Solving mixed-integer nonlinear programming (MINLP) problems

Marcel Hunting
AIMMS Optimization Specialist

AIMMS

Webinar, June 15, 2016

Overview

- > Introduction
- > MINLP solvers
- > Algorithms used by solvers
- > Improving model formulation (preprocessing, linearizations)
- > Troubleshooting

Mixed-Integer Nonlinear Program

minimize	$f(x,y)$	
subject to	$g_j(x,y) \leq 0$	$j \in J$
	$Ax + By \leq b$	
	x continuous	
	y integer	

Important special cases:

- Convex MINLP: f and g_j are convex
- MIQP: f is quadratic and g_j are linear
- MIQCP: g_j are quadratic or second-order cone (and f is linear or quadratic)

MISOCP

AIMMS

MINLP solvers (+ linear solvers)

Problem class	First choice	Second choice
Convex MIQP	CPLEX, Gurobi	AOA, BARON, Knitro
Convex MIQCP	CPLEX, Gurobi	AOA, BARON, Knitro
Convex MINLP	AOA	BARON, Knitro
Non-convex MIQP	AOA, BARON, Knitro, CPLEX	
Non-convex MINLP*	AOA, BARON, Knitro	

*: Including Non-convex MIQCP

- CPLEX and Gurobi are linear solvers
- BARON is a *global* nonlinear solver
- AOA and Knitro are *local* nonlinear solvers

MINLP solvers (+ linear solvers)

Problem class	First choice	Second choice
Convex MIQP	CPLEX, Gurobi	AOA, BARON, Knitro
Convex MIQCP	CPLEX, Gurobi	AOA, BARON, Knitro
Convex MINLP	AOA	BARON, Knitro
Non-convex MIQP	AOA, BARON, Knitro, CPLEX	
Non-convex MINLP	AOA, BARON, Knitro	

Global optimum

- CPLEX and Gurobi are linear solvers
- BARON is a *global* nonlinear solver
- AOA and Knitro are *local* nonlinear solvers

AIMMS

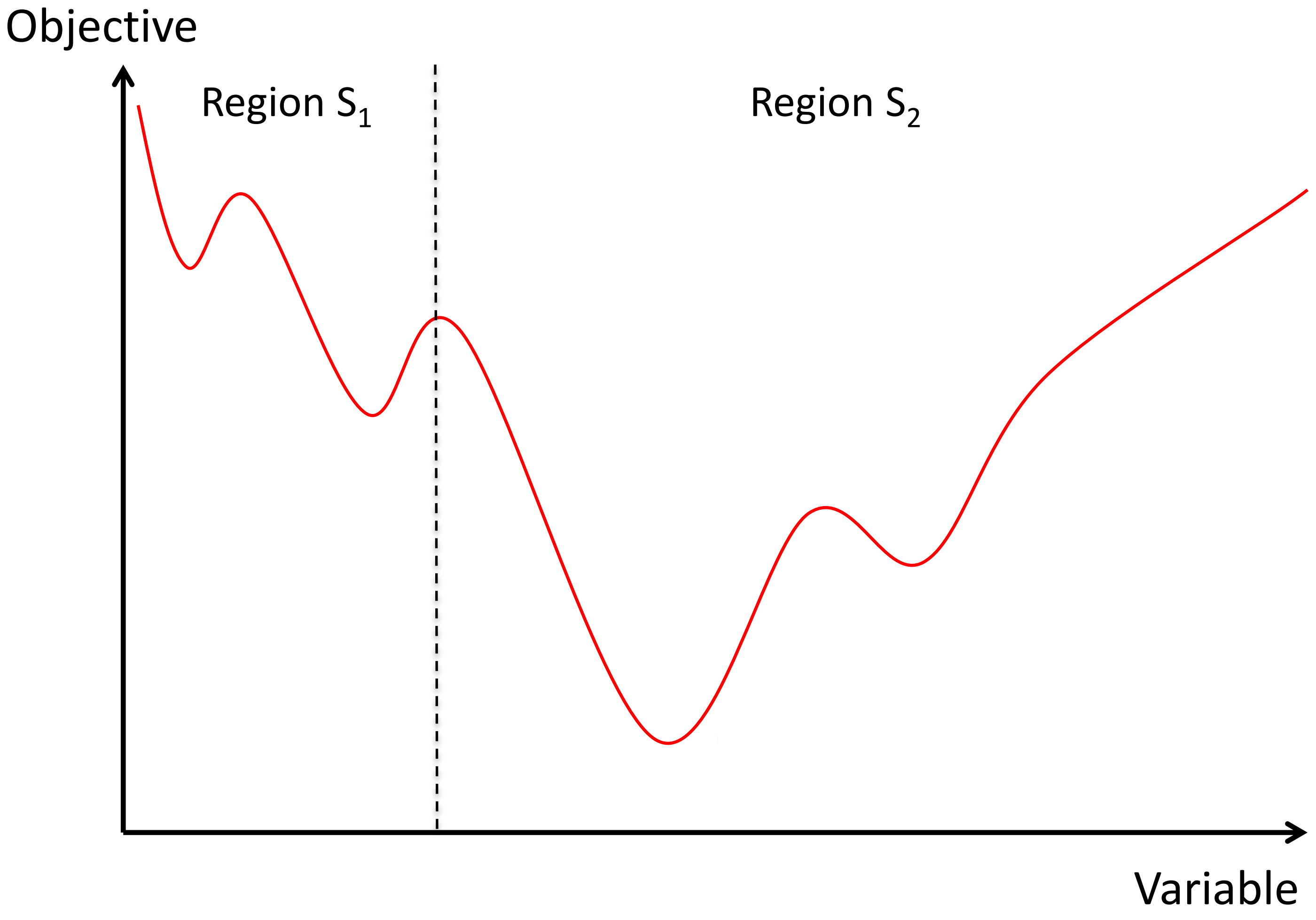
Algorithms used by Solvers

Solver	Algorithm
AOA	Outer Approximation, Quesada-Grossmann OA
BARON	Spatial Branch-and-Bound
Knitro	Branch-and-Bound, Quesada-Grossmann OA, MISQP
CPLEX	Branch-and-Cut, McCormick Relaxation Branch-and-Bound
Gurobi	Branch-and-Cut

