Nonsmooth Optimization of Systems with Varying Structure

Mehmet Yunt, * **Paul I. Barton** **

* *Dept. of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: myunt@mit.edu)*
** *Dept. of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: pib@mit.edu)*

Abstract: A novel method based on the generalized gradient and nonsmooth optimization techniques called bundle methods is introduced to optimize the performance of a class of dynamic systems whose governing equations change depending on the values of the parameters, controls and the current state of the system.

Keywords: Nonsmooth Optimization, Generalized Gradient, Bundle Methods, Dynamic Optimization

1. INTRODUCTION

This paper describes a novel method to determine the optimal controls and parameters for a large class of engineering systems with varying structure which have the following characteristics:

(1) The systems evolve according to different ordinary differential equations depending on the values of the states, controls and parameters.
(2) The vector field is unique and continuous.
(3) The values of the continuous states, controls and parameters solely determine the vector field.

Example 1. The liquid level dynamics of a tank with a weir with multiple inlet and outlet flows is

$$\dot{h}(t,\mathbf{p}) = \sum_{i=1}^{n} F_i(t)/A - F_W(h(t,\mathbf{p}),\mathbf{p}) \tag{1}$$

$$F_W(h(t,\mathbf{p}),\mathbf{p}) = \begin{cases} 0 & \text{if } h(t,\mathbf{p}) \le \bar{h} \\ k(h(t,\mathbf{p}) - \bar{h}) & \text{if } h(t,\mathbf{p}) \ge \bar{h} \end{cases}$$

$$\mathbf{p} = (\bar{h}, k, A)^{\mathrm{T}}$$

where h is the liquid level; A is the cross-sectional area of the tank; F_i are volumetric flows; $\bar{h}$ is the weir height; k is an equivalent valve constant for the weir. The governing ordinary differential equations are determined by h and $\bar{h}$. At $h = \bar{h}$, two ordinary differential equations are applicable but the vector field is unique and continuous.

These systems can be expressed in the form,

$$\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{u}(t),\mathbf{p})$$

where $\mathbf{x}$ represents the continuous-valued states, $\mathbf{p}$ is a finite set of continuous-valued parameters, $\mathbf{u}(t)$ are the bounded controls with possible discontinuities at finitely many points in time. $\mathbf{f}$ is continuous on its domain. The domain of $\mathbf{f}$ is partitioned into finitely many sets, D_k, such that if $(\mathbf{x}(t,\mathbf{p}),\mathbf{u}(t),\mathbf{p}) \in D_k$ then $\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{u}(t),\mathbf{p}) = \mathbf{f}_k(\mathbf{x}(t,\mathbf{p}),\mathbf{u}(t),\mathbf{p})$ where $\mathbf{f}_k$ are continuously differentiable with respect to their arguments. As a result of this particular structure $\mathbf{f}$ is piecewise continuously differentiable with respect to its arguments.

In order to formulate a finite dimensional optimization problem, the controls are reformulated as piecewise constant functions of parameters and time with finitely many discontinuities at $t_i,\ i = 1,\ldots,n$ with $t_i < t_{i+1}$, $t_1 = 0$ and $t_n < \infty$. As a result of this reformulation, the dynamics can be represented by a set of equations $\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ if $t \in (t_i, t_{i+1}]$. Note that each $\mathbf{f}^i$ has a partitioned domain with finitely many partitions, $\{D_k^i,\ k = 1,\ldots,n_i\}$, and corresponding continuously differentiable vector fields $\{\mathbf{f}_k^i\}$.

The mathematical program to be solved is

$$\min_{\mathbf{p}} J(\mathbf{p}) = \sum_{i=1}^{n-1} \int_{t_i}^{t_{i+1}} h(\mathbf{x}(t,\mathbf{p}),\mathbf{p})dt + H(\mathbf{x}(t_n,\mathbf{p}),\mathbf{p}) \tag{2}$$

$$\text{s.t} \sum_{i=1}^{n-1} \int_{t_i}^{t_{i+1}} \mathbf{g}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})dt + \mathbf{G}(\mathbf{x}(t_n,\mathbf{p}),\mathbf{p}) \le \mathbf{0}$$

$$\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p}),\ \forall t \in (t_i, t_{i+1}], \tag{3}$$

$$\mathbf{x}(0,\mathbf{p}) = \mathbf{f}^0(\mathbf{p}),\ i = 1,\ldots,n-1$$

where H, h, $\mathbf{G}$, $\mathbf{g}$, $\mathbf{f}^0$ are continuously differentiable functions of their arguments that are used to compute values of the objective, values of the constraints and the initial conditions.

