An Introduction to Bi-level programming in Constrained Control

M. Hovd
TTK18 – Optimaliseringsbasert reguleringsdesign og analyse
Høst 2019

1

Outline

- What is bi-level programming?
- Reformulation of lower-level problem
- Solution strategies, heuristics, for 'simple' problems
- Examples from recent literature on constrained control

2

What is a bi-level programming problem?

- An optimization problem whose formulation depends on the solution of another optimization problem
  - Commonly referred to as 'upper-level' (UL) and 'lower-level' (LL) problems

$$\min_x f_{UL}(x,z)$$
$$G_{UI}(x,z) \leq 0$$
$$G_{UE}(x,z) = 0$$
$$z = \arg\min_z f_{LL}(x,z)$$
$$G_{LI}(x,z) \leq 0$$
$$G_{LE}(x,z) = 0$$

Bi-level programming

- Origins (at least) back to Stackelberg games (economics, 1930's)
  - Competition in a market, with a 'leader' (UL) and 'follower' (LL)
- Survey by Colson et al. (2005) lists several contributions in the control domain, going back to the 1980's
  - But little used due to inherent computational difficulty
- Recent improvements in processing capacity and optimization solvers make these techniques more relevant to real-life application

Disclaimer

- Will only attempt to give a few examples of
  - Problem formulations relevant for control
  - Solution strategies (mainly 'big-M')
- Will not attempt to give a 'complete picture', neither with regards to solution techniques nor applications to control.
- Want to present bi-level optimization as a useful tool in control

Restricting problem classes

- Formulation on slide 3 uses very general optimization formulations.
- Need to restrict the classes of optimization problems considered in order to develop efficient solution techniques.
- Will throughout most of the presentation assume that the LL problem is convex and regular
  - Denote by z_U the component of the optimal argument z of the LL problem that is used in the UL problem formulation.
  - z_U should be uniquely specified when the arguments of the UL problem (x) are specified.
- Allows replacing the LL problem with corresponding KKT conditions

Restricting problem classes

- Will consider linear constraints
- Linear (w/ unique z_U) or quadratic objective functions

- Convexity/uniqueness sufficient for subsequent reformulations to apply
  - but the additional assumptions lead to `fairly standard' optimization problems

7

The KKT conditions of the LL problem

- Lagrangian function
$$\mathcal{L}(x, z, \lambda, \mu) = f_{LL}(x, z) + \lambda^T G_{LI}(x, z) + \mu^T G_{LE}(x, z)$$
- KKT conditions:
$$\nabla_z \mathcal{L}(x, z, \lambda, \mu) = 0$$
$$G_{LI}(x, z) \leq 0$$
$$G_{LE}(x, z) = 0$$
$$\lambda \geq 0$$
$$\lambda \times G_{LI}(x, z) = 0$$

8

Overall problem

$$
\begin{aligned}
\min_{x,z,\lambda,\mu} f_{UL}(x,z)& \\
G_{UI}(x,z) &\leq 0 \\
G_{UE}(x,z) &= 0 \\
\nabla_z \mathcal{L}(x,z,\lambda,\mu) &= 0 \\
G_{LI}(x,z) &\leq 0 \\
G_{LE}(x,z) &= 0 \\
\lambda &\geq 0 \\
\lambda \times G_{LI}(x,z) &= 0
\end{aligned}
$$

- Complementarity constraints highly non-linear
- May be advantageous to solve problem using this formulation
  - Software available for problems of small size
  - YALMIP/bmibnb
- For larger problems, 'Big-M'-formulation commonly used

'Big-M' formulation

- Replace nonlinear complementarity constraints with linear constraints using binary variables $s \in \{0,1\}$

$$
\begin{aligned}
G_{LI}(x,z) &\leq 0 \\
G_{LI}(x,z) &\geq -M^u(1-s) \\
\lambda &\geq 0 \\
\lambda &\leq M^\lambda s
\end{aligned}
$$

- We observe that
  - $s = 1 \Rightarrow G_{LI}(x,z) = 0$ (Inequality constraint active)
  - $s = 0 \Rightarrow \lambda = 0$ (Inequality constraint inactive)
  - Complementarity constraints fulfilled
  - Does not affect optimal solution provided M^u and M^λ sufficiently large

