# LECTURE 5: OPTIMIZATION

PDF available at http://math.la.asu.edu/∼chris/CIME09/CIME09.htm

# CONTENTS

- Optimize cost functional with the network model acting as a constraint.

- Two approaches

  – Adjoint calculus.

  – Mixed integer programming.

- Examples:

  – A simple model for re-entrance.

  – Bankruptcies in networks.

## Optimization

Standard problem:

- Given a model for the flux through a production system, optimize influx and policies to meet a certain projected demand.
- Models based on differential equations can relatively easily be optimized using an adjoint calculus formulation.
- Optimize a cost functional with constraints given by a conservation law.

Example:

$$\partial_t u + \partial_x \phi(x,t,u) = 0, \quad \phi(0,t,u) = q(t)$$

$$J = A \int_0^T [\phi(1,t,u) - d(t)]^2 \; dt + B \int_0^T dt \; \int_0^1 dx \; u(x,t) \to \min$$

$q(t)$: control variable, $d(t)$: given demand, $J$: cost functional

# CONTENTS

- Optimize cost functional with the network model acting as a constraint.

- Two approaches

  – Adjoint calculus.

  – Mixed integer programming.

- Examples:

  – A simple model for re-entrance.

  – Bankruptcies in networks.

## Abstract setting:

$$J(u,q) \rightarrow \text{min}, \quad G(u,q) = 0$$

$J$: cost functional, $q$: control variable, $G$: constraint (conservation law)

- Spaces: $u \in \mathcal{H}_u$, $q \in \mathcal{H}_q$

$$J: \mathcal{H}_u \times \mathcal{H}_q \rightarrow \mathbb{R}, \quad G: \mathcal{H}_u \times \mathcal{H}_q \rightarrow \mathcal{H}_G$$

- Assumption: $G(u,q)$ has a well defined inverse for fixed $q$

$$G(u,q) = 0 \iff u = F(q), \quad G(F(q),q) = 0$$

- $\Rightarrow$ minimize $\mathcal{L}(q) = J(F(q),q)$

Assumption: We can solve the unconstrained minimization problem $\mathcal{L}(q) \rightarrow \min$ if we can evaluate $\mathcal{L}(q)$ and $\nabla\mathcal{L}(q)$ for any $q$. (Steepest descent, conjugate gradient $\rightarrow$ canned software).

# Computing the gradient $_{07}$

- Define the gradient $\nabla\mathcal{L}$ as the adjoint of the Frechet derivative

$$\nabla\mathcal{L}(q) = D_q\mathcal{L}(q)^*1$$

$$D_q\mathcal{L} : \ \mathcal{H}_q \to \mathbb{R}, \quad D_q\mathcal{L}^* : \ \mathbb{R} \to \mathcal{H}_q,$$

- Problem: $u = F(q)$ can be evaluated for each $q$ (solve $G(u,q) = 0$), but not derivative $D_qF$.

- However: $\nabla\mathcal{L}$ can be evaluated easily by solving the adjoint differential equation backwards.

$$\boxed{\mathcal{L}(q) = J(F(q), q) \Rightarrow D_q\mathcal{L} = D_u J(u,q) D_q F + D_q J(u,q), \quad u = F(q)}$$

Differentiating the constraint:

$$G(F(q), q) = 0 \Rightarrow D_u G(u,q) D_q F + D_q G(u,q) = 0, \ \forall q, \quad u = F(q)$$

$$\Rightarrow D_q\mathcal{L}(q) = -D_u J D_u G^{-1} D_q G + D_q J,$$

$$\boxed{\nabla\mathcal{L} = -D_q G^* D_u G^{-*} D_u J^* + D_q J^*, \quad D_q J^* \in \mathcal{H}_q, \quad D_u J^* \in \mathcal{H}_u}$$

---

$$\boxed{\nabla \mathcal{L} = -D_q G^* D_u G^{-*} D_u J^* + D_q J^*, \quad D_q J^* \in \mathcal{H}_q, \quad D_u J^* \in \mathcal{H}_u}$$

- $Y = -D_u G^{-*} D_u J^* \in \mathcal{H}_G$,

- $Z = D_q G^* Y \in \mathcal{H}_q$,

- $\nabla \mathcal{L} = Z + D_q J^*$,

## Weak formulation:

$$\boxed{(1)\ \langle D_u G\psi, Y\rangle + D_u J\psi = 0,\ \forall\psi \in \mathcal{H}_u \Rightarrow Y}$$

$$\boxed{(2)\ \langle \Delta q, Z\rangle = \langle D_q G \Delta q, Y\rangle,\ \forall\Delta q \in \mathcal{H}_q \Rightarrow Z}$$

- (1) involves the solution of the backward adjoint differential equation.