Standard numerical methods of dynamic optimization (Betts, 1998) are not applicable to solve (2) because the $\mathbf{f}^i$ are not continuously differentiable. Sensitivity results (Galán et al., 1999) are also not applicable because given $t \in (t_i, t_{i+1}]$, it is not a priori known which $\mathbf{f}_k^i$ govern the dynamics of the system. Furthermore, at $t \in (t_i, t_{i+1}]$ the governing $\mathbf{f}_k^i$ depends on the value of $\mathbf{p}$. The solution of (2) not only determines an optimal $\mathbf{p}$ but prescribes a sequence of $\mathbf{f}_k^i$ in time. This implicit selection complicates the solution process.

The *mixed-integer* (Avraam et al., 1998; Bemporad and Morari, 1999) and the *differential variational inequalities, DVI* (Schumacher, 2004; Pang and Stewart, 2008; Raghunathan et al., 2004) approaches make this selection explicit using transcription. Given $[t_i, t_{i+1}]$, a mesh of time points $\{\tau_j : \ j = 1, 2, \ldots n_i, \ \tau_1 = t_i \ \tau_{n_i} = t_{i+1}\}$ is determined a priori. For each τ_j, a variable, $\mathbf{x}_j$, representing the continuous state is created. In order to make the selection of $\mathbf{f}_k^i$ explicit, (3) needs to be replaced with suitable algebraic relationships and additional constraints. The governing dynamics need to be determined in the intervals $[\tau_j, \tau_{j+1}]$. Both approaches replace (3) with a discretization such as the forward Euler method. Both methods introduce additional variables, μ_j^k, at each τ_j for each $\mathbf{f}_k^i$ and approximate the dynamics between time points using the function; $\sum_k \mu_j^k \mathbf{f}_k^i(\mathbf{x}_j, \mathbf{p})$. For example, if the forward Euler discretization is used, an algebraic relationship replacing (3) is $\mathbf{x}_{j+1} - \mathbf{x}_j = (\tau_{j+1} - \tau_j) \sum_{k=1}^{n_i} \mu_j^k \mathbf{f}_k^i(\mathbf{x}_j, \mathbf{p}), \ j = 1, \ldots, n_i - 1$. Both approaches replace the integral terms in (2) by appropriate quadratures.

The approaches differ in the way the values of μ_j^k are determined. In the mixed-integer approach, μ_j^k are binary variables. Additional constraints enforce that a single μ_j^k is non-zero at a given τ_j and that this is consistent with parameter, $\mathbf{p}$, and state values, $\mathbf{x}_j$ at τ_j. The final formulation is a mixed-integer nonlinear program. In the DVI approach, μ_j^k are part of the solution of an embedded mathematical program of the states and parameters at each τ_j. These programs are replaced with their equivalent KKT conditions.

Both approaches result in large optimization problems as a result of transcription. Only linear and quadratic formulations of the mixed-integer approach can be solved practically. This restricts the underlying ODEs to be linear and the sets D_k to be polyhedral. The DVI approach requires that the embedded programs are convex to guarantee that the KKT conditions are sufficient to determine an optimal solution. Complementarity constraints in the KKT conditions necessitate special solvers because complementarity conditions violate constraint qualifications which are necessary for ordinary nonlinear programming, NLP, algorithms to work. The required special solvers are not as efficient as usual NLP solvers.

This paper describes a method where the selection of dynamics is not handled explicitly. The method is based on *single-shooting*. In single shooting, the dynamics in (3) are solved by an initial value solver, IVP. In the differentiable case, the IVP solver is also used to solve an auxiliary set of equations to obtain parametric sensitivities. These sensitivities are used to calculate gradient information for numerical optimization methods. There are no additional variables or constraints. The resultant optimization problems do not grow with the number of time points in the mesh nor the size of the possible set of ODEs. In addition, the convexity constraints on the dynamics as in the mixed-integer and DVI approaches can be relaxed.

In order to use single-shooting on (2), which is a nonsmooth program, derivative-like information needs to be obtained. Nonsmoothness also prevents the application of standard nonlinear programming solvers. In order to handle these complications, *Clarke's generalized Jacobian* (Clarke, 1990) is employed in conjunction with a class of nonsmooth optimization methods called *bundle methods* (Kiwiel, 1985; Mäkelä, 2001) to solve (2).

In the remainder, the necessary mathematical background is summarized. The method is then described. The paper ends with an illustrative example and directions of further research.

