Order-Based Error for Managing Ensembles of Surrogates in Derivative-Free Optimization

Sébastien Le Digabel
Bastien Talgorn
Charles Audet
Michael Kokkolaras

GERAD, École Polytechnique de Montréal, McGill University.

SIOPT, Vancouver

2017–05–24

Presentation outline

Derivative-Free Optimization

The MADS algorithm

Surrogate-assisted optimization

Order error and ensembles of surrogates

Computational results

Discussion

Derivative-Free Optimization (DFO) problems

- Optimization problem:

$$\min_{x \in \Omega} \quad f(x)$$

- $\Omega = \left\{ x \in \mathcal{X} : c_j(x) \leq 0, j \in \{1, 2, \ldots, m\} \right\} \subseteq \mathbb{R}^n$.
- $\mathcal{X}$: Bounds and/or nonquantifiable constraints.

Blackbox and derivative-free optimization

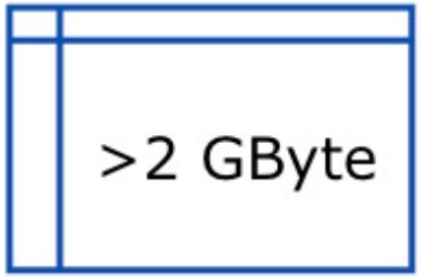
From [Audet and Kokkolaras, 2016]:

- "Derivative-Free Optimization *refers to the use of algorithms that use only function values because their derivatives are either not defined or not available.*
  Gradient approximations may sometimes be obtained, but the amount of work required to ensure they are dependable may not be worth the effort."

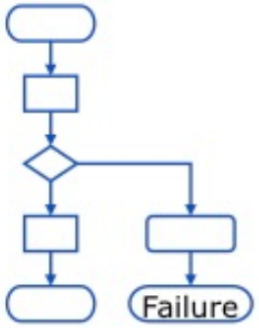
- "Blackbox Optimization *refers to problems where the structure of the objective and constraint functions cannot be exploited.*
  Often the case when their evaluation requires the execution of a (usually time-consuming) simulation using computational models, typically inaccessible by the user."

Blackboxes as illustrated by J. Simonis [ISMP 2009]

Long runtime

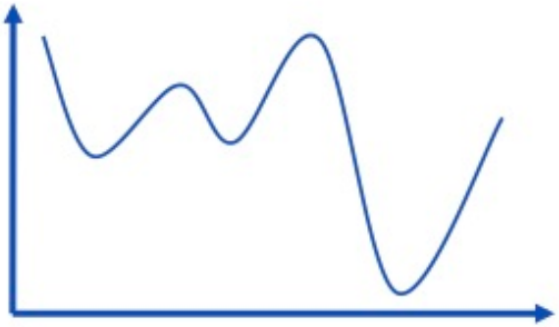


Large memory requirement



Software might fail

No derivatives available



Local optima

Non-smooth, noisy

Derivative-Free Optimization

The MADS algorithm

Surrogate-assisted optimization

Order error and ensembles of surrogates

Computational results

Discussion

Mesh Adaptive Direct Search (MADS)

- [Audet and Dennis, Jr., 2006].

Mesh Adaptive Direct Search (MADS)

- [Audet and Dennis, Jr., 2006].
- Iterative algorithm that evaluates the blackbox at some trial points on a spatial discretization called the mesh.
- One iteration = search and poll.

Mesh Adaptive Direct Search (MADS)

- [Audet and Dennis, Jr., 2006].
- Iterative algorithm that evaluates the blackbox at some trial points on a spatial discretization called the mesh.
- One iteration = search and poll.
- The search allows trial points generated anywhere on the mesh.
- The poll consists in generating a list of trial points constructed from poll directions. These directions grow dense.

Mesh Adaptive Direct Search (MADS)

- [Audet and Dennis, Jr., 2006].
- Iterative algorithm that evaluates the blackbox at some trial points on a spatial discretization called the mesh.
- One iteration = search and poll.
- The search allows trial points generated anywhere on the mesh.
- The poll consists in generating a list of trial points constructed from poll directions. These directions grow dense.
- At the end of the iteration, the mesh size is reduced if no new success point is found.