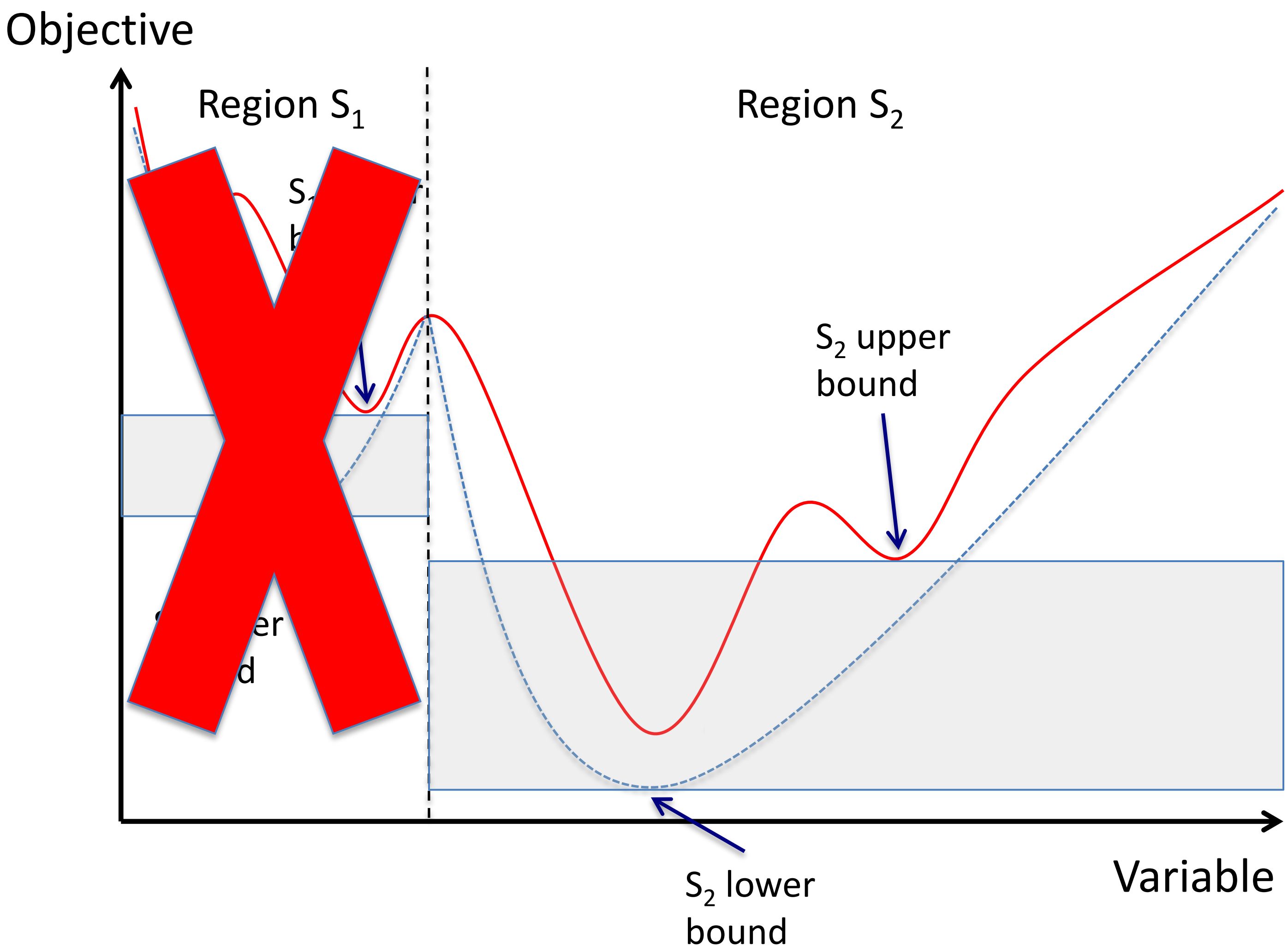
Algorithms used by Solvers

Solver	Algorithm
AOA	Outer Approximation, Quesada-Grossmann OA
BARON	Spatial Branch-and-Bound
Knitro	Branch-and-Bound, Quesada-Grossmann OA
CPLEX	Branch-and-Cut, McCormick Relaxation Branch-and-Bound
Gurobi	Branch-and-Cut