2. MATHEMATICAL BACKGROUND

$\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$ is *locally Lipschitz continuous* at $\mathbf{x}$ if there exists a neighborhood, $N_\epsilon(\mathbf{x})$ and a finite positive constant, K, such that $\|\mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{z})\| \leq K\|\mathbf{y} - \mathbf{z}\| \ \forall \mathbf{z}, \mathbf{y} \in N_\epsilon(\mathbf{x})$. Rademacher's theorem states that locally Lipschitz continuous functions are almost everywhere differentiable on their domain (Rockafellar and Wets, 1998). The locally Lipschitz continuous property is preserved under addition and composition of functions.

Let $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$ be locally Lipschitz continuous at $\mathbf{x}$, then the *generalized Jacobian* (Clarke, 1990) at $\mathbf{x}$ is

$$\partial \mathbf{F}(\mathbf{x}) = \text{co}\{ \lim_{\mathbf{x}_i \to \mathbf{x}} J\mathbf{F}(\mathbf{x}_i) : \ \mathbf{x}_i \notin S \cup T\} \quad (4)$$

where co is the convex hull, $J\mathbf{F}$ denotes the Jacobian of $\mathbf{F}$ where it exists; S is the set of measure zero nondifferentiable points and T is an arbitrary set of measure zero. In words, the generalized Jacobian at $\mathbf{x}$ is the convex hull of all the limits of convergent Jacobian sequences with the Jacobians evaluated at points converging to $\mathbf{x}$. In finite dimensional Euclidean spaces, the *generalized gradient* is the generalized Jacobian when $m = 1$ and the elements of the generalized Jacobian are transposed.

Example 2. $\dot{h}(t, \mathbf{p})$ in (1) is locally Lipschitz continuous with respect to h and the generalized gradient is:

$$\partial_{\bar{h}} \dot{h}(t, \mathbf{p}) = \begin{cases} 0 & \text{if } h(t, \mathbf{p}) < \bar{h} \\ \text{co}[0, k] & \text{if } h(t, \mathbf{p}) = \bar{h} \\ k & \text{if } h(t, \mathbf{p}) > \bar{h}. \end{cases}$$

Chain rules can be derived for generalized gradients and Jacobians. Implicit function theorems can be formulated. Necessary conditions of optimality for mathematical programs with locally Lipschitz continuous functions can be defined in terms of generalized gradients. For numerical methods, if the generalized gradient at a point is known, a direction of descent can be obtained by using the element of minimum norm.

In general, it is not possible to calculate all the elements of the generalized gradient at a point to determine directions of descent. *Bundle Methods* (Kiwiel, 1985; Mäkelä, 2001) use an approximation to the generalized gradient to solve

$$\min_{\mathbf{z} \in Z} f(\mathbf{z}) \text{ s.t: } g_m(\mathbf{z}) \leq 0, \ m = 1, \ldots, M$$

where f and g_m are locally Lipschitz continuous functions. Bundle methods require that only an element of $\partial f(\mathbf{z})$ and of each $\partial g_m(\mathbf{z})$ are available. The generalized gradient at an iterate is approximated by the convex

hull of a set of generalized gradients of nearby points called the *bundle*. The element of minimum norm of the approximation is used as the descent direction. Using a specialized line search procedure, either the next iterate is determined or another element is added to the bundle to change the direction of descent.

Bundle methods converge to stationary points satisfying KKT type conditions under a *semismoothness* (Mifflin, 1977; Qi, 1993) assumption on f and g_m. Semismoothness guarantees that the iterative line search algorithm in bundle methods terminates after finite number of iterations. Similar to local Lipschitz continuity, semismoothness is conserved under addition, multiplication and composition. Piecewise continuously differentiable functions, finite convex functions and continuously differentiable functions are all examples of functions that are semismooth and locally Lipschitz continuous.

3. METHOD DESCRIPTION

In this section, the theoretical discussion focuses on the dynamic systems,

$$\begin{aligned}&\dot{\mathbf{x}}(t,\mathbf{p}) = \mathbf{f}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p}),\ \forall t \in (t_i, t_{i+1}] \\ &\mathbf{x}(t_1,\mathbf{p}) = \mathbf{f}^0(\mathbf{p}),\ \mathbf{p} \in P,\ i = 1 \dots n-1\end{aligned} \tag{5}$$

where $\mathbf{f}^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ are piecewise continuously differentiable with respect to their arguments, $\mathbf{f}^0 : \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ is continuously differentiable and P is a compact set with non-empty interior. In order to develop a numerical method, the following assumptions are made:

Assumption 1. Equation (5) has a solution on $[t_i, t_{i+1}]$, $i = 1, \dots, n-1$ for each $\mathbf{p} \in P$.