- (2) involves usually the numerical evaluation of integrals.

# Example: 12

- $G$ IBVP for a system of conservation laws, controlled by the influx and the flux function (routing).

- Track a certain projected demand.

$$G(u, q) = 0 \iff \partial_t u + \partial_x \phi(x, t, u, q) = 0,$$

$$u(x, 0) = a(x), \quad \phi(0, t, u) = b(t, q)$$

$$J(u, q) = \frac{1}{2} \int_0^T |\phi(1, t, u, q) - d(t)|^2 \, dt$$

# The computation of $Y$: 14

Have to compute $Y$ from

$$\boxed{\langle D_u G\psi, Y\rangle + D_u J\psi = 0, \ \forall \psi \in \mathcal{H}_u}$$

**Weak formulation of the constraint equation:**

$$\langle G(u,q), Y\rangle = 0, \ \forall Y$$

$$\langle G, Y\rangle = \int_0^1 Y(x,0)^T a - Y^T u(x,\tau) \ dx + \int_0^\tau Y^T(0,t)b - Y^T \phi(1,t,u,q) \ dt$$

$$+ \int_0^\tau dt \ \int_0^1 \ dx \ [u^T \partial_t Y + (\partial_x Y)^T \phi(x,t,u,q)]$$

**The derivative of $G$:**

$$\boxed{\langle D_u G\psi, Y\rangle = -\int_0^1 Y^T \psi(x,\tau) \ dx - \int_0^\tau Y^T D_u \phi(1,t,u,q)\psi(1,t) \ dt}$$

$$\boxed{+\int_0^\tau dt \ \int_0^1 \ dx \ [\psi^T \partial_t Y + (\partial_x Y)^T D_u \phi(x,t,u,q)\psi]}$$

**The derivative of the functional $J$**

$$D_u J(u,q)\psi = \int_0^\tau [\phi(1,t,u.q) - d(t)]^T D_u \phi(1,t,u,q)\psi(1,t) \ dt$$

# The adjoint backward problem $_{16}$

$$\langle D_u G\psi, Y\rangle + D_u J\psi = 0, \ \forall \psi \in \mathcal{H}_u$$

$$\boxed{\partial_t Y + D_u\phi(x,t,u,q)\partial_x Y^T = 0,}$$

$$\boxed{Y(x,\tau) = 0, \quad D_u\phi(1,t,u)^T[-Y(1,t) + \phi(1,t,u,q) - d] = 0}$$

Remark: If the forward equation is well posed ($\exists F(q)$) than the adjoint equation is well posed as well.

the computation of $Z$:

$$(2)\ \langle \Delta q, Z\rangle = \langle D_q G \Delta q, Y\rangle,\ \forall \Delta q \in \mathcal{H}_q \Rightarrow Z$$

$$\langle D_q G \Delta q, Y\rangle = \int_0^\tau Y^T(0,t) D_q b \Delta q \ dt$$

$$+ \int_0^\tau dt \ \int_0^1 \ dx \ [(\partial_x Y)^T D_q \phi(x,t,u,q)\Delta q]$$

if $q \in \mathbb{R}^K$:

$$\boxed{Z = \int_0^\tau D_q b^T Y(0,t) \ dt + \int_0^\tau dt \ \int_0^1 \ dx \ [D_q \phi(x,t,u,q)^T \partial_x Y]}$$

# Summary 

**Given** $q$

- **Solve**

$$\partial_t u + \partial_x \phi(x,t,q,u) = 0, \quad u(x,0) = a(x), \quad u(0,t) = b(t,q)$$

$\Rightarrow u = F(q)$

- **Solve the adjoint backward problem**

$$\partial_t Y + D_u \phi(x,t,u,q)^T \partial_x Y = 0,$$

$$Y(x,\tau) = 0, \quad D_u \phi(1,t,u)^T [-Y(1,t) + \phi(1,t,u,q) - d] = 0$$

**for** $Y$

- **Compute** $\nabla \mathcal{L}$ **from**

$$\nabla \mathcal{L} = Z = \int_0^\tau D_q b^T Y(0,t)\ dt + \int_0^\tau dt \ \int_0^1 \ dx\ [D_q \phi(x,t,u,q)^T \partial_x Y]$$

# CONTENTS

- Optimize cost functional with the network model acting as a constraint.