Overall problem with big-M formulation

$$\min_{x,z,\lambda,\mu,s} f_{UL}(x,z)$$
$$G_{UI}(x,z) \leq 0$$
$$G_{UE}(x,z) = 0$$
$$\nabla_z \mathcal{L}(x,z,\lambda,\mu) = 0$$
$$G_{LI}(x,z) \leq 0$$
$$G_{LE}(x,z) = 0$$
$$\lambda \geq 0$$
$$G_{LI}\,(x,z) \geq -M^u(1-s)$$
$$\lambda \leq M^\lambda s$$
$$s \in \{0,1\}$$

- With linear or quadratic f_{UL}, we get MILP or MIQP (respectively).
- Non-convex, np-hard, but efficient software does exist
  - E.g., CPLEX, Gurobi, NAG,...

11

Solving resulting MILP/MIQP problems

- Branch-and-bound solvers: remove known symmetries
- Restrict possible combinations of binary variables
- Use 'small' M^u and M^λ.

12

Remove known symmetries

- MILP/MIQP solvers typically use a Branch-and-Bound approach to find the solution
- Partition the search space into regions
- In each region, find
  - An upper bound to (UB) the optimal solution (typically local optimum)
  - A lower bound (LB) to the solution (from convex under-approximation of the problem)
- If LB(region i) > UB(region j) we can conclude that optimum is not in region i.
- Discard region i, sub-partition remaining regions to get improved bounds

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

13

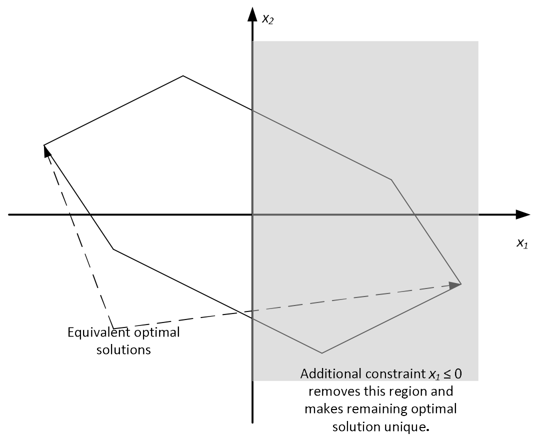
Remove known symmetries

- Symmetric problems will also have symmetric, identically valued local minima
- BnB solvers will spend a lot of effort trying to distinguish between equivalent solutions.
- Add additional constraint to make only one of the solutions feasible
  - E.g., $x_1 \leq 0$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

14

Removing known symmetries



Post-processing to recover removed optimal solution is trivial

Restrict possible combinations of binary variables

- Some combinations of active constraints are clearly absurd
  - E.g., a constrained variable cannot be at max and min value at the same time
  - Easily implemented using linear constraints on binary variables
- Depending on the problem at hand, additional constraints on binary variables may be allowable
- Reduces number of binary variable combinations that have to be investigated
  - At least in the worst case

‘Small’ M^u and M^λ

- Numerical solution will become inaccurate if M^u and M^λ are large
  - While the (unavailable) analytical solution would only require them to be sufficiently large
- Use positive diagonal matrices for M^u and M^λ
  - i.e., separate parameters for each constraint!
- Sufficiently large values for M^u readily available from solving series of LPs
  - Assuming LL constraint set is bounded
- Appropriate values for M^λ not readily available
  - May require ‘try and fail’
  - Value of λ should never be constrained by M^λ, re-run after increasing corresponding element (only) of M^λ

17

Examples from recent literature on constrainted control

Illustrating the versatility of bilevel programming.

Not a comprehensive presentation of these works.

18

Approximate Explicit MPC using Bilevel Optimization

C. N. Jones and M. Morari

Proc. European Control Conference, 2009

Brief background

- Explicit MPC an interesting alternative to conventional MPC
  - Does not require online optimization
  - Simple and efficient on-line calculations, verifiable code can be realistic
  - Identical to conventional MPC in closed loop
- Drawbacks
  - Developing the explicit (parametric) solution can be VERY demanding
  - May require VERY much memory to store the solution
  - Complex post-processing of solution to allow the (most) simple on-line calculations
  - Severity of drawbacks increase rapidly with problem size
- Desirable
  - Tractable approximation to the explicit solution
  - Trade off optimality against simplicity of solution
  - Retain closed loop stability and recursive feasibility

Jones & Morari proposal

- Start with a set of of points V whose hull $conv(V)$ describes the feasible region of the (online) MPC, or some subset thereof.
- Solve MPC problem for each of these points in the state space.
- Partition $conv(V)$ into simplices using triangulation (e.g., Delaunay, with extra point at origin)
- Linear interpolation within each simplex S_i gives piecewise affine state feedback $u = K^i x + k^i$ for $x \in S_i$. Resulting control law will
  - give feasible u
  - be continuous (also across simplex borders)
- Refine triangulation (add extra point) if
  - Input u differs too much from optimal
  - Stability cannot be proven
- Address both issues using bilevel optimization

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

21

Error from optimal input

- Optimal input u_0, approximate explicit input u
- For each S_i solve

$$\gamma_i = \max_x \|u_0 - u\|_\infty$$
$$x \in S_i$$
$$u = K^i x + k^i$$
$$u_0 = \arg\min \frac{1}{2} x_N^T Q_N x_N + \frac{1}{2}\sum_{i=0}^{N-1} u_i^T R u_i + x_i^T Q x_i$$
$$x_{k+1} = Ax_k + Bu_k$$
$$Fx_i \le f, Gu_i \le g, Hx_N \le h$$
$$x_0 = x$$

- Fits perfectly with transforming to single-level problem as shown previously
  - Inf norm in UL objective made linear using binary variables
- If error too large, introduce new point at position of largest error, and update triangulation
  - Algorithms for 'incremental' Delaunay triangulation available

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

22

Linearizing the infinity norm

- $\max_x \|u_0 - u\|_\infty$ can be reformulated as

$$\max_{x,t,s_1,s_2} \; t$$
$$\mathbf{1}t \leq u_0 - u + M(\mathbf{1} - s_1)$$
$$\mathbf{1}t \leq u - u_0 + M(\mathbf{1} - s_2)$$
$$\begin{bmatrix}\mathbf{1}\\ \mathbf{1}\end{bmatrix}^T \begin{bmatrix}s_1\\ s_2\end{bmatrix} = 1$$
$$s_1 \in \{0,1\}, s_2 \in \{0,1\}$$

(+ other constraints)

- For diagonal matrices M with sufficiently large diagonal elements

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

23

Stability

- Common to prove stability in MPC by showing that the value function is a Lyapunov function for the closed loop system
  - And for linear systems with linear constraints, combined with a quadratic objective function for the MPC, design criteria to ensure this are readily available
- Jones & Morari develop conditions on the value function of the original MPC to be a Lyapunov function for the approximate explicit MPC

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

24

Stability

- For each simplex S_i solve
$$\zeta_i = \min_{x_0} \tilde{x}_0^T Q \, \tilde{x}_0 + \tilde{u}_0^T \, R \tilde{u}_0 + J(\boldsymbol{x}, \boldsymbol{u}) - J(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}})$$
$$x_0 \in S_i$$
- Two lower level problems
  1. Ordinary MPC problem (giving $(\boldsymbol{x}, \boldsymbol{u})$)
  2. Modified MPC problem (giving $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}})$), with $\tilde{x}_0 = x_0$ and $\tilde{u}_0 = K^i x_0 + k^i$
- If $\zeta_i > 0 \; \forall i$, the value function of the original MPC is a Lyapunov function also for the system controlled with the approximate MPC
  - Otherwise, introduce an additional point at the x_0 where ζ_i attains its minimum, update triangulation, etc.
- Here, the UL problem is an *indefinite* QP. Jones & Morari show how this can be converted into a (long) series of MILPs.

Stability

- In addition to a LF, positive invariance (PI) is also required to conclude on stability.
- Positive invariance of the region in question may be known *a priori* (e.g., if it corresponds to the feasible region of a reasonably designed conventional MPC)
- Otherwise, may use that the level sets of a LF are PI.
  - Calculate LF value (MPC value function) for the boundary of V, and let J_{min} be the smallest LF value among these points. Stability is guaranteed for the subset of $hull(V)$ where the LF value < J_{min}.

Bilevel programming for analysis of low-complexity control of linear systems with constraints

H. Manum, C. N. Jones, J. Löfberg, M. Morari and S. Skogestad

48th IEEE CDC,
Shanghai, China, 2009

27

Analysis of low-complexity control...

- Similar approach to Jones & Morari ECC paper, but considers a wider range of low-complexity controllers

- Reference controllers
  - Nominal MPC
  - Robust MPC (tube MPC)
- Low-complexity controllers
  - LQR with saturation
  - Partial enumeration
    - Storing only some of the regions of the explicit MPC solution
    - If outside stored regions, choose affine controller from nearest region
  - Delaunay triangulation

28

Oops! I cannot do it again: Testing for recursive feasibility in MPC

J. Löfberg

Automatica, 48 (2012), pp. 550 - 555

Brief background

- Want to guarantee recursive feasibility for any initial state in the required operating region
  - There should exist a feasible solution to the optimization problem
  - Moreover, the resulting state trajectory should be such that feasible solutions can be found for all future times when repeatedly using the calculated MPC input (i.e. strong recursive feasibility)
- Strong recursive feasibility usually ensured by embedding a (controlled) positive invariant terminal region in the MPC formulation
  - Note: such a terminal region may restrict the feasible region and/or require a longer control horizon

Löfberg's problem formulation

- Does not require a particular terminal region
  - Any linear system with linear constraints and linear or (strictly convex) quadratic objective function.

$$U_k^* = \arg\min_{U_k} \frac{1}{2} U_k^T H U_k + U_k^T G x_k$$
$$E x_k + F U_k < b$$

$$u_{k|k}^* = [I \quad 0 \quad \cdots \quad 0] U_k^*$$

Farkas' lemma

Let $A \in \mathbb{R}^{m\times n}$, and c $\in \mathbb{R}^m$. Then either there is an x $\in \mathbb{R}^n$ such that $Ax < c$, or there exists a y $\in \mathbb{R}^m$ such that $y > 0$, $y^T A = 0$, and $y^T c < 0$.

- For given x_k and $u_{k|k}^*$, infeasibility at next timestep implies the existence of a y such that

$$y > 0, \qquad y^T F = 0, \quad y^T \left(b - E\left(Ax_k + Bu_{k|k}^*\right)\right) < 0$$

- However, must account for the fact that x_k and $u_{k|k}^*$ are coupled through optimality

Bilevel program

$$\min_{y,x_k} y^T\left(b - E\left(Ax_k + Bu_{(k|k)}{}^*\right)\right)$$
$$y > 0, \quad y^T F = 0, \quad u_{k|k}{}^* = [I \quad 0 \quad \cdots \quad 0]U_k{}^*$$
$$U_k{}^* = \arg\min_{U_k} \frac{1}{2} U_k{}^T H U_k + U_k{}^T G x_k$$
$$Ex_k + FU_k < b$$

- If the optimal objective is negative, we have found proof of infeasibility in the next timestep

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

33

Reformulation and normalization

Normalization

$$\min_{y,x_k,U_k,\lambda} y^T\left(b - E\left(Ax_k + Bu_{(k|k)}{}^*\right)\right)$$
$$y > 0, \quad y^T F = 0, \quad u_{k|k}{}^* = [I \quad 0 \quad \cdots \quad 0]U_k, \quad y^T\mathbf{1} = 1$$
$$\mathrm{H}U_k + Gx_k + F^T\lambda = 0$$
$$Ex_k + FU_k - b < 0$$
$$\lambda > 0$$
$$\lambda \times (Ex_k + FU_k - b) = 0$$

KKT conditions

- Normalization essential to make formulation numerically sound
- KKT conditions may be converted into linear constraints using binary variables (as ‘usual’)
- Bilinear UL is a problem

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

34

Reformulating the bilinear objective

- Express objective as

$$\min_{x_k, U_k, \lambda} \min_{y} \quad y^T \left(b - E\left(Ax_k + Bu_{(k|k)}{}^{*}\right)\right)$$

- The inner problem is here an LP in terms of y
  - In which we also have to account for the constraints involving y in the original problem:
    $$y > 0, y^T F = 0, y^T \mathbf{1} = 1$$
  - Constraints not involving y are moved to the outer problem

- Use strong duality of LP:
  $$\min_{y \geq 0, Ay=} \quad c^T y = \max_{c - A^T x \geq 0} \quad b^T x$$
- The 'inner' problem above therefore corresponds to
  $$\max_{z,t} t$$
  $$\text{subject to} \quad b - E\left(Ax_k + Bu^{*}_{(k|k)}\right) - Fz \; \geq t\mathbf{1}$$

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

35

Reformulating the bilinear objective

- The overall problem is thus a *min max* problem. Replacing the inner problem by its KKT conditions, the overall problem becomes

$$\min_{x_k, U_k, \lambda, z, t, \mu} \quad t$$
$$\mathrm{H}U_k + Gx_k + F^T\lambda = 0$$
$$\lambda \times (Ex_k + FU_k - b) = 0$$
$$\mu \times (b - E\left(Ax_k + Bu^{*}_{(k|k)}\right) - Fz - t\mathbf{1} = 0$$
$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} + \begin{bmatrix} F^T \\ \mathbf{1}^T \end{bmatrix} \mu = 0$$
$$\lambda \geq 0$$
$$\mu \geq 0$$

- As before, the complementarity constraints ('×' constraints) can be replaced by binary variables.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

36

Extensions

- Calculating the largest common slack that can be added to all constraints
- Robustness to disturbances
- Relaxing optimality
  - Feasible interior point approach
  - Relaxing complementarity constraints

See original paper for details

37

On the design of exact penalty functions for MPC using mixed integer programming

M. Hovd and F. Stoican

Hovd, M. and Stoican, F., Computers & Chemical Engineering, Vol 70, pp. 104 – 113, 2014

38

Background

- In MPC, with (hard) constraints on inputs and outputs, there will be a bounded region within which a solution to the optimization problem can be found.
- This is known as the *feasible region*.
- Outside the feasible region the optimization problem has no solution, and the MPC will fail (to provide an input).
  - This is not acceptable.
- Commonly handled in industrial MPC using *soft constraints*
  - Introduce slack variables in the constraints, such that feasibility can always be ensured.
  - Add penalty terms in the objective function to penalize the slack variables

Problem

- Soft constraints may allow violation of the original constraints even when this is not necessary
  - i.e., even when there exists a feasible solution to the 'hard constrained' problem
- It is known from optimization theory that a *sufficiently large* weight on constraint violations will make the penalty functions *exact*
  - i.e., the original constraints are only violated when absolutely necessary (no other solution exists).
- Has been no tractable and systematic way of determining what is *sufficiently large.* Too large weights on penalty terms may cause
  - unnecessarily violent control action when constraints are violated
  - poor numerical conditioning of the optimization problem.

Soft constrained MPC formulation

$$\min_{u,\epsilon} \frac{1}{2} u^T H u + x_0^T F u + \phi(\epsilon)$$
$$s.t.\ Gu \leq W + E x_0 + \theta(\epsilon), \epsilon \geq 0$$

- The hard constraints are fulfilled ($\epsilon = 0$ at optimum) whenever possible provided

$$\left.\frac{\partial \phi(\epsilon)}{\partial u}\right|_{\epsilon=0} = 0$$
$$\theta(0) = 0$$
$$\left.\frac{\partial \phi(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} \geq \left.\frac{\partial \theta(\epsilon)}{\partial u}\right|_{\epsilon=0} \lambda$$

- Two first conditions trivially fulfilled
- Third condition generalization of typical result in optimization textbooks
  - Weight on linear term of p-norm penalty function should exceed the maximum dual norm of Lagrangian multipliers of hard-constrained problem

l_p norms and dual norms

- The l_p norm of a vector a

$$\|a\|_p = \left(\sum_i |a_i|^p\right)^{\frac{1}{p}}$$

- Lagrangian multipliers are non-negative
  - Absolute value not needed

- Dual norm given by

$$\frac{1}{p} + \frac{1}{p_d} = 1$$

- l_1 and l_∞norms are mutually dual
- l_2 norm its own dual (but not very useful for exact penalty function design)

Typical penalty function design

$$\phi(\epsilon) = F_\epsilon^T \epsilon + \epsilon^T H_\epsilon \epsilon$$
$$\theta(\epsilon) = G_\epsilon \epsilon$$

- Quadratic term in $\phi(\epsilon)$ not critical
  - $H_\epsilon$positive definite to maintain standard QP formulation
  - May affect distribution of constraint violations
  - Does not affect whether penalty function is exact
  - Quadratic term ignored in the following
- Penalty function exact provided
$$F_\epsilon \geq G_\epsilon^T \lambda$$
- Choose $F_\epsilon = k\mathbf{1}_{d_\epsilon}, k \geq \max_\lambda \max_i \{(G_\epsilon^T \lambda)_i\}$
  - $\theta(\epsilon) = I_N \epsilon,\ \phi(\epsilon) = k\|\epsilon\|_1,\ k \geq \max_\lambda \|\lambda\|_\infty$. Exact l_1 norm penalty function.
  - $\theta(\epsilon) = \mathbf{1}_N \epsilon,\ \phi(\epsilon) = k\|\epsilon\|_\infty,\ k \geq \max_\lambda \|\lambda\|_1$. Exact l_∞ norm penalty function.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

43

Penalty function design

- $l_1$norm penalty functions add many variables to the optimization problem
  - Without increasing computational load if specially tailored QP solvers are used (Rao et al.)
  - Well behaved for difficult problems
- l_∞norm penalty functions add a single extra variable to the optimization problem
  - Strange closed loop behaviour if inverse response in constrained output
  - See Hovd & Braatz for modifications alleviating this problem
- l_2 norm not recommended for exact penalty functions
  - Due to the (non-linear) square root required to extract the linear term
  - l_1 and l_∞ norm penalty functions add only linear constraints

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

44

Bi-level programming formulation

- Want to solve

Polyhedral norm function

$$\max_{x_0} \Psi(G_\epsilon^T, \lambda)$$
$$\min_u \frac{1}{2} u^T H u + x_0^T F u$$
$$s.t.\ Gu \leq W + E x_0$$

- Upper level objective not defined until LL problem is replaced by KKT conditions.

- What is $\Psi(G_\epsilon^T, \lambda)$?
- Maximizing $\|\lambda\|_1$: $max\ \mathbf{1}^T \lambda$ (for ∞-norm penalty function)
- Maximizing $\|\lambda\|_\infty$: $\min t$

$1t \geq \lambda$ (for 1-norm penalty function)

(additional inequality $-1t \leq \lambda$ is not needed since $\lambda \geq 0$)

45

Replacing lower-level problem with KKT conditions

- Using binary variables to linearize KKT conditions, this gives

$$\max_{\lambda, u, x_0, s} \Psi(G_\epsilon^T, \lambda)$$
$$Hu + F^T x_0 + G^T \lambda = 0$$
$$\lambda \geq 0$$
$$\lambda \leq M^\lambda s$$
$$Gu - W - E x_0 \leq 0$$
$$Gu - W - E x_0 \geq -M^u (1 - s)$$
$$s \in \{0,1\}^q$$

- However, although KKT conditions uniquely determine u, they do not uniquely determine the Lagrangian multipliers
  - May be increased without bound at locations where the set of active constraints is not uniquely determined
  - Artefact of KKT conditions, not relevant to penalty function design

46

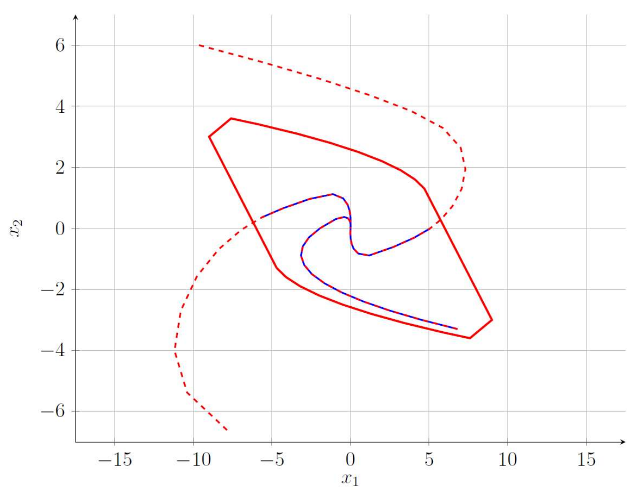
Avoiding unbounded growth of Lagrangian multipliers

- Introduce Linear Independence Constraint Qualification (LICQ)
  - Number of (strongly) active constraints cannot exceed the degrees of freedom in the optimization
  - Implemented as constraints on binary variables
  - Account also for structure of constraint matrix G
    - Block triangular structure further limits which constraints can be active simultaneously
- Include in addition physical understanding
  - E.g., the same variable cannot be at max value and min value constraint simultaneously
- In addition to making the Lagrangian multipliers unique, also reduces solution times

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

47

A double integrator example



- Trajectories are indentical inside the (hard constrained) feasible region

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

48

Numerical experience

- Good
  - Can solve problems of size of industrial relevance (> 100 constraints)
  - Good choices for M^u and M^λ critical
  - MILPs are np hard anyway...

- Avoids explicitly calculating feasible region
  - Itself a very demanding calculation
  - Have found solution (maximizing norms of Lagrangian multipliers) also for problems where the feasible region cannot be calculated with available methods.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

49

Design of Active Inputs for Set-based Fault Diagnosis

J. K. Scott, R. Findeisen, R. D. Braatz, D. M. Raimondo
American Control Conference 2013

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

50

Design of Active Inputs...

- Uses bi-level programming formulation to find inputs which ensure reliable fault diagnosis
  - Inputs are such that different faults (and the fault free condition) can be reliably identified from outputs
- See ACC paper for details

51

Non-convex lower level problems

Some illustrations of what can go wrong – and how appropriate reformulations sometimes can make the problem very simple

52

Non-convex lower level problems

- Numerically (typically) very difficult
  - Consult specialized literature if solution to such problems are required.
- KKT conditions are *only necessary conditions* for an optimum
  - Replacing LL problem with KKT conditions will typically give the UL problem the opportunity to choose a non-optimal KKT point

53

Example: scaling to maximize size of polytope inside another polytope

- Work with S. Olaru/G. Bitsoris: Have found way of finding controlled contractive regions with very simple shape
  - Controlled contractive region in the shape of a parallelepiped
  - Excellent starting point for very simple constrained control
  - May result in explicit constrained control for dimensions much larger than what is handled by ordinary explicit MPC.
- To maximize operating region of the controller, want to scale the simple polytope to cover as much as possible of the acceptable operating region
- Polytope to scale:

$$\begin{bmatrix} R \\ -R \end{bmatrix} x \leq \begin{bmatrix} k \\ k \end{bmatrix}$$

- Acceptable operating region: $Hx \leq h$

54

Possible formulation

$$\max_{k} f(k)$$ Maximize size of contractive region

$$\hat{s} \leq 0$$ Ensures contractive region is feasible

Finds x within contractive region where infeasibility w.r.t. acceptable region is maximized

$$\left\{ \begin{array}{l} \hat{s} = \max_{x} s^* \\ \begin{bmatrix} R \\ -R \end{bmatrix} x \leq \begin{bmatrix} k \\ k \end{bmatrix} \end{array} \right.$$

Measures infeasibility relative to acceptable operating region for given x

$$s^* = \left\{ \begin{array}{l} \min_{s} s \\ Hx - s\mathbf{1} \leq h \\ s \geq 0 \end{array} \right.$$

- Possibly(?) a reasonable formulation, but how to solve it?

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

55

Faulty solution strategy 1

- Replace lowest-level problem with KKT conditions, linearize using binary variables (OK)
- But middle-level problem now cannot be replaced by KKT conditions
  - Since some decision variables are binary, KKT conditions do not apply (derivative w.r.t. discrete-valued variable does not make sense)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

56

Faulty solution strategy 2

- Replace the lowest optimization problem with its dual (maximization).
  - Dual of an LP has identical optimal objective value
- Merge two lower optimization into a single maximizaton

- However, the merged maximization problem will now be an indefinite (non-convex) QP
  - Cannot be replaced by KKT conditions, since the problem is non-convex, and therefore some (many) KKT points may not be optimal solutions

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

57

Alternative formulation

$$\max_k f(k)$$
$$y_i - h_i \leq 0 \ \ \forall i = 1, \cdots, n_H$$
$$y_i = \max_x H_i x \ \ \forall i = 1, \cdots, n_H$$
$$\begin{bmatrix} R \\ -R \end{bmatrix} x \leq \begin{bmatrix} k \\ k \end{bmatrix}$$

- $n_H$lower-level problems in parallel
- Lower-level problems are convex, but replacing with KKT conditions not tractable
  - Each lower-level problem must have its own x variables and its own binary variables
  - Tractable only for *very* small problems

- In the end, the problem was solved using an alternative approach (not described here)

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

58

Sensitivity of lower level problem i

- Lagrangian $\mathcal{L}_i$ of lower-level problem i:

$$\mathcal{L}_i = H_i x - \lambda_i^T \left(\begin{bmatrix} R \\ -R \end{bmatrix} x - \begin{bmatrix} k \\ k \end{bmatrix} \right)$$

- Sensitivity at $k = k^*$
- Corresponding optimal argument x^*, active constraints $R_i^a x^* = k_i^a$.
- Observe (details omitted) that
  - Optimum is always at the same vertex of the parallelepiped. I.e., set of active constraints does not change with k
  - All elements of k associated with an active constraint, i.e.,$k_i^a = k$.

59

Sensitivity of lower level problem i

$$\frac{\partial \mathcal{L}_i}{\partial x} = -H_i + (\lambda_i^a)^T R_i^a$$

$$\lambda_i^a = [R_i^a]^{-T} H_i^T$$

$$\frac{\partial \mathcal{L}_i}{\partial \lambda_i} = x^T [R_i^a]^T - k^T$$

- Thus

$$\delta x^* = [R_i^a]^{-1} \delta k$$

- Each lower-level optimization problem supplies set of linear constraints to upper-level problem

$$H_i [R_i^a]^{-1} \delta k \leq (h_i - y_i)$$

- Starting from an initial k:
  - Linear $f(k)$: Get a simple LP
  - $f(k) = \prod_i k_i$: Easily reformulated as a series of QP problems

60

Disclaimer

- Not claiming any novelty in this last part
  - Find efficient formulation
  - This (or other) solutions may be available elsewhere
    - YALMIP documentaton?
- Just intended as an illustration of how careful problem formulation can convert a seemingly intractable problem into one that can be efficiently solved

Controlled contractive sets for low-complexity constraint control

S. Munir, M. Hovd, G. Sandou, S. Olaru

IEEE Multi-conference on Systems and Control, 2016

Finding low-complexity controlled contractive sets allowing the trade-off between set complexity and set volume.

Illustration of an approach to bi-(tri-)level programming with non-convex LL problem

A bilevel optimization approach for D-invariant set design

M.-T. Laraba, M. Hovd, S. Olaru, S.-I. Niculescu

13th IFAC Workshop on Time Delay Systems, Istanbul, Turkey, June 2016.

Bilevel programming for finding positive invariant sets for time delay systems. See original paper for details (available through IFAC Papers Online)

Main message

- Bi-level programs are a useful tool in design of constrained control
- There is probably a number of applications of bi-level programming waiting to be discovered

Additional literature and references

- S. Dempe: *Foundations of Bi-level programming.* Springer, 2002.
- J. F. Bard: *Practical bilevel optimization. Algorithmes and Applications.* Kluwer Academic, 1998
- I. Rauova: Evaluation of different bilevel optimization algorithms with application to control. Diploma thesis, STUBA, 2010
- YALMIP Wiki: http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Tutorials.BilevelProgramming
- C. Jones, M. Morari: *Approximate explicit MPC using bilevel optimization*. European Control Conference, 2009.
- J. Löfberg: Oops! I cannot do it again: Testing for recursive feasibility in MPC. *Automatica* 48(3), pp. 550-555.
- H. Manum, C. Jones, J. Löfberg, M. Morari, S. Skogestad: Bilevel programming for analysis of low-complexity control of linear systems with constraints. *IEEE Conference on Decision and Control*, 2009.
- J. K. Scott, R. Findeisen, R. D. Braatz, D. M. Raimondo: Design of active inputs for set-based fault diagnosis. *American Control Conference*, 2013.
- B. Colson, P. Marcotte, G. Savard: Bilevel programming: A survey. *4OR: A Quarterly Journal of Operations Research*, 3(2), pp. 87-107.
- J. Fortuny-Amat, B. McCarl: A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 1981, pp. 783-792.

65

Additional literature and references

- M. Hovd, F. Stoican: On the design of exact penalty function for MPC using mixed integer programming. *Computers & Chemical Engineering*, Vol 70, pp. 104 – 113.
- M. Hovd, R. Braatz: Handling state and output constraints in MPC using time-dependent weights. *American Control Conference*, 2001.
- M. Hovd, R. Braatz: On the use of soft constraints in MPC controller for plants with inverse response. *Proc. IFAC Symposium DYCOPS*, 2001.
- C. Rao, S. Wright, J. Rawlings: Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3), pp. 723 – 757
- M.-T. Laraba, M. Hovd, S. Olaru, S.-I. Niculescu: A bilevel optimization approach for D-invariant set design. 13th IFAC Workshop on Time Delay Systems, Istanbul, Turkey, 2016.
- S. Munir, M. Hovd, G. Sandou, S. Olaru: Controlled contractive sets for low-complexity constrained control. IEEE Multi-conference on Systems and Control, Buenos Aires, 2016.

Internet:www.itk.ntnu.no/ansatte/Hovd_Morten

66