Assumption 2. The domain of each $\mathbf{f}^i$ is partitioned into a finite set of subdomains with nonempty interior, $\{D_k^i,\ k = 1, \dots, n_i\}$ and $D_k^i = \{(\mathbf{v},\mathbf{p}) : d_{k,j}^i(\mathbf{v},\mathbf{p}) \leq 0,\ j = 1, \dots, n_{i,k}\}$ where $d_{k,j}^i$ are continuously differentiable. In other words, the partitions have boundaries that can be expressed using continuously differentiable functions and $d_{k,j}^i(\mathbf{v},\mathbf{p}) = 0$ represent continuously differentiable manifolds of dimension $n_x \times n_p - 1$. There exists a corresponding set of $\{\mathbf{f}_k^i\}$ such that $\mathbf{f}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) = \mathbf{f}_k^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ if $(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) \in D_k^i$ and $\mathbf{f}_k^i$ are continuously differentiable with respect to their arguments.

$\mathbf{x}(t,\mathbf{p})$ is a locally Lipschitz continuous (Coddington and Levinson, 1955) and semismooth function of $\mathbf{p}$ (Pang and Stewart, 2009) as a result of continuity and piecewise continuous differentiability of $\mathbf{f}^i$. The constraints and objective of (2) are composite functions of semismooth and locally Lipschitz functions and $\mathbf{x}(t,\mathbf{p})$. As a result, the constraints and objective of (2) are locally Lipschitz and semismooth functions.

In order to calculate the necessary generalized gradient information for optimization, an element of the generalized gradient of the states with respect to $\mathbf{p}$, $\partial_{\mathbf{p}}\mathbf{x}(t,\mathbf{p})$, is required. The next theorem provides a sufficient condition to detect points where an element can be calculated. It can be deduced from Theorem 7.4.1 in Clarke (1990) using appropriate chain rules and the definition of the Jacobian.

Theorem 1. Let $\mathbf{x}(t,\bar{\mathbf{p}})$ be a solution of (5). If the set-valued mapping, $\partial_{(\mathbf{x},\mathbf{p})}\mathbf{f}_{\mathbf{p}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}})$ is a singleton for almost all $t \in (t_1, t_n]$, then for each $i = 1, \dots, n-1$, there exist unique solutions to the matrix differential inclusions:

$$\begin{aligned}&\dot{Y}_{\mathbf{p}}(t) \in \partial\mathbf{f}_{\mathbf{x}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}})Y_{\mathbf{p}}(t) + \partial\mathbf{f}_{\mathbf{p}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}}), \\ &\qquad\qquad \forall t \in (t_i, t_{i+1}] \\ &\dot{Y}_{\mathbf{x}}(t) \in \partial\mathbf{f}_{\mathbf{x}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}})Y_{\mathbf{x}}(t), \forall t \in (t_i, t_{i+1}] \\ &Y_{\mathbf{p}}(t_1) = 0,\ Y_{\mathbf{x}}(t_1) = I.\end{aligned}$$

The following relations hold for all $t \in (t_i, t_{i+1}]$ except on set of measure zero,

$$\dot{Y}_{\mathbf{p}}(t) = J\mathbf{f}_{\mathbf{x}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}})Y_{\mathbf{p}}(t) + J\mathbf{f}_{\mathbf{p}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}}) \tag{6}$$

$$\dot{Y}_{\mathbf{x}}(t) = J\mathbf{f}_{\mathbf{x}}^i(\mathbf{x}(t,\bar{\mathbf{p}}),\bar{\mathbf{p}})Y_{\mathbf{x}}(t) \tag{7}$$

where $J\mathbf{f}_{\mathbf{x}}^i$ and $J\mathbf{f}_{\mathbf{p}}^i$ are the partial derivatives of $\mathbf{f}^i$ with respect to $\mathbf{x}$ and $\mathbf{p}$ respectively.

Finally, $\partial_{\mathbf{p}}\mathbf{x}(t_n,\bar{\mathbf{p}}) = Y_{\mathbf{x}}(t_n)J\mathbf{f}_0(\bar{\mathbf{p}}) + Y_{\mathbf{p}}(t_n)$.

Definition 1. A trajectory, $\mathbf{x}(t,\mathbf{p})$, is called *singleton* if $\partial\mathbf{f}_{(\mathbf{x},\mathbf{p})}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ is a singleton for almost all $t \in [t_i, t_{i+1}]$. Otherwise, it is called *non-singleton*.

Note that due to assumption (2) and the piecewise continuous differentiable nature of $\mathbf{f}^i$, the points where $\partial\mathbf{f}_{(\mathbf{x},\mathbf{p})}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ may not be a singleton are where $d_{k,j}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) = 0$ holds for some k and j. Trajectories that are not singleton have arcs that lie on the surfaces defined by $d_{k,j}^i(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) = 0$.