Spatial Branch-and-Bound

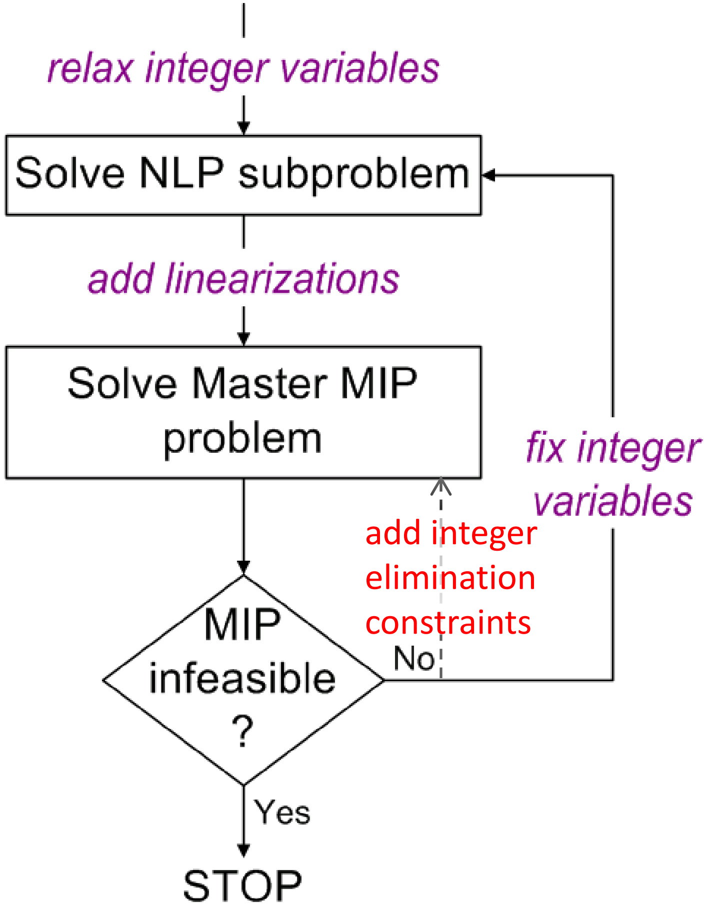


Spatial Branch-and-Bound



AIMMS

Outer Approximation



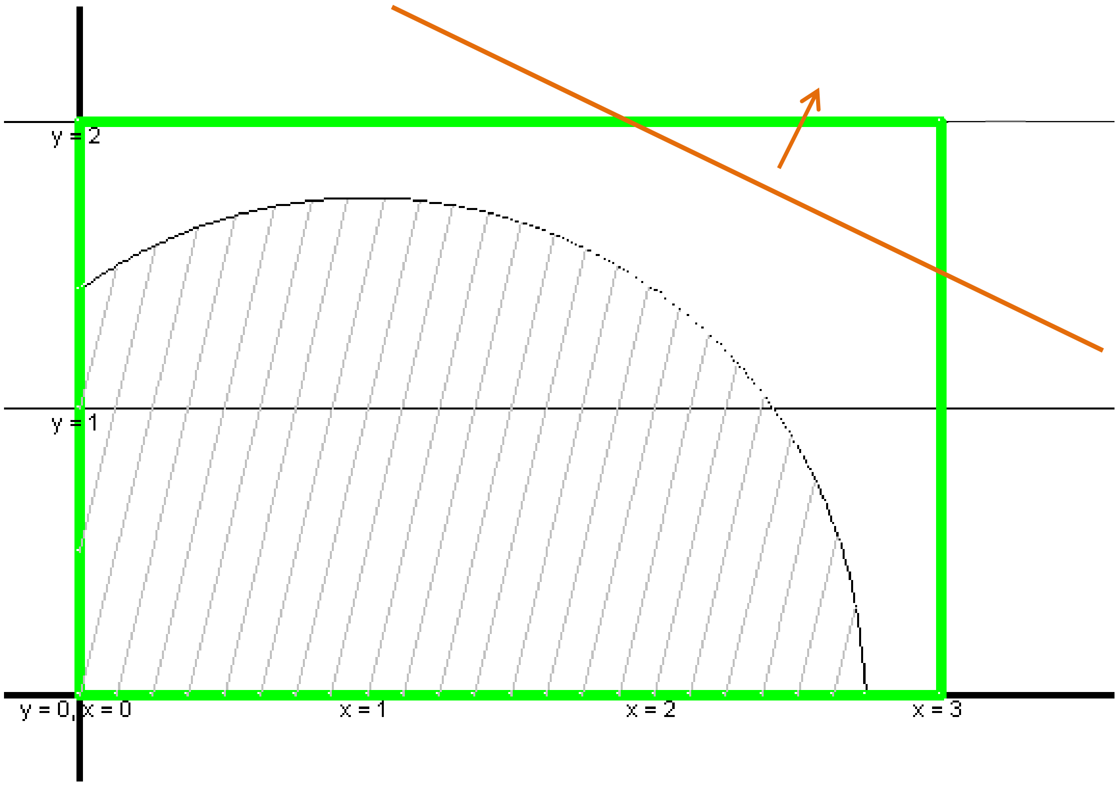
Outer Approximation: Example

max $0.5\,x + y$

s.t. $(x\text{-}1)^2 + y^2 \leq 3$

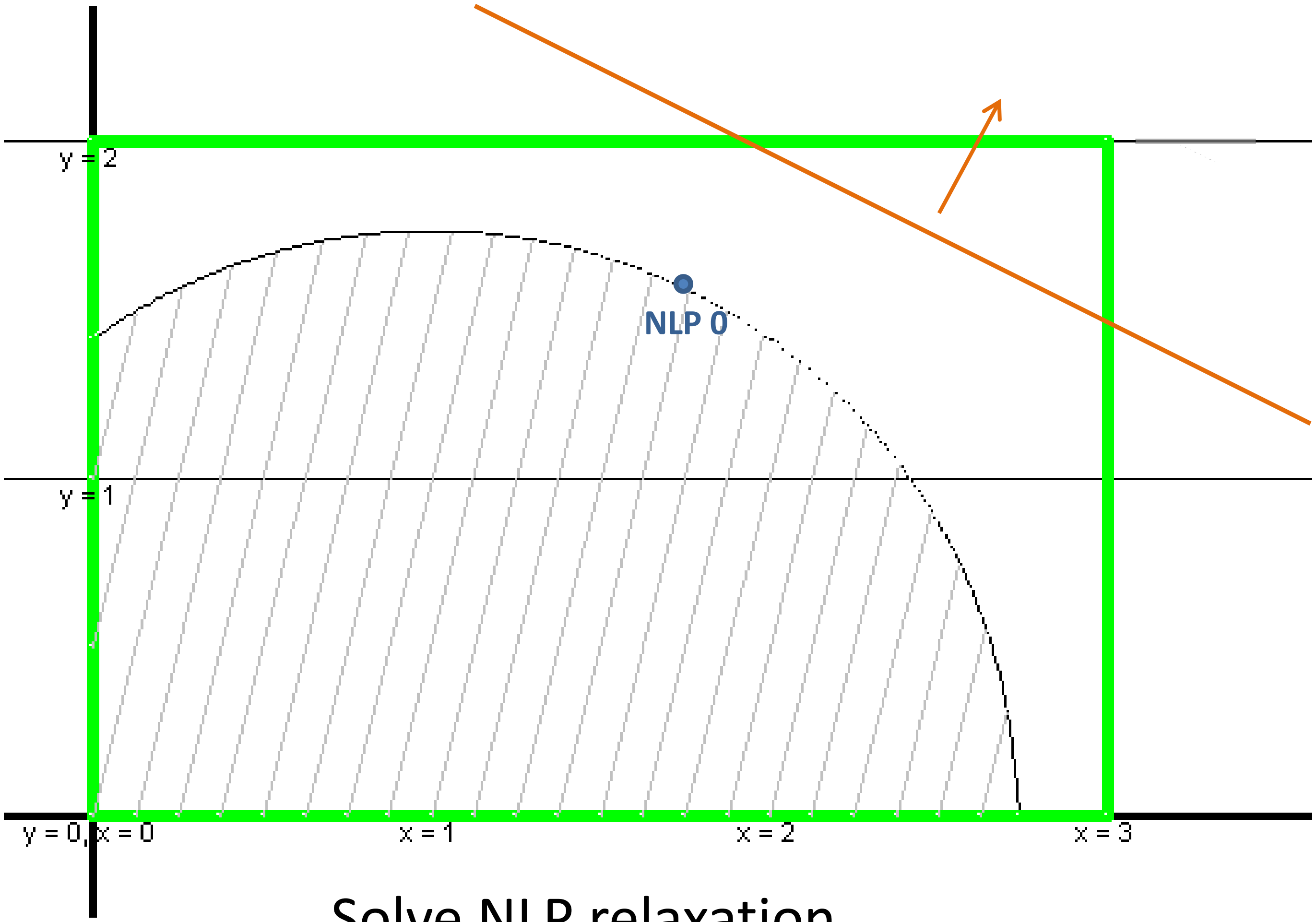
$x \in [0,3]$

$y \in \{0..2\}$



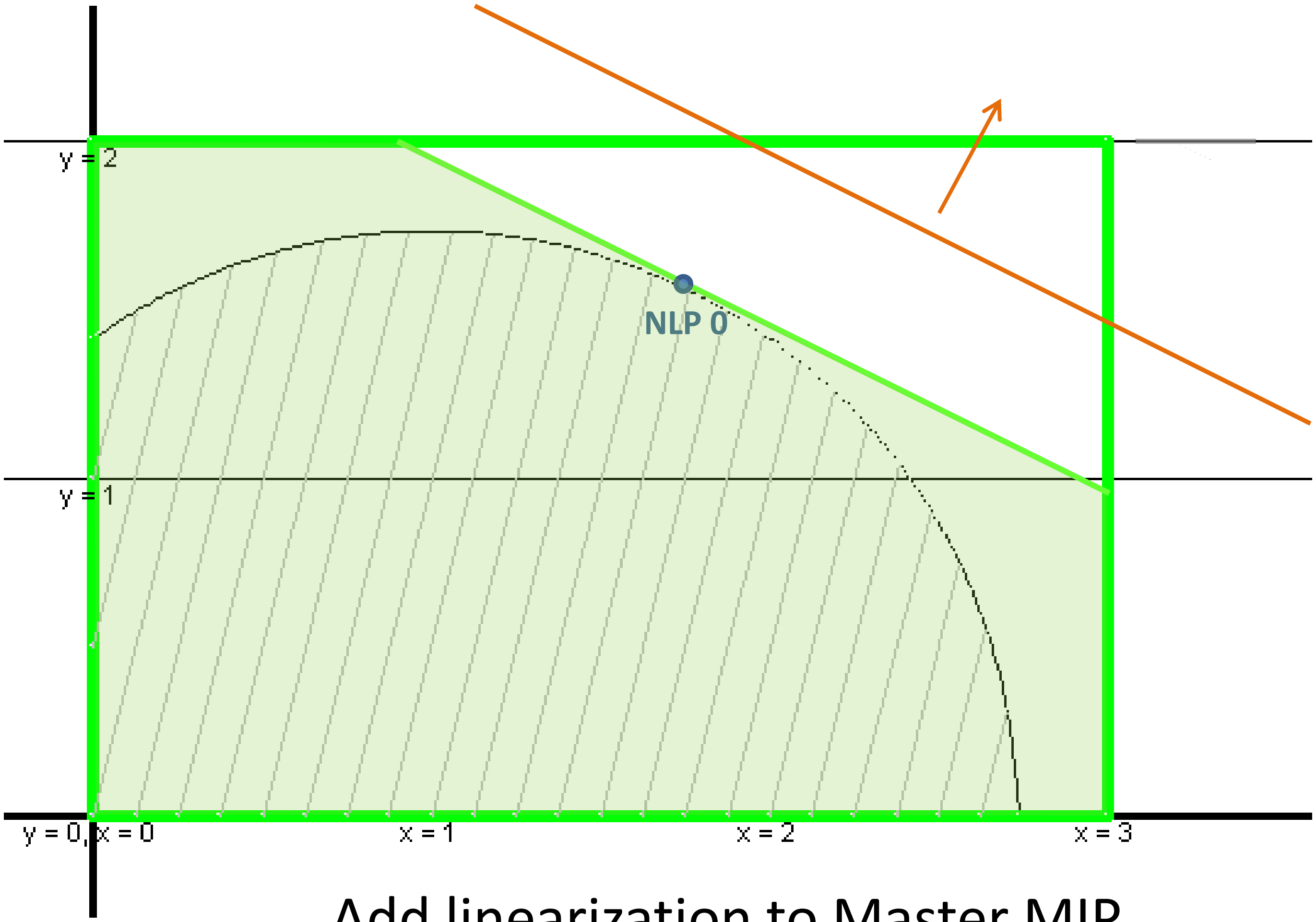
AIMMS

Outer Approximation: Example



Solve NLP relaxation

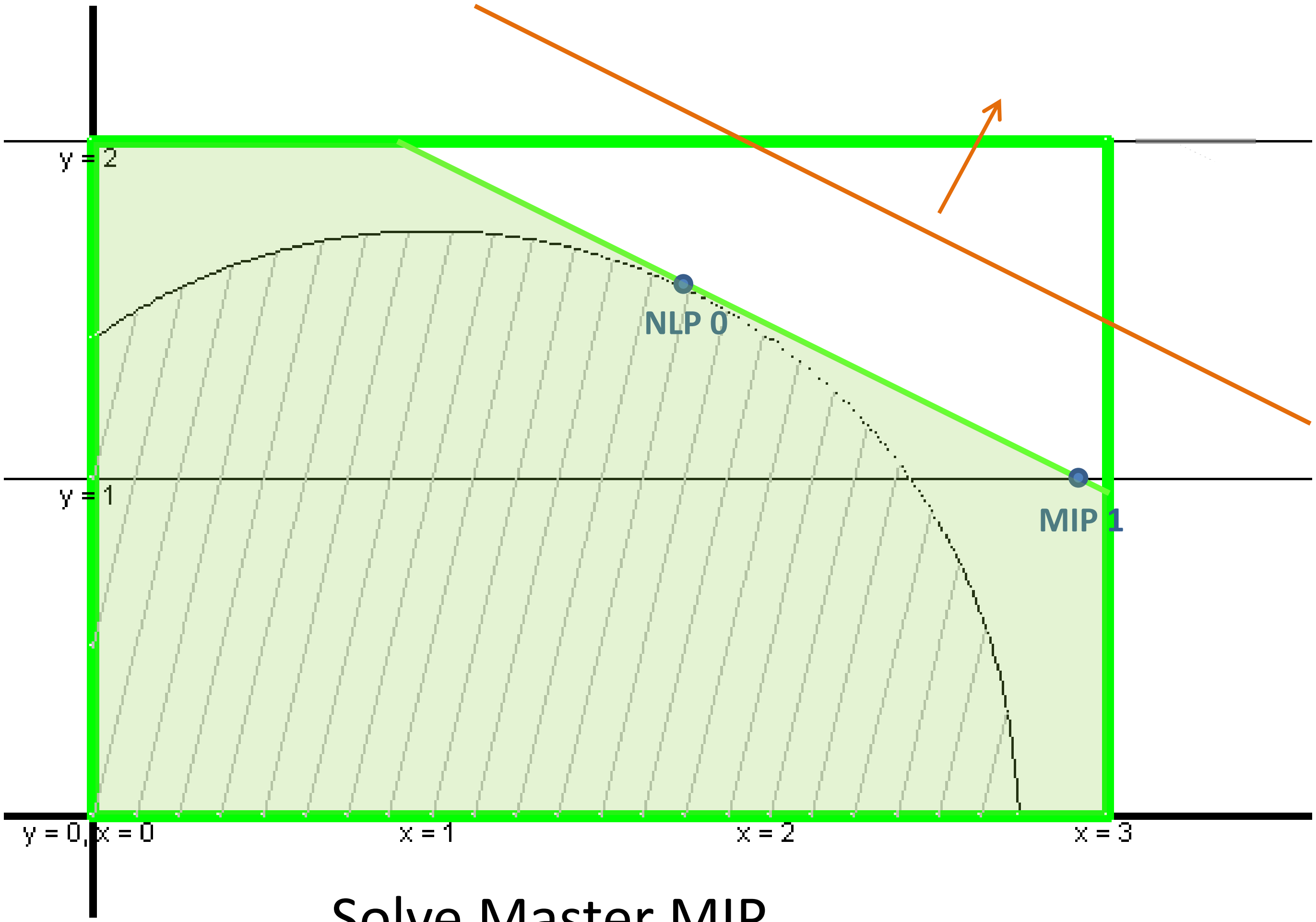
Outer Approximation: Example



Add linearization to Master MIP

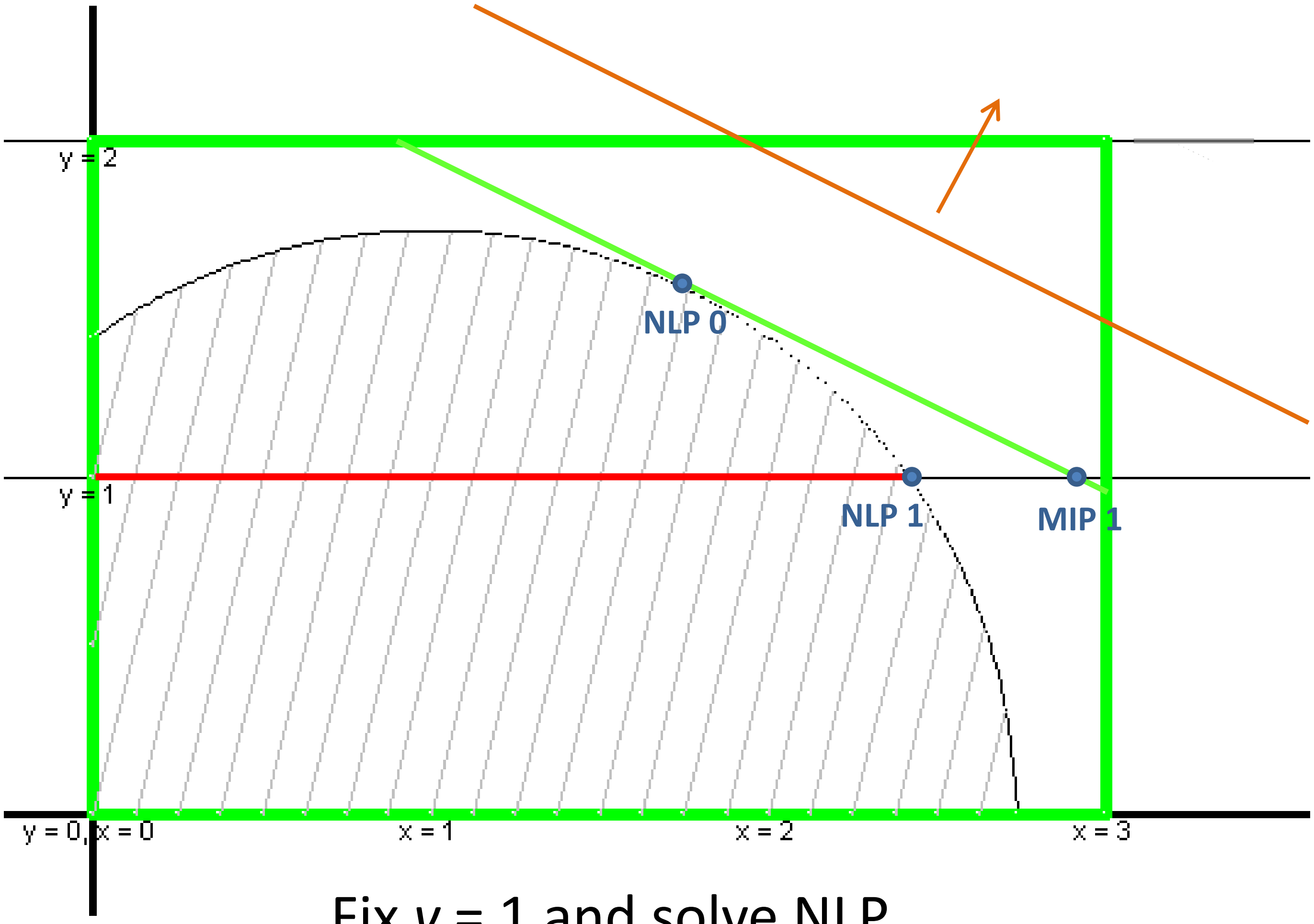
AIMMS

Outer Approximation: Example



Solve Master MIP

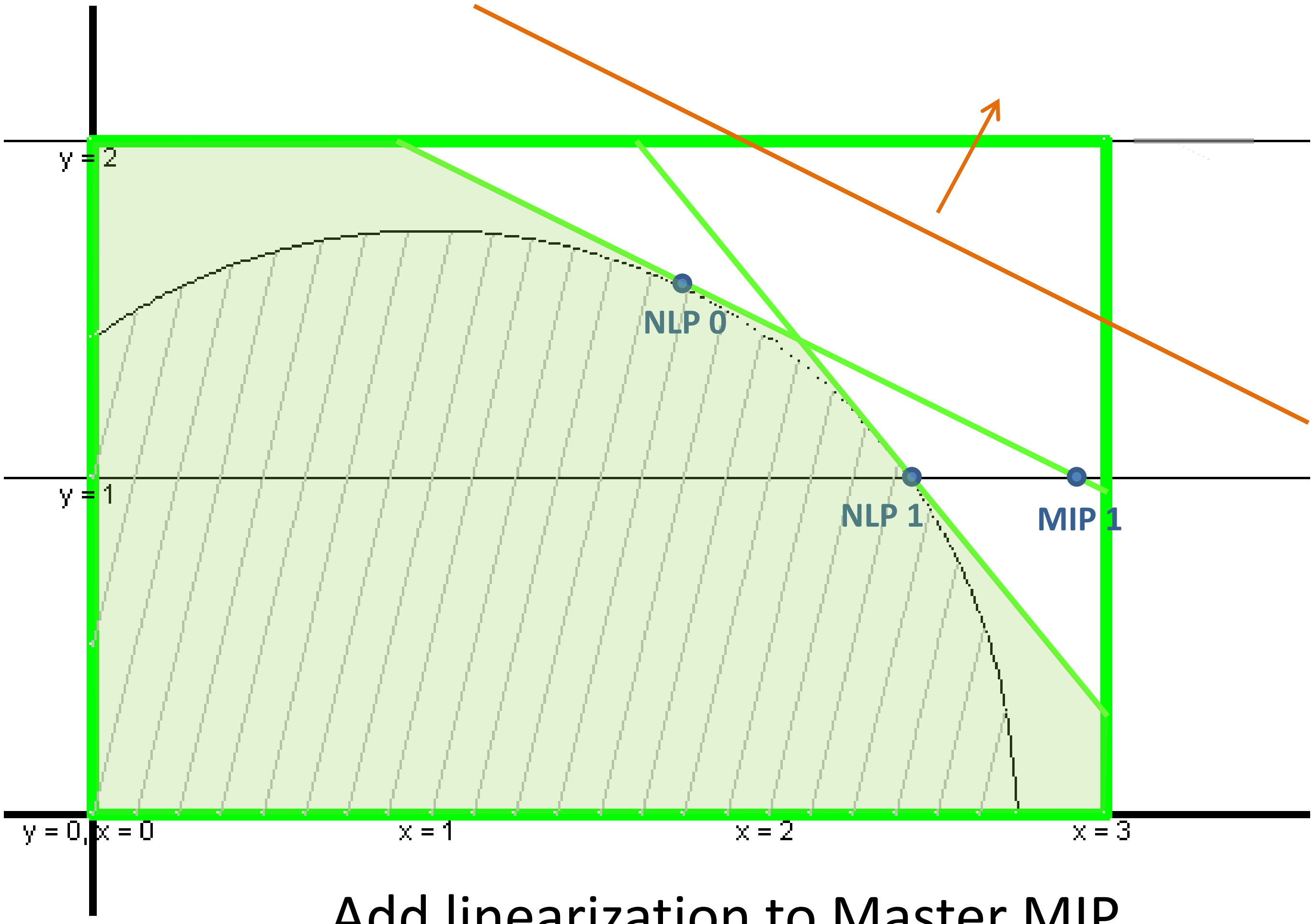
Outer Approximation: Example