- Two approaches
  - Adjoint calculus.
  - Mixed integer programming.

- Examples:
  - A simple model for re-entrance.
  - Bankruptcies in networks.

# Example [21]

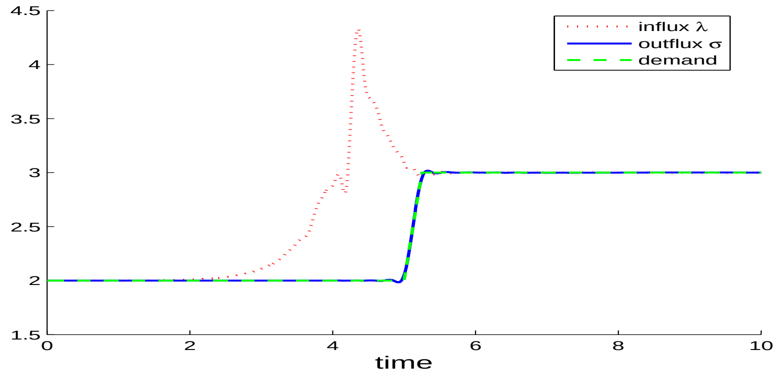
A very simplified model of a production chain

$$\partial_t \rho + \partial_x \phi = 0, \quad \phi(x,t,\rho) = \frac{\rho(x,t)}{1 + \int_0^1 \rho(y,t)\ dy}, \quad \phi(0,t,\rho) = q(t)$$

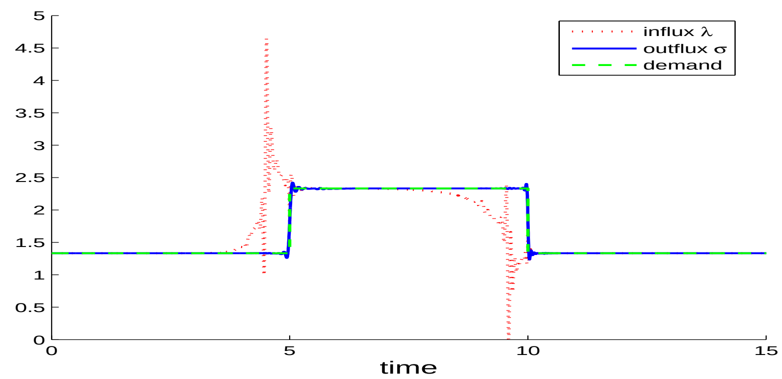
- Models a highly re-entrant system, where the velocity of a part is determined by the load in the total system.

- Given a certain demand, optimize the influx as a control parameter to minimize

$$\int_0^\tau [\phi(1,t,\rho) - d(t)]^2\ dt$$




Optimized influx, anticipating a sudden jump in demand.




Step up and step down.

# CONTENTS

- Optimize cost functional with the network model acting as a constraint.

- Two approaches

  - Adjoint calculus.

  - Mixed integer programming.

- Examples:

  - A simple model for re-entrance.

  - Bankruptcies in networks.

Linear program:

- Minimize a linear cost functional in $\mathbb{R}^N$ within a domain bounded by $K$ hyperplanes
$$u^T x \to min, \quad a_k^T x \leq c_k$$
by searching along the vertices of the resulting $N-$ dimensional polygon.

- $+$: Very efficient $N >> 1$.

- $-$: Restricted to the linear case.

- Today: replaced by more efficient (interior point) methods.

## MIP: 26

Constraints dependent on $J$ additional binary variables to be optimized

$$\boxed{u^T x \to min, \quad a_k(\overrightarrow{\xi})^T x \le c_k, \quad \overrightarrow{\xi} = (\xi_1, .., \xi_J) \in \{0,1\}^J}$$

Allows for the implementation of piecewise linear constraints (and cost functionals).