The next theorem is a result on the occurrence of non-singleton trajectories for autonomous systems with piecewise continuously differentiable vector fields.

Theorem 2. Consider the dynamic system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)),\ t \in (0, t_f],\ \mathbf{x}(0) = \mathbf{x}_0,\ \mathbf{x}_0 \in X_0 \subset D \tag{8}$$

where X_0 is an open set in $\mathbb{R}^{n_x}$, D is the bounded domain where all solutions with initial conditions in X_0 remain for $t \in (0, t_f]$. Let $\mathbf{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ be piecewise continuously differentiable on D. Let assumptions (2) hold where all $d_{k,j}^i$ are continuously differentiable functions of $\mathbf{x}$. Then the set of initial conditions producing non-singleton trajectories is a measure zero subset of X_0.

Proof. Since $\mathbf{f}$ is locally Lipschitz continuous on D, the set of points where it is not differentiable is a set of measure zero in D. In addition, due to piecewise continuous differentiability, the generalized Jacobian is a singleton where $\mathbf{f}$ is differentiable. The only points where the generalized Jacobian is not a singleton are on the boundaries of the subdomains which constitute a set of measure zero in D. The solutions of (8) are unique due to the Lipschitz continuous property of $\mathbf{f}$ on D. Due to the autonomous nature of the dynamics, no two solutions intersect for any $t \in (0, t_f]$. Now consider trajectories that pass through boundary points. Since points on the boundaries are a set of measure zero in D, the set of initial conditions producing these trajectories are a set of measure zero in D. Since the set of initial conditions that produce non-singleton trajectories is a subset of the set of initial conditions that produce trajectories passing through boundary points, the set of initial conditions producing non-singleton trajectories is a set of measure

zero in D. Finally the intersection of X_0 which is open in $\mathbb{R}^{n_x}$ with this set of initial conditions is a set of measure zero.

If the initial conditions of (8) are functions of a set of parameters, then the result of the previous theorem may not hold. It is possible that the functions always map the parameters to initial conditions resulting in non-singleton trajectories. Sufficient conditions to determine functions that map parameters to sets of initial conditions for which theorem (2) holds is under investigation.

In the remainder of the paper, the following statement is assumed to hold.

Assumption 3. Let $\tilde{P}$ the set of parameters such that $\mathbf{x}(t,\mathbf{p})$ with $\mathbf{p}\in\tilde{P}$ is not a singleton trajectory. The set $\tilde{P}$ has measure zero in P.

The equations (5), (6) and (7) have to be solved numerically. The solution of (5) poses no problems and can be accomplished with an ordinary IVP solver because $\mathbf{f}^i$ are Lipschitz continuous vector field on their domains. On the other hand, solutions of (6) and (7) require further discussion. Since $\mathbf{f}^i$ are only piecewise continuously differentiable, $J_{\mathbf{p}}\mathbf{f}^i$ and $J_{\mathbf{x}}\mathbf{f}^i$ are not continuous on the domain of $\mathbf{f}^i$. As a result (6) and (7) are differential equations with discontinuities in time. These discontinuities need to be detected for an efficient and correct solution. By virtue of assumption (2), the points where $\mathbf{f}^i$ are not differentiable are on the boundaries of the partitions and satisfy $d^i_{k,j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})=0$ for some values of k and j. Event detection algorithms (Park and Barton, 1996) are used to determine t^* such that $d^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})=0$. The event detection algorithm tracks the signs of $d^i_{k,j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ for each i and j. A sign change implies that the state trajectory, $\mathbf{x}(t^*,\mathbf{p})$ crossed a boundary of discontinuity. The event detection algorithm finds the earliest time when the boundary crossing occurred. The integration of (6) and (7) are restarted at t^*.

Event detection algorithms are also used to detect trajectories that are non-singleton. Once t^* is detected where $d^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})=0$ for some k and j, $\dot{d}^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})$ is checked. If $\dot{d}^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})=0$ for some k and j, this implies that the state trajectory may not leave the surface defined by $d^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})=0$ resulting in a trajectory that is possibly non-singleton. Integration is continued until $t+\delta t$ where δt is a small quantity. If there exists any k and j such that $d^i_{k,j}(\mathbf{x}(t^*,\mathbf{p}),\mathbf{p})=0$ and $d^i_{k,j}(\mathbf{x}(t^*+\delta t,\mathbf{p}),\mathbf{p})=0$, the trajectory is considered to be not a singleton trajectory.

In case $\bar{\mathbf{p}}$ does not correspond to a singleton trajectory, the definition of the generalized Jacobian (4) can be used to approximate an element of the generalized Jacobian. Random parameter values in an ϵ neighborhood, $N_\epsilon(\bar{\mathbf{p}})$, can be used to find a nearby singleton trajectory and calculate an approximate generalized Jacobian.

