and feed streams which were limited by lower and upper box-constraints. Furthermore, we added further inequality constraints besides the cyclic steady state condition to the guarantee a purity over 95 percent of the extract and raffinate. As objective functional we maximized the time-averaged throughput in terms of the feed stream.

The optimization was done for a different number of time steps N_{el} and number of compartments N_{dis} using IPOPT and LRAMBO. In all cases both optimizer found an identical optimal function value Q_{FE} for the same initialization but in terms of computation time we were able to outperform IPOPT as can be concluded from 5.

References

[1] W. Van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research*, 71:535–549, 2010.

[2] W. Van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On joint probabilistic constraints with Gaussian coefficient matrix. *Operations Research Letters*, 39:99–102, 2011.

[3] J.T. Betts. *Practical methods for optimal control using nonlinear programming*, volume 3 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[4] T. Bosse and A. Griewank. The relative cost of function and derivative evaluations in the cuter test set. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, editors, *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, pages 233–240. Springer Berlin Heidelberg, 2012.

[5] A.R. Conn, K. Scheinberg, and L. Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming*, 111(1-2):141–172, 2008.

[6] J. Dick and F. Pillichshammer. *Digital Nets and Sequences – Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, 2010.

[7] M. Diehl, H.G. Bock, H. Diedam, and P.B. Wieber. *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*, pages 65–94. Springer, 2005.

[8] A. Eichhorn, H. Heitsch, and W. Römisch. Stochastic optimization of electricity portfolios: Scenario tree modeling and risk management. In S. Rebennack, P. M. Pardalos, M. V. F. Rereira, and N. Iliadis, editors, *Handbook of Power Systems, Vol. II*, pages 405–432. Springer, 2010.

[9] A. Eichhorn and W. Römisch. Polyhedral risk measures in stochastic programming. *SIAM Journal on Optimization*, 16:69–95, 2005.

[10] A. Eichhorn, W. Römisch, and I. Wegner. Mean-risk optimization of electricity portfolios using multiperiod polyhedral risk measures. In *IEEE St. Petersburg Power Tech Proceedings*, 2005.

[11] H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. Walter de Gruyter, 2002 (Third Edition 2011).

[12] M. Gerdts, R. Henrion, D. Hömberg, and C. Landry. Path planning and collision avoidance for robots. *Numerical Algebra, Control and Optimization*, 2(3):437 – 463, 2012.

[13] E.G. Gilbert and D.W. Johnson. Distance functions and their application to robot path planning in the presence of obstacle. *IEEE Journal of Robotics and Automation*, RA-1:21–30, 1985.

[14] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.

[15] A. Griewank. Broyden Updating, the Good and the Bad! In M. Grötschel, editor, *Optimization Stories*, volume Extra Volume: Optimization Stories, pages 301–315. Documenta Mathematica, Bielefeld, 2012.

[16] A. Griewank. Who Invented the Reverse Mode of Differentiation? In M. Grötschel, editor, *Optimization Stories*, volume Extra Volume: Optimization Stories, pages 389–400. Documenta Mathematica, Bielefeld, 2012.

[17] A. Griewank, D. Juedes, and J. Utke. ADOL-C: A package for automatic differentiation of algorithms written in C/C++. *TOMS*, 22:131–167, 1996.

[18] A. Griewank, S. Schlenkrich, and A. Walther. A quasi-Newton method with optimal R-order without independence assumption. *OMS*, 23(2):215–225, 2008.

[19] N. Gröwe-Kuska, K.C. Kiwiel, M.P. Nowak, W. Römisch, and I. Wegner. Power management in a hydro-thermal system under uncertainty by lagrangian relaxation. In Claude Greengard and Andrzej Ruszczyński, editors, *Decision Making under Uncertainty – Energy and Power*, pages 39–70. Springer, 2002.

[20] V. Guigues and W. Römisch. Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM Journal on Optimization*, 22:286–312, 2012.

[21] E. Haber, U.M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16(5):1263–1280, October 2000.

[22] L. Hascoët and V. Pascual. The Tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3):20:1–20:43, 2013.

[23] H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118:371–406, 2009.

[24] H. Heitsch and W. Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 6:117–133, 2009.

[25] H. Heitsch, W. Römisch, and C. Strugarek. Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17:511–525, 2006.

[26] R. Henrion. Structural properties of linear probabilistic constraints. *Optimization*, 56:425–440, 2007.

[27] R. Henrion. Gradient estimates for Gaussian distribution functions: Application to probabilistically constrained optimization problems. *Numerical Algebra, Control and Optimization*, 2:655–668, 2012.

[28] R. Henrion and A. Möller. A gradient formula for linear chance constraints under Gaussian distribution. *Mathematics of Operations Research*, 37:475–488, 2012.

[29] R. Henrion, J. Outrata, and T. Surowiec. Analysis of M-stationary points to an EPEC modeling oligopolistic competition in an electricity spot market. *ESAIM: Control, Optimization and Calculus of Variations*, 18:295–317, 2012.

[30] R. Henrion and W. Römisch. On M-stationary points for a stochastic equilibrium problem under equilibrium constraints in electricity spot market modeling. *Applications of Mathematics*, 52:473–494, 2007.

[31] R. Henrion and W. Römisch. Lipschitz and differentiability properties of quasi-concave and singular normal distribution functions. *Annals of Operations Research*, 177:115–125, 2010.

[32] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.

[33] C. Landry, M. Gerdts, R. Henrion, and D. Hömberg. Path-planning with collision avoidance in automotive industry. In *System Modeling and Optimization*, 25th IFIP TC 7 Conference, Berlin, Germany, September 12-16, 2011, Revised Selected Papers IFIP AICT 391, Springer, Heidelberg; Approx. IX, 575 pp, 2013.

[34] C. Landry, R. Henrion, D. Hömberg, M. Skutella, and W. Welz. Task assignment, sequencing and path-planning in robotic welding cells. In *Methods and Models in Automation and Robotics (MMAR), 2013 18th International Conference on*, 2013. accepted.

[35] H. Leövey, W. Römisch, A. Steinkamp, and I. Wegner-Specht. Modellierung der Gasabnahme als Funktion der Temperatur: Optimierung der Temperaturgewichte. *gwf Gas/Erdgas*, 11:778–785, 2011.

[36] G.C. Pflug and W. Römisch. *Modeling, Measuring and Managing Risk*. World Scientific, 2007.

[37] W. Römisch. Scenario generation. In James J. Cochran, editor, *Wiley Encyclopedia of Operations Research and Management Science*, pages 515–558. Wiley, 2010.

[38] K. Schittkowski. On the convergence of a sequential quadratic programming method with an augmented lagrangian line search function. *Mathematische Operationsforschung und Statistik*, 14:197–216, 1983.

[39] S. Sen. Algorithms for stochastic mixed-integer programming models. In K. Aardal, G. L. Nemhauser, and R. Weismantel, editors, *Handbook of Discrete Optimization*, pages 515–558. Elsevier, 2005.

[40] S. Sen, L. Yu, and T. Genc. A stochastic programming approach to power portfolio optimization. *Operations Research*, 54:55–72, 2006.

[41] M. Skutella and W. Welz. Route planning for robot systems. In B. Hu, K. Morasch, S. Pickl, and M. Siegle, editors, *Operations Research Proceedings 2010*, pages 307–312. Springer, 2011.

[42] A. Walther and L.T. Biegler. Numerical experiments with an inexact jacobian trust-region algorithm. *Comp. Opt. and Appl*, 48(2):255–271, 2011.

Index