Basic idea:

Implement the function $U(a, b) = \min\{a, b\}$

Introduce $U$ as an independent variable, and an auxiliary binary variable $\xi \in \{0, 1\}$, in the program, satisfying the constraints

$$\boxed{a - M\xi \leq U \leq a, \quad b - M(1 - \xi) \leq U \leq b, \quad M >> 1}$$

Proposition: The only solution to these constraints is

$$U = \min\{a, b\}$$

Remark:

- This allows the implementation of any convex (concave) piecewise linear function, and therefore the approximation of nonlinear functions.

- Introduces, however a lot of additional variables.

# CONTENTS

- Optimize cost functional with the network model acting as a constraint.

- Two approaches

  - Adjoint calculus.

  - Mixed integer programming.

- Examples:

  - A simple model for re-entrance.

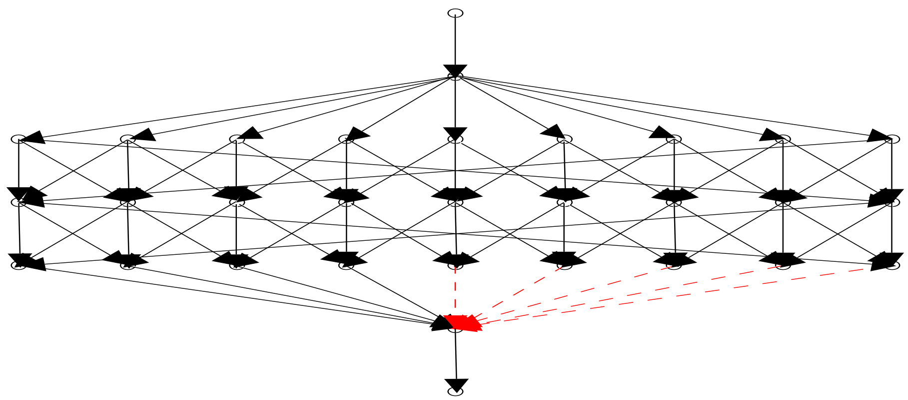
  - Bankruptcies in networks.

## Example 30

Bankruptcy cascades in a distributed supply network (Battiston).

- 3 levels of production
- Suppliers on each level can order raw material from a defined set of suppliers on the previous level.
- Each supplier has a limited capacity to produce, a capital, and a defined set of costs.
- The capital is dynamically defined via the influx of price and the outflux of costs.

- If the capital falls below a threshhold, the supplier goes into bankruptcy, and is unable to order.



Network structure

# The model 33

- Each node $j$ is governed by a conservation law with a maximum production capacity $c_j$.
- Each node can place orders to a defined set of nodes.
- Each node has a front end (input) inventory to be able to order more than his capacity.
- Each node has a back end (output) inventory to store over-produced items.
- Each node has to pay a defined price / item to its suppliers and receives a price /item from its customers if he delivers.
- This gives the node a certain capital. Bankruptcy occurs if the capital falls below a certain (negative) threshold.
- In this case the node has to stop ordering.

# Conservation law 35

$$\partial_t \rho_j + \partial_x \phi_j = 0$$

Use $\phi_j = \min\{c_j, v_j\rho_j$ as model to capture simple time delays, using a first order upwind difference scheme.

$$\boxed{\rho_{jm}(t + \Delta t) = \rho_{jm}(t) + \frac{\Delta t}{\Delta x}(\phi_{jm} - \phi_{j,m-1}),\ \ j = 1 : M}$$

$$\phi_{jm} = \min\{c_j, v_j\rho_{jm}\},\ \ m = 1 : M,$$

$\phi_{0m}$: influx

Proposition:

$$\boxed{v_j \rho_{jm} \le c_j, \;\; m = 2 : M}$$

This implies that the conservation law can be replaced by

$$\rho_{j1}(t + \Delta t) = \rho_{j1}(t) + \frac{\Delta t}{\Delta x}(\min\{c_j, v_j \rho_{j1}\} - \phi_{j,0}),$$

$$\rho_{jm}(t + \Delta t) = (1 - \frac{\Delta t v_j}{\Delta x})\rho_{jm}(t) + \frac{\Delta t v_j}{\Delta x}\rho_{j,m-1}(t)$$

For the optimal Courant number $\frac{\Delta t v_j}{\Delta x} = 1$ this just shifts the product through a register. What flows out of the first cell leaves the last cell after $M$ time steps, or after a time $M\Delta t = \frac{\Delta t}{\Delta x} = \frac{1}{v_j}$

The contents of the first cell $\Delta x \rho_{j1}$ can be regarded as an input inventory, storing the excess product, if more than the capacity is ordered.