The necessary generalized gradient information for the objective and constraint functions of (2) is obtained by applying the chain rules for generalized gradients once an element or an approximation of $\partial_{\mathbf{p}}\mathbf{x}(t,\mathbf{p})$ is calculated. The calculated generalized gradient information is used in conjunction with a bundle method to obtain a stationary point of (2).

4. ILLUSTRATIVE EXAMPLE

In this section, a modified version of the cell cycle specific chemotherapy model introduced in Pannetta and Adam (1995) is used to determine an optimal chemotherapy drug schedule. The dynamics,

$$\alpha = a-m-n$$
$$\dot{P} = \alpha P + bQ - F_A(v_A,P) \tag{9}$$
$$F_A(v_A,P)=\begin{cases}0 & \text{if } v_A-\bar{v}_A\le 0\\ k_A(v_A-\bar{v}_A)P & \text{if } v_A-\bar{v}_A\ge 0\end{cases}$$
$$\dot{Q} = mP - bQ - F_B(v_B,Q) \tag{10}$$
$$F_B(v_B,Q)=\begin{cases}0 & \text{if } v_B-\bar{v}_B\le 0\\ k_B(v_B-\bar{v}_B)Q & \text{if } v_B-\bar{v}_B\ge 0\end{cases}$$
$$\dot{Y} = \sigma Y(1-Y/K) - k_A v_A Y - k_B v_B Y \tag{11}$$
$$\dot{v}_A = u_A - \gamma_A v_A \tag{12}$$
$$\dot{v}_B = u_B - \gamma_B v_B \tag{13}$$

represents the behavior of tumor cells and healthy cells in human tissue under chemotherapy. The tissue comprises healthy cells, Y, proliferating tumor cells, P, and quiescent tumor cells, Q. Chemotherapy comprises two drugs; A and B. u_A and u_B are the chemotherapy drug schedules. v_A and v_B are the exponentially decaying drug concentrations in the tissue. Tumor cells develop resistance to drugs. As a result, drugs are effective against the tumor cells only if their concentrations in the tissue are above $\bar{v}_A$ and $\bar{v}_B$. A fraction, n, of proliferating cells die of natural causes and and a fraction, m, of proliferating cells become quiescent cells. The increase in proliferating cell population by cell division is represented as another fraction, a, of the proliferating cell population. In addition, a fraction, b of quiescent cells become proliferating cells. The tumor cell dynamics are in (9) and (10).

A logistic equation (11) governs the healthy cell population to ensure that the number of healthy cells does not exceed the carrying capacity, K. Numerical values for the various parameters are displayed in Table 1. Most of the values are obtained from (Dua et al., 2008) where cell cycle specific chemotherapy with a single drug and without drug resistance is considered. Note that $[D]$ is a unit of drug concentration. The program,

$$\min_{\mathbf{u}_A,\mathbf{u}_B} P(t_f,\mathbf{u}_A,\mathbf{u}_B)+Q(t_f,\mathbf{u}_A,\mathbf{u}_B) \tag{14}$$
$$\text{s.t: } Y(t_f,\mathbf{u}_A,\mathbf{u}_B)\ge Y_{\min} \tag{15}$$
$$u_{\min}\le u_{A,j}\le u_{\max},\ j=1,\dots,n_f,$$
$$u_{\min}\le u_{B,j}\le u_{\max},\ j=1,\dots,n_f,$$
$$P(1,\mathbf{u}_A,\mathbf{u}_B)=P_0,\ Q(1,\mathbf{u}_A,\mathbf{u}_B)=Q_0,$$
$$Y(1,\mathbf{u}_A,\mathbf{u}_B)=Y_0,\ v_A(1,\mathbf{u}_A,\mathbf{u}_B)=0,$$
$$v_B(1,\mathbf{u}_A,\mathbf{u}_B)=0$$

where $\mathbf{u}_A=\{u_{A,j}\}$ and $\mathbf{u}_B=\{u_{B,j}\}$ are the set of daily drug doses for an n_f-day treatment, is solved to minimize the tumor cell population without totally destroying the healthy cell population. The numerical values used are in Table 2.

DSL48SE is the IVP solver (Tolsma, 2001; Tolsma and Barton, 2002; Feehery et al., 1997) used to integrate the dynamics and the corresponding auxiliary equations to obtain an element of the generalized Jacobian. The event detection algorithm of DSL48SE (Park and Barton, 1996) is used to detect non-singleton trajectories. The necessary Jacobians for the auxiliary system of equations are obtained using automatic differentiation algorithms implemented in DAEPACK (Tolsma and Barton, 2000). The differential equations are integrated with an absolute tolerance of 1×10^{-7} and a relative tolerance of 1×10^{-9}.