Fix *y* = 1 and solve NLP

AIMMS

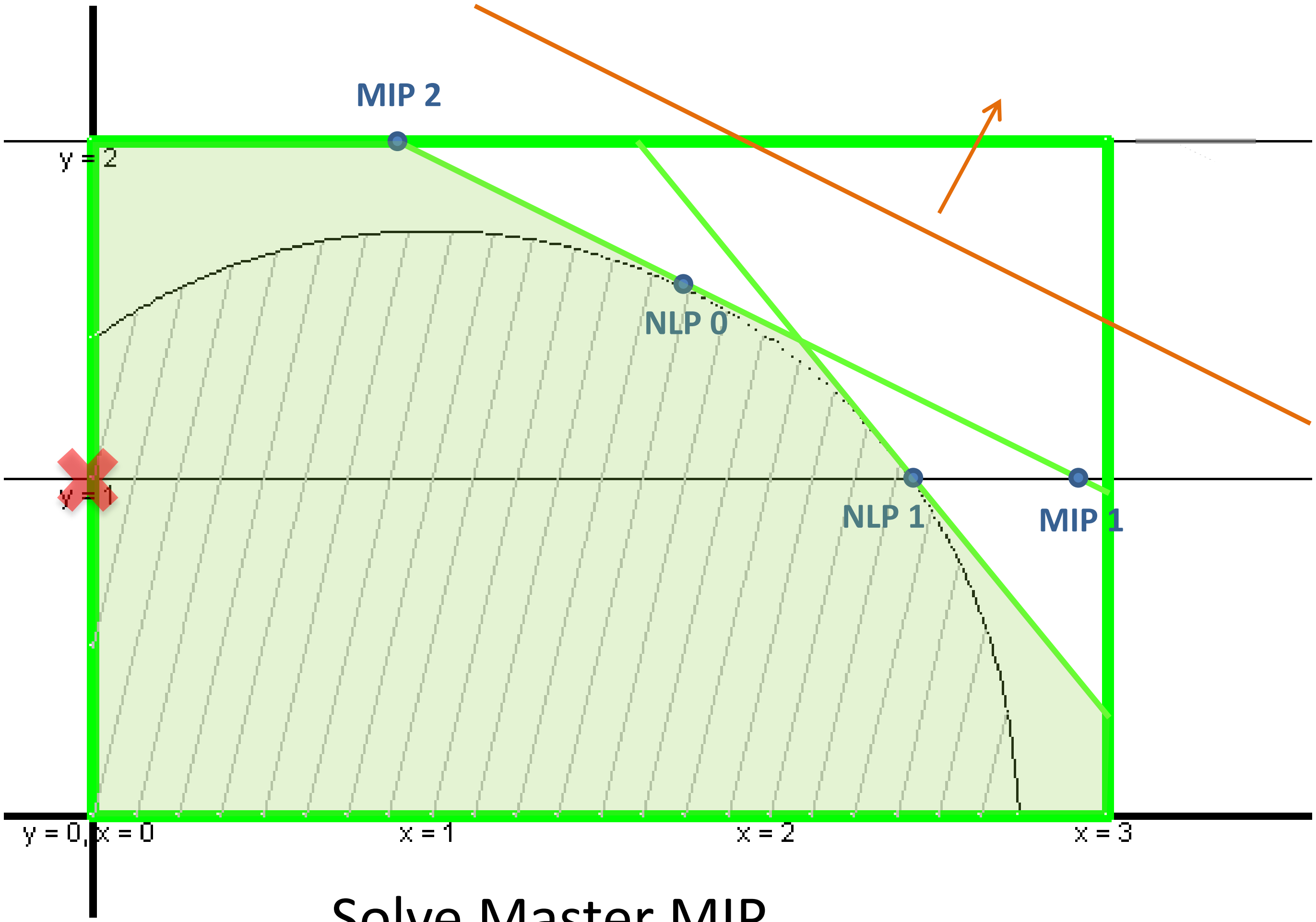
Outer Approximation: Example



Add linearization to Master MIP

AIMMS

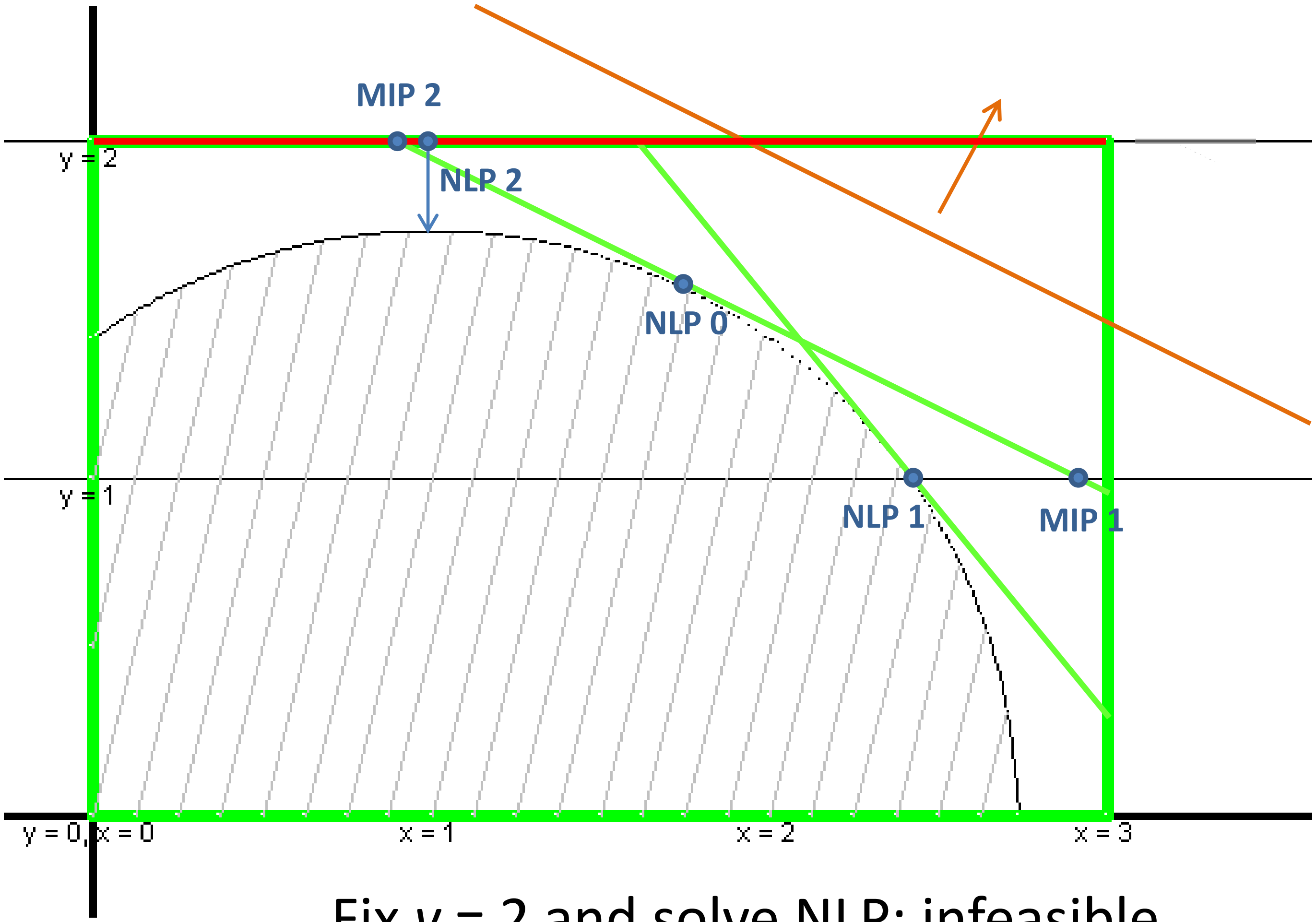
Outer Approximation: Example



Solve Master MIP

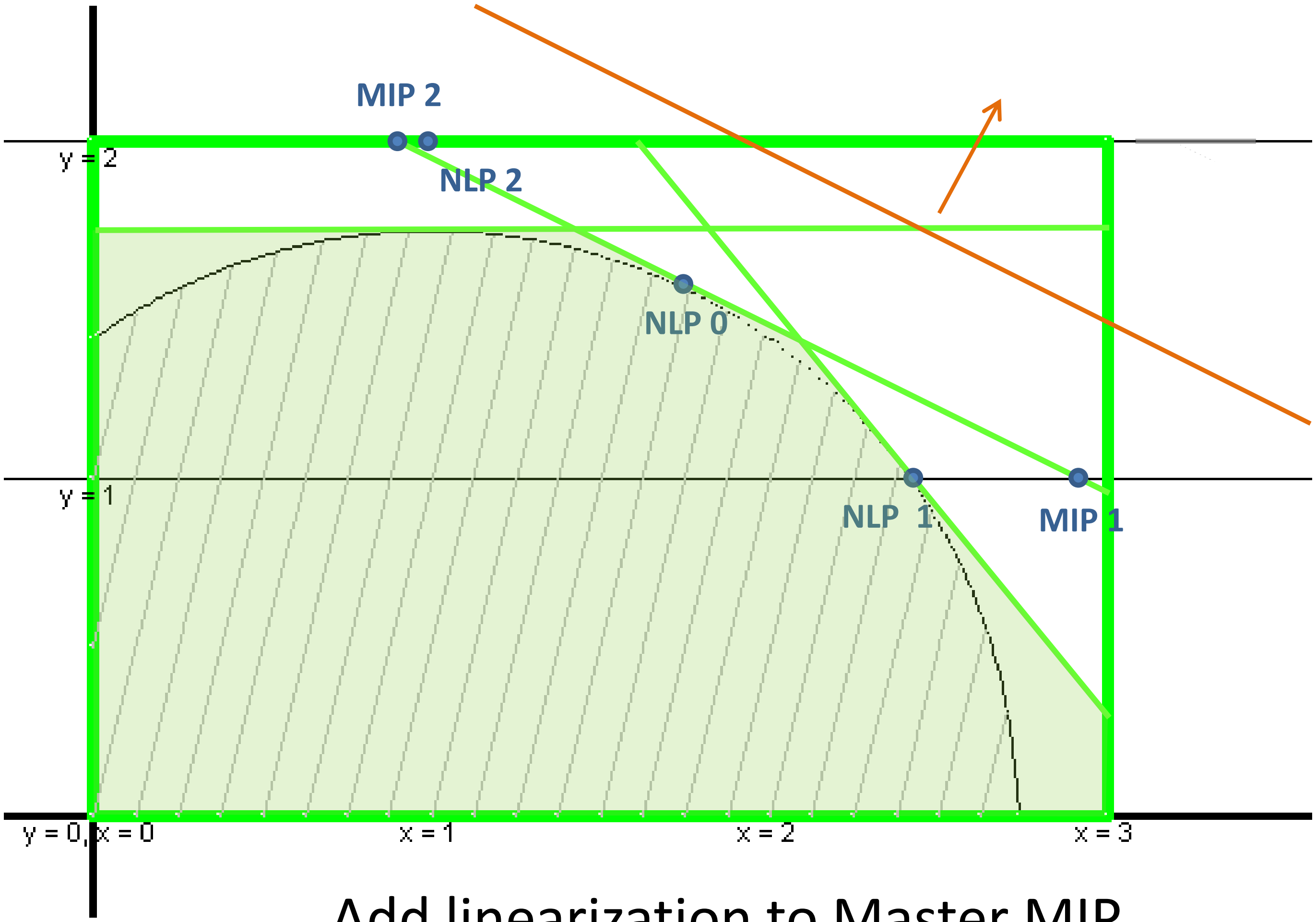
AIMMS

Outer Approximation: Example



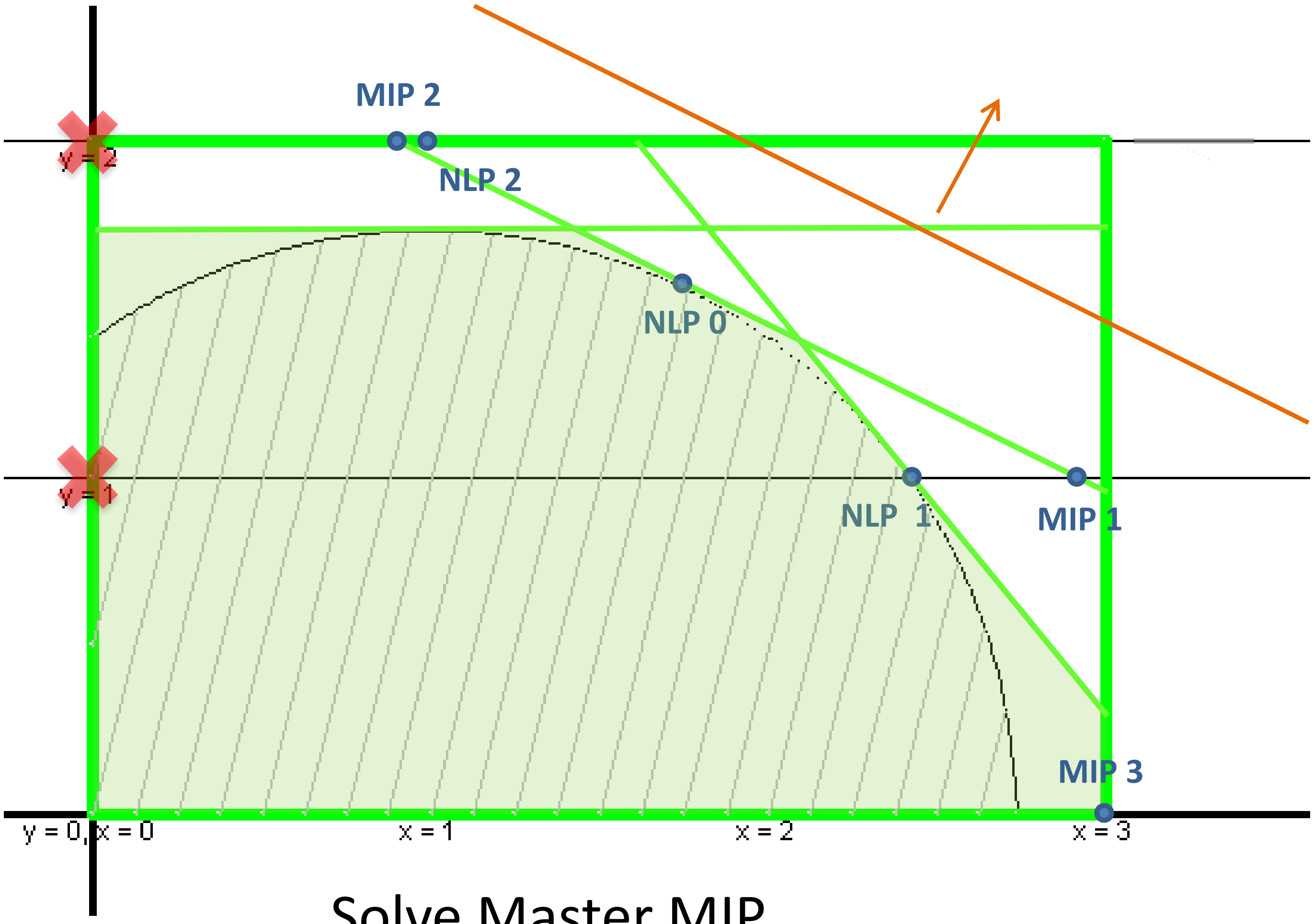
Fix *y* = 2 and solve NLP: infeasible

Outer Approximation: Example



Add linearization to Master MIP

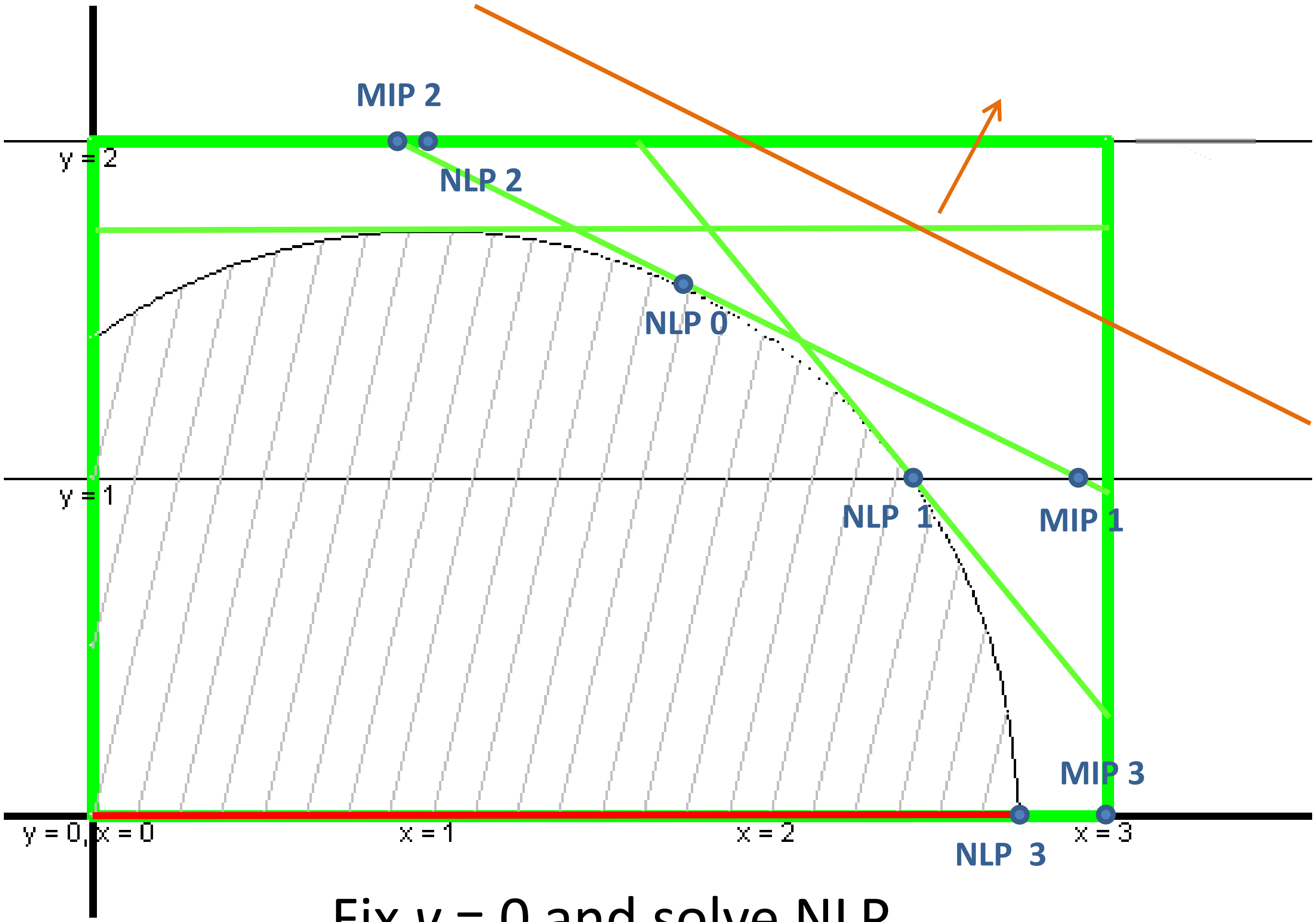
Outer Approximation: Example



Solve Master MIP

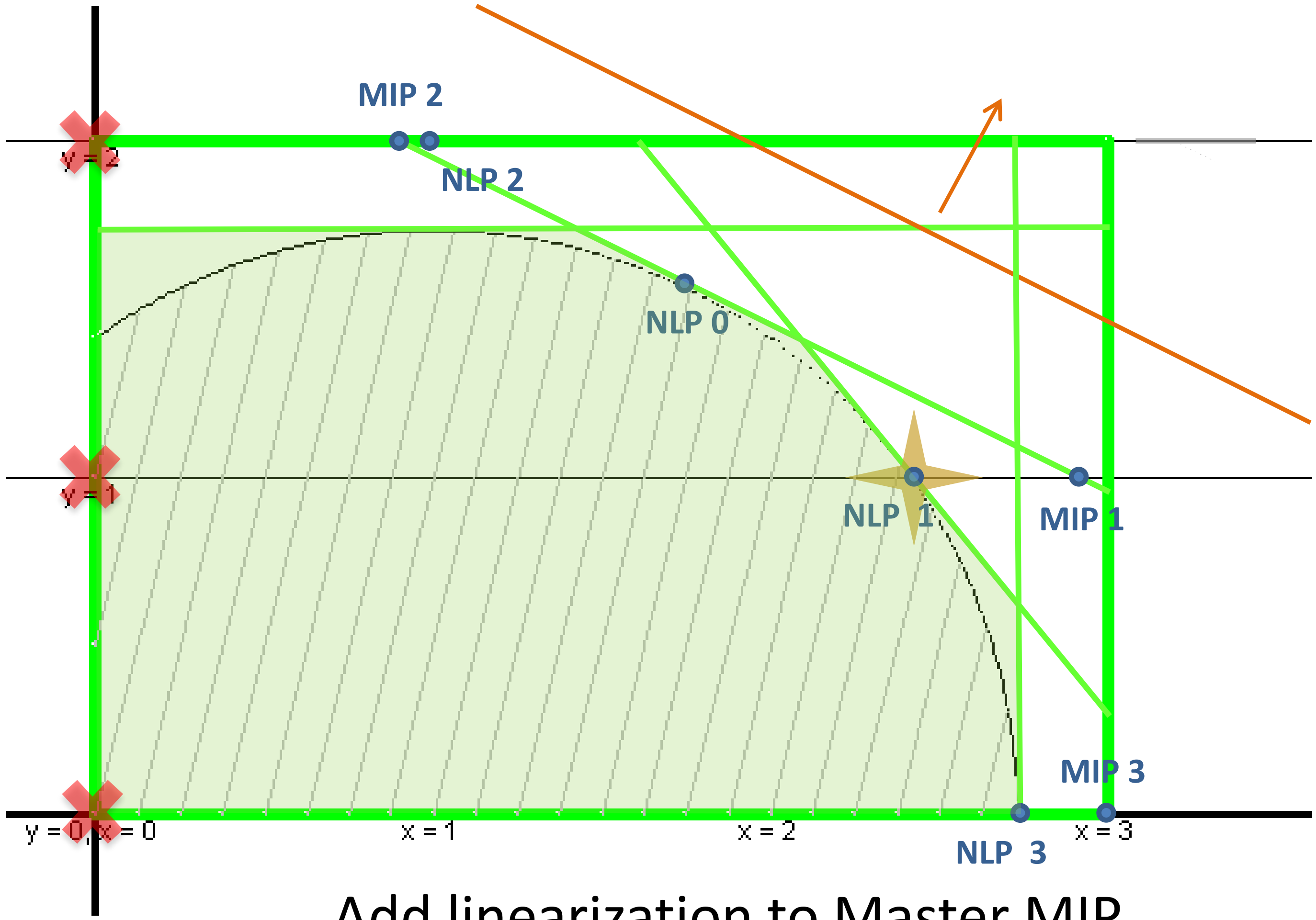
AIMMS

Outer Approximation: Example



Fix *y* = 0 and solve NLP

Outer Approximation: Example



Add linearization to Master MIP

AIMMS