[0] Initializations (x_0, Δ^0: initial poll size)

[1] Iteration k

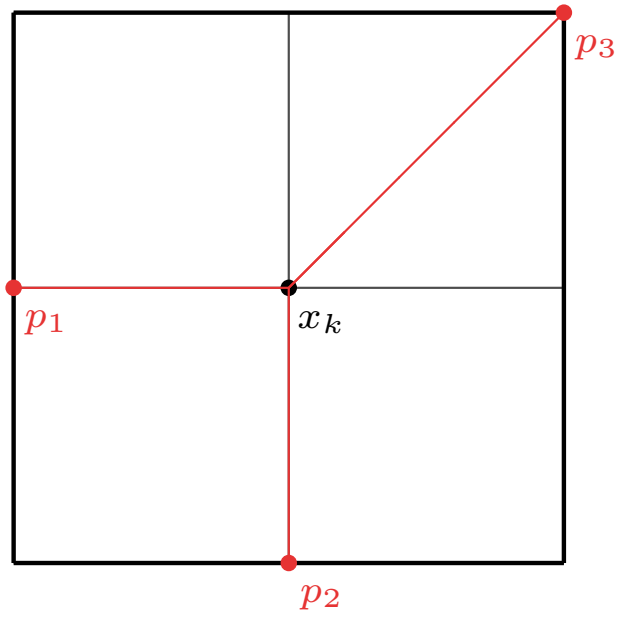
- let $\delta^k \leq \Delta^k$ be the mesh size parameter
- **Search**
  - test a finite number of mesh points
- **Poll** (if the Search failed)
  - construct set of directions D_k
  - test poll set $P_k = \{x_k + \delta^k d : d \in D_k\}$ with $\|\delta^k d\| \simeq \Delta^k$

[2] Updates

- if success
  - $x_{k+1} \leftarrow$ success point
  - increase Δ^k
- else
  - $x_{k+1} \leftarrow x_k$
  - decrease Δ^k
- $k \leftarrow k+1$, stop if $\Delta^k \leq \Delta_{\min}$ or go to **[1]**

Poll illustration (successive fails and mesh shrinks)

$$\delta^k = 1$$
$$\Delta^k = 1$$

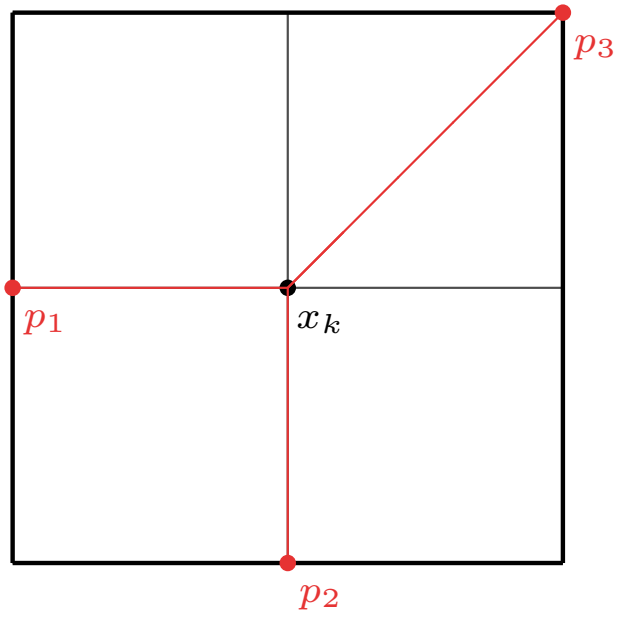


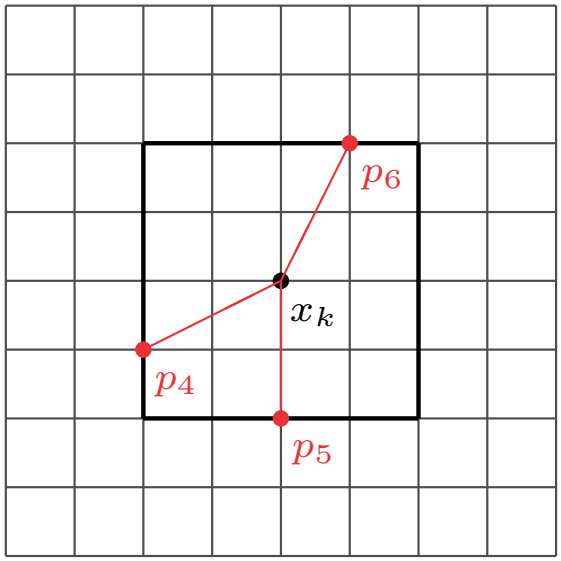
trial points$=\{p_1, p_2, p_3\}$

Poll illustration (successive fails and mesh shrinks)

$$\delta^k = 1 \qquad \Delta^k = 1$$

$$\delta^{k+1} = 1/4 \qquad \Delta^{k+1} = 1/2$$





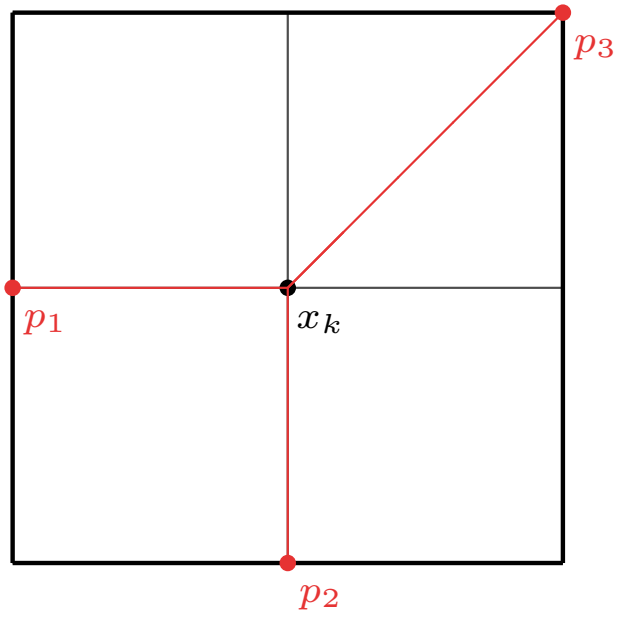
trial points$=\{p_1, p_2, p_3\}$ $\qquad = \{p_4, p_5, p_6\}$

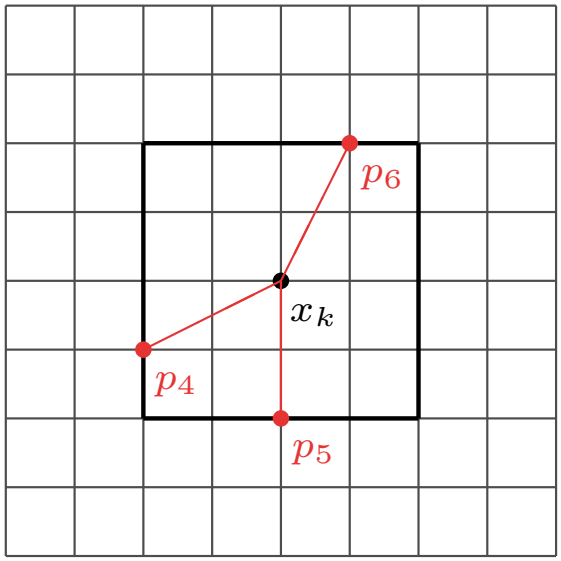
Poll illustration (successive fails and mesh shrinks)

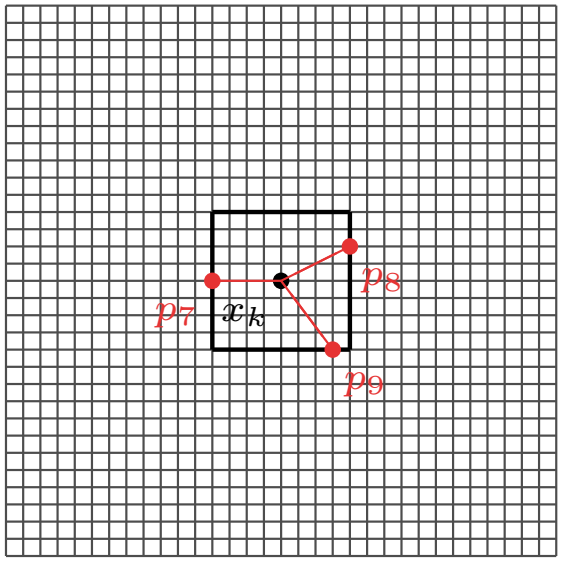
$\delta^k = 1$
$\Delta^k = 1$

$\delta^{k+1} = 1/4$
$\Delta^{k+1} = 1/2$

$\delta^{k+2} = 1/16$
$\Delta^{k+2} = 1/4$







trial points$=\{p_1, p_2, p_3\}$ $= \{p_4, p_5, p_6\}$ $= \{p_7, p_8, p_9\}$

Convergence results

- MADS is backed by a convergence analysis based on the calculus for nonsmooth functions [Clarke, 1983].
- It produces solutions satisfying optimality conditions "proportional" to the smoothness of the problem.
- Summary of the results:

	Unconstrained	Constrained
Smooth	$\nabla f(x) = 0$	$f'(x;d) \geq 0$ for all $d \in T_\Omega(x)$
Nonsmooth	$0 \in \partial f(x)$	$f^\circ(x;d) \geq 0$ for all $d \in T_\Omega^H(x)$

Derivative-Free Optimization

The MADS algorithm

Surrogate-assisted optimization

Order error and ensembles of surrogates

Computational results

Discussion

Static versus dynamic surrogates

- Static surrogate: A cheaper model defined a priori by the user. It is used as a blackbox. Typically a simplified physics model. Variable precision is not yet considered.
- Dynamic surrogate: Model managed by the algorithm, based on past evaluations. It can be periodically updated.

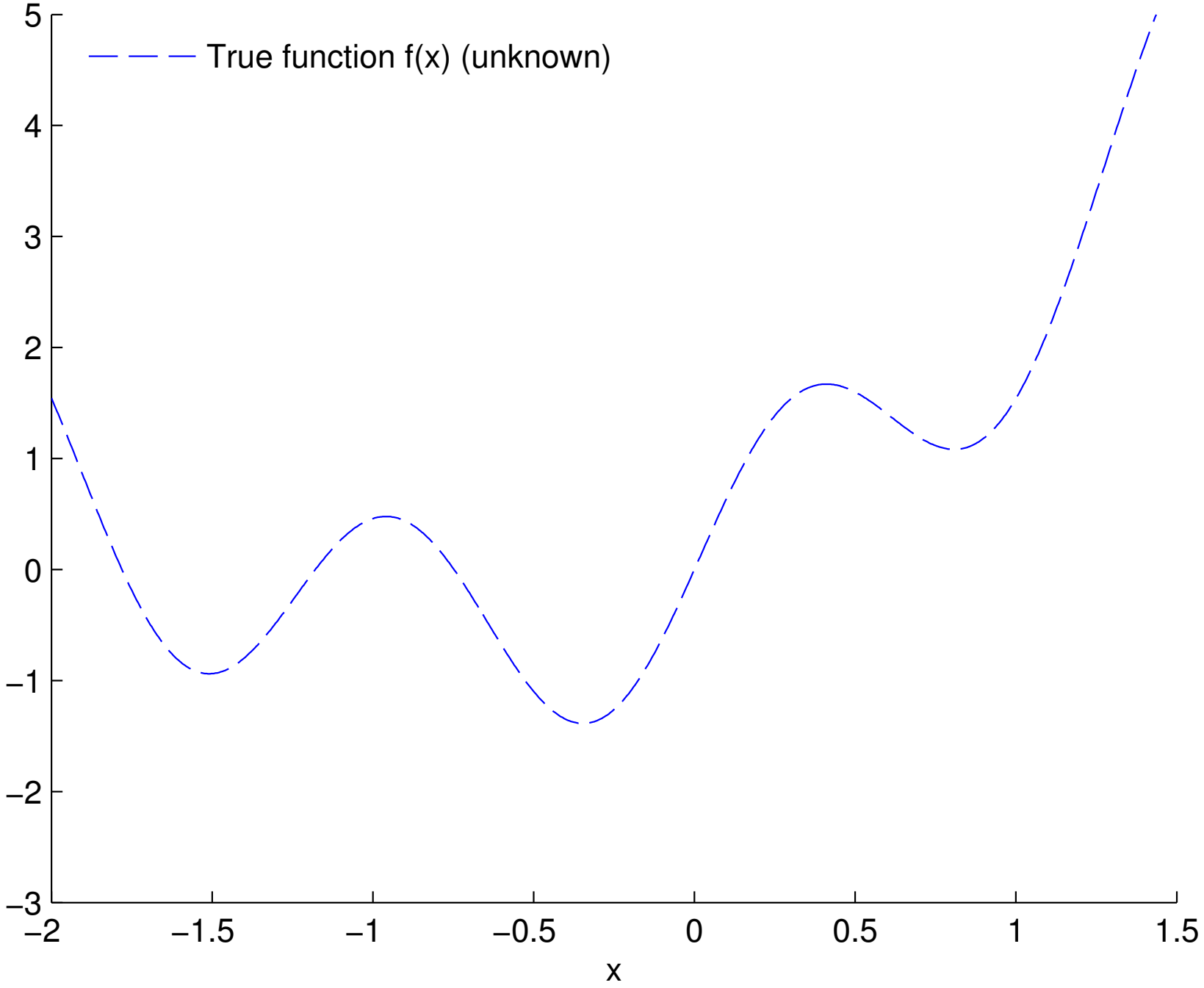
In the remaining of this presentation, we focus on dynamic surrogates.

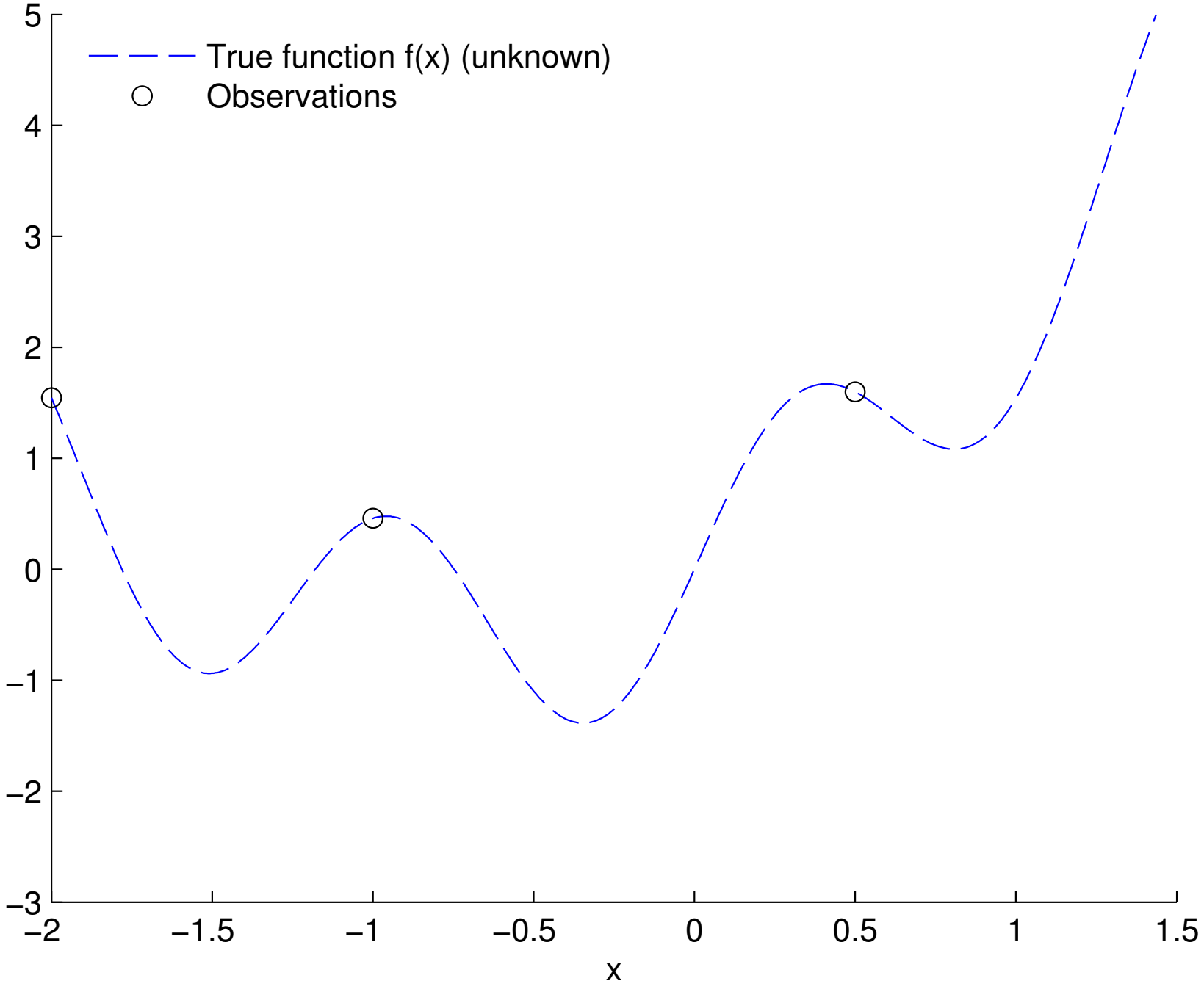
Surrogate-assisted optimization