A modified proximal bundle method based on the algorithm in (Lukšan and Vlček, 2001) is used to solve (14). A penalty approach to handle (15) is used because the algorithm in (Lukšan and Vlček, 2001) is handles only linear constraints on the decision variables. The objective of (14) is augmented with (15) to obtain

$$J(\mathbf{u}_A, \mathbf{u}_B) = P(t_f, \mathbf{u}_A, \mathbf{u}_B) + Q(t_f, \mathbf{u}_A, \mathbf{u}_B) + \mu_k \max(Y_{\min} - Y(t_f, \mathbf{u}_A, \mathbf{u}_B), 0)$$

where μ_k is the penalty parameter. The modified program is successively solved three times with increasing penalty parameter to an optimality tolerance of 1×10^{-5}. The solution of the preceding programs are used as the initial guesses for the following programs. For the first program, the drug schedules are assigned random values between 0.0 and 5.0. The penalty parameter values are 5000, 25000 and 125000.

The cell population numbers at the beginning and end of the treatment are in Table 3. The tumor cell population is reduced to one percent of its initial size. The drug schedules are shown in Figure 1 and Figure 2. The preference to use drug B is clearly seen. The effects of the drugs are proportional to the corresponding cell populations. Therefore using drug B results in more effective treatment as the population of quiescent cells is greater than that of proliferating cells. In addition, the ratio of tumor cells killed to the ratio of healthy cells killed per unit drug concentration is larger for drug B.

The drug B schedule has four distinctive phases. The initial four-day treatment reverses the increase in the tumor cell population by using drug B as much as possible. In the next week, the drug B concentration is allowed to decay to a tolerable level for the patient. The treatment until the last three days keeps the drug B concentration at that tolerable level. In the last days of the treatment, the drug dose is increased to kill the maximum number of tumor cells. This spike in the drug concentration shows its effect on the healthy cell population after the treatment is over and does not affect (15) significantly.

5. CONCLUSION

In this document, a novel method to optimize the performance of a class of systems with varying structure has been introduced. The theoretical basis and an initial implementation has been described. An illustrative numerical example has been presented.

The implementation of the optimization method will be streamlined in the future. There are different variants of

Table 1. Parameters of equations (9)-(13)

a	0.500 day^{-1}	$\bar{v}_A$	10.000 [D]
m	0.218 day^{-1}	$\bar{v}_B$	10.000 [D]
n	0.477 day^{-1}	k_A	8.400×10^{-3} day^{-1}[D]$^{-1}$
b	0.100 day^{-1}	k_B	8.400×10^{-3} day^{-1}[D]$^{-1}$
σ	0.100 day^{-1}	K	$10000M$ cells
γ_A	0.100 day^{-1}	γ_B	0.100 day^{-1}

Table 2. Parameters of the mathematical program (14)

t_f	31 days	$Y_{\min}$	$100M$ cells
n_f	30	Y_0	$10000M$ cells
$u_{\max}$	20.00 [D]day^{-1}	Q_0	$8.00 \times 10^5 M$ cells
$u_{\min}$	0.00 [D]day^{-1}	P_0	$2.00 \times 10^5 M$ cells

Table 3. Cell Populations at the beginning and end of treatment

	Beginning of Treatment	End of Treatment
Y	$10000M$ cells	$100M$ cells
Q	$8.00 \times 10^5 M$ cells	$4.80 \times 10^4 M$ cells
P	$2.00 \times 10^5 M$ cells	$4.60 \times 10^4 M$ cells

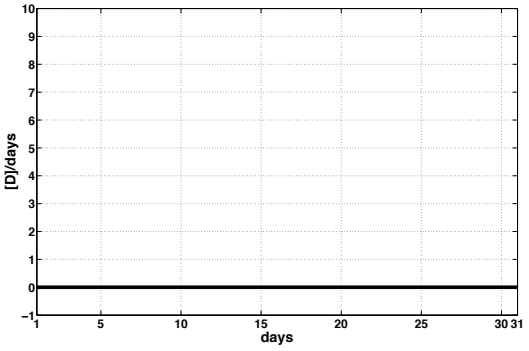

Fig. 1. Drug A Schedule

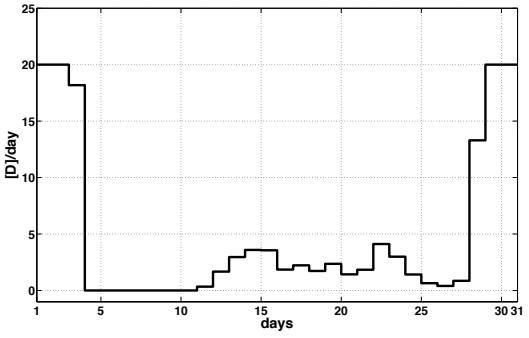

Fig. 2. Drug B schedule

bundle methods available (Mäkelä, 2001). These variants will be investigated in addition to different methods to handle nonlinear constraints. In this paper, a penalty approach is described. Improvements to this approach will be considered and the improvement function approach will be investigated (Mäkelä, 2001).