AIMMS Presolver

> Delete redundant constraints & fixed variables

> Bounds Tightening

- Variable *x*: range [0,inf) ➔ range [10,55]
- Linear / nonlinear constraints

> Improve coefficients (“probing”)

> Linearize quadratic constraints

Linearize constraints – Example 1

> Initial constraint:

Y = sum[j, sqrt(p(j) * sum(i, X(i,j) * q(i))]

Here X(i,j) is a binary variable such that X(i,j) = 0 if i <> j

> Remark: You can use X(i) instead of X(i,j)

> Step 1: Y = sum[j, sqrt(p(j) * X(j,j) * q(j))]

> Step 2: Y = sum[j, X(j,j) * sqrt(p(j) * q(j))]

Linearize constraints – Example 2

> Initial constraints ($0 \le X(i) \le u$):

$\text{sum}(i, Y(i)) \ge 1$

$Z = \max(i, X(i))$

$Y(i) = \text{if } (X(i) = Z) \text{ then } 1 \text{ else } 0 \text{ endif;} \qquad \forall i$

X(i)	3	7	5
Y(i)	0	1	0

> Reformulation:

$\text{sum}(i, Y(i)) \ge 1$

$Z \ge X(i) \qquad \forall i$

$X(i) \ge Z \qquad \forall i$

Troubleshooting

> Solver returns Infeasible

- Look for errors & warnings (e.g. derivative evaluation errors)
- Check solver log/status file
- BARON: Infeasibility finder (option: Compute IIS)
- Display Infeasibility Analysis using AIMMS Presolver
- Remove nonlinear constraints; is MIP also infeasible?

  GMP::Instance::CreateMasterMIP
- Reformulate (parts of) model

Troubleshooting AOA

> AOA returns Infeasible

- First MIP is infeasible: the linear problem obtained by removing all nonlinear constraints is infeasible