# The output inventory:39

If more / less is produced than orders received the difference is stored in $I^{out}$

$$\boxed{\partial_t I_j^{out} = \phi_{jM}(t) - \sum_k \omega_{jk}}$$

Remark: $I_j < 0$ denotes a backlog in production. The actual product outflux of $I_j^{out}$ is given by $H(I_j^{out}) \sum_k \omega_{jk}$

$\omega_{jk}$: rate of orders from supplier $k$ to supplier $j$.

$$\begin{pmatrix} \omega_{1k} \\ . \\ \omega_{Nk} \end{pmatrix}(t) = \sum_{m=1}^{S_k} e_{s_m} q_m(t)$$

Actual product influx:

$$\boxed{\phi_{j0}(t) = \sum_k H(I_k)\omega_{kj}}$$

## Capital [41]

$$\partial_t C_j = p_j H(I_j) \sum_k \omega_{jk} - \sum_k p_k H(I_k) \omega_{kj}$$

$$-\alpha_j \int_0^1 \rho_j(x,t)\ dx - \beta_j p_j H(-I_j) \sum_k \omega_{jk}$$

$p_j$: price supplier $j$ charges.

$\alpha_j$: production cost

$\beta_j$: discount for late delivery

## Bankruptcy 42

- If the capital sinks below a certain (negative) threshold orders, are terminated.

- Replace $\omega_{kj}$ by

$$\tilde{\omega}_{kj} = \omega_{kj} H(C_j + B_j)$$

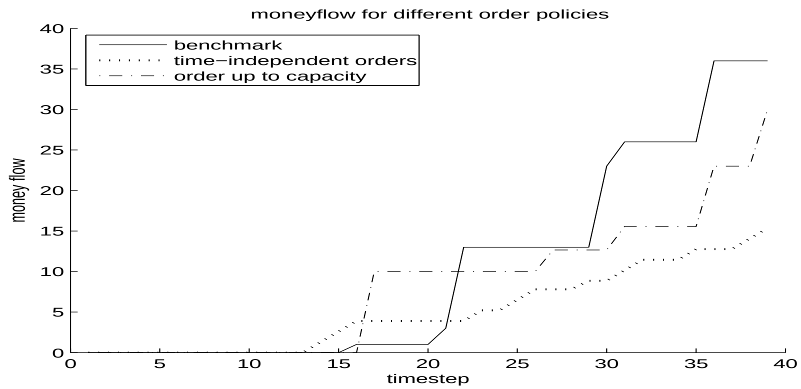
## Cost functional [43]

$$J = -\sum_j C_j(T_{end})$$

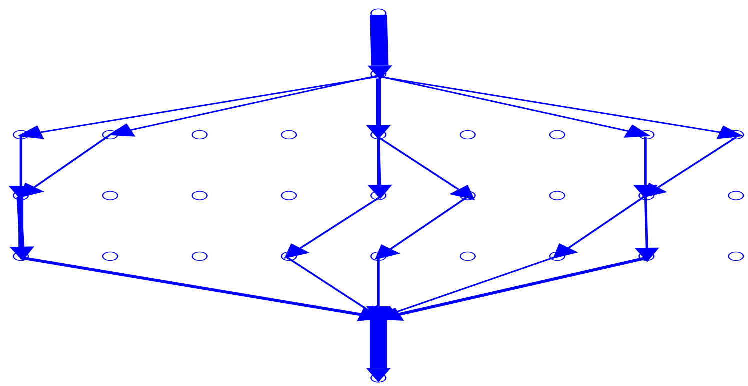
## Control parameters:

Rates $q_m(t)$ at which each node orders from its given set of suppliers.

$$\begin{pmatrix} \omega_{1k} \\ . \\ \omega_{Nk} \end{pmatrix}(t) = \sum_{m=1}^{S_k} e_{s_m} q_m(t)$$




Total capital over time for Inventory threshhold, order up to policy, and global optimization.



Survivors