The performance of the proposed algorithm will be compared to the performance of the available transcription-based algorithms. It is expected that the method introduced in this document can be advantageous in problems with a large number states and candidate dynamics.

REFERENCES

Avraam, M.P., Shah, N., and Pantelides, C. (1998). Modelling and Optimisation of General Hybrid Systems in the Continuous Time Domain. *Computers and Chemical Engineering*, 22(Suppl), S221–S228.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.

Betts, J.T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance Control and Dynamics*, 21(2), 193–207.

Clarke, F.H. (1990). *Optimization and Nonsmooth Analysis*. Number 5 in Classics in Applied Mathematics. SIAM.

Coddington, E.A. and Levinson, N. (1955). *Theory of Ordinary Differential Equations*. McGraw Hill Co., Inc., New York.

Dua, P., Dua, V., and Pistikopoulos, E.N. (2008). Optimal delivery of chemotherapeutic agents in cancer. *Computers & Chemical Engineering*, 32(1-2), 99–107.

Feehery, W.F., Tolsma, J.E., and Barton, P.I. (1997). Efficient sensitivity analysis of large-scale differential-algebraic systems. *Applied Numerical Mathematics*, 25(1), 41–54.

Galán, S., Feehery, W.F., and Barton, P.I. (1999). Parametric sensitivity functions for hybrid / discrete / continuous systems. *Applied Numerical Mathematics*, 31, 17–47.

Kiwiel, K.C. (1985). *Methods of Descent for Nondifferentiable Optimization*, volume 1133 of *Lecture Notes in Mathematics*. Springer-Verlag.

Lukšan, L. and Vlček, J. (2001). Algorithm 811: NDA: Algorithms for nondifferentiable optimization. *ACM Transactions on Mathematical Software*, 27(2), 193–213.

Mäkelä, M.M. (2001). Survey of Bundle Methods for Nonsmooth Optimization. *Optimization Methods and Software*, 17(1), 1 –29.

Mifflin, R. (1977). Semismooth and Semiconvex Functions in Constrained Optimization. *SIAM Journal of Control and Optimization*, 15(6), 959–972.

Pang, J.S. and Stewart, D.E. (2008). Differential variational inequalities. *Mathematical Programming*, 113(2), 345–424.

Pang, J.S. and Stewart, D.E. (2009). Solution dependence on initial conditions in differential variational inequalities. *Mathematical Programming*, 116(1-2), 429–460.

Pannetta, J. and Adam, J. (1995). A mathematical model of cycle-specific chemotherapy. *Mathematical and Computer Modelling*, 22(2), 67–82.

Park, T. and Barton, P.I. (1996). State event location in differential-algebraic models. *ACM Trans. Model. Comput. Simul.*, 6(2), 137–165.

Qi, L. (1993). A nonsmooth version of Newton's method. *Mathematical Programming*, 58, 353–367.

Raghunathan, A., Diaz, M., and Biegler, L. (2004). An MPEC formulation for dynamic optimization of distillation operations. *Computers & Chemical Engineering*, 28(10), 2037–2052.

Rockafellar, R.T. and Wets, R.J.B. (1998). *Variational Analysis*. Number 317 in Grundlehren der mathematischen Wissenschaften. Springer.

Schumacher, J.M. (2004). Complementarity systems in optimization. *Mathematical Programming*, 101(1), 263 – 295.

Tolsma, J. and Barton, P.I. (2000). DAEPACK: an open modeling environment for legacy models. *Industrial & Engineering Chemistry Research*, 39(6), 1826–1839. (http://yoric.mit.edu/daepack/daepack.html).

Tolsma, J.E. (2001). DSL48SE manual (version 1.0). Technical report, Process Systems Engineering Laboratory, Department of Chemical Engineering, Massachusetts Institute of Technology. (http://yoric.mit.edu / daepack / download / Manuals_Pres / dsl48se.ps).

Tolsma, J.E. and Barton, P.I. (2002). Hidden discontinuities and parametric sensitivity analysis. *SIAM Journal on Scientific Computing*, 23(6), 1861–1874.