- First MIP is unbounded: add finite upper and lower bounds for variables
- All NLP are infeasible: use multistart, switch NLP solver or increase iteration limit

> AOA takes a long time

- Decrease iteration limit
- Increase value of option ‘MIP Relative Optimality Tolerance’

> AOA returns poor solution

- Use multistart, or call AOA for second time (better starting point)

AIMMS

(Dis)Advantages solvers

AOA	BARON	Knitro
+ Open algorithm	+ Global solver	+ Three algorithms
+ Combine with multistart	+ Find *k* best solutions	+ Tuning tool
+ We know it very well	+ Branching priorities	+ Branching priorities
− Algorithm not solver	+ Infeasibility finder	+ Several NLP algorithms
	− No goniometric func.	− Derivative errors
	− Not in Free Academic License	− Not in Free Academic License

option: honor bounds

AIMMS

References

> AIMMS Language Reference

- The AIMMS OA algorithm for MINLP: Chapter 18
- AIMMS Presolver & Multistart: Chapter 17

> AIMMS Optimization Modeling

- Integer Linear Programming Tricks: Chapter 7

> White papers on website

- The AIMMS Outer Approximation Algorithm for MINLP (using GMP)
- Solving convex MINLP problems with AIMMS

AIMMS

References

> Webinars

- http://aimms.com/english/developers/resources/webinars/
- AIMMS Presolver
- Multistart
- Introduction to GMP

> Example: Water Distribution

- http://aimms.com/english/developers/resources/examples/
- Generic Branch-and-Bound algorithm (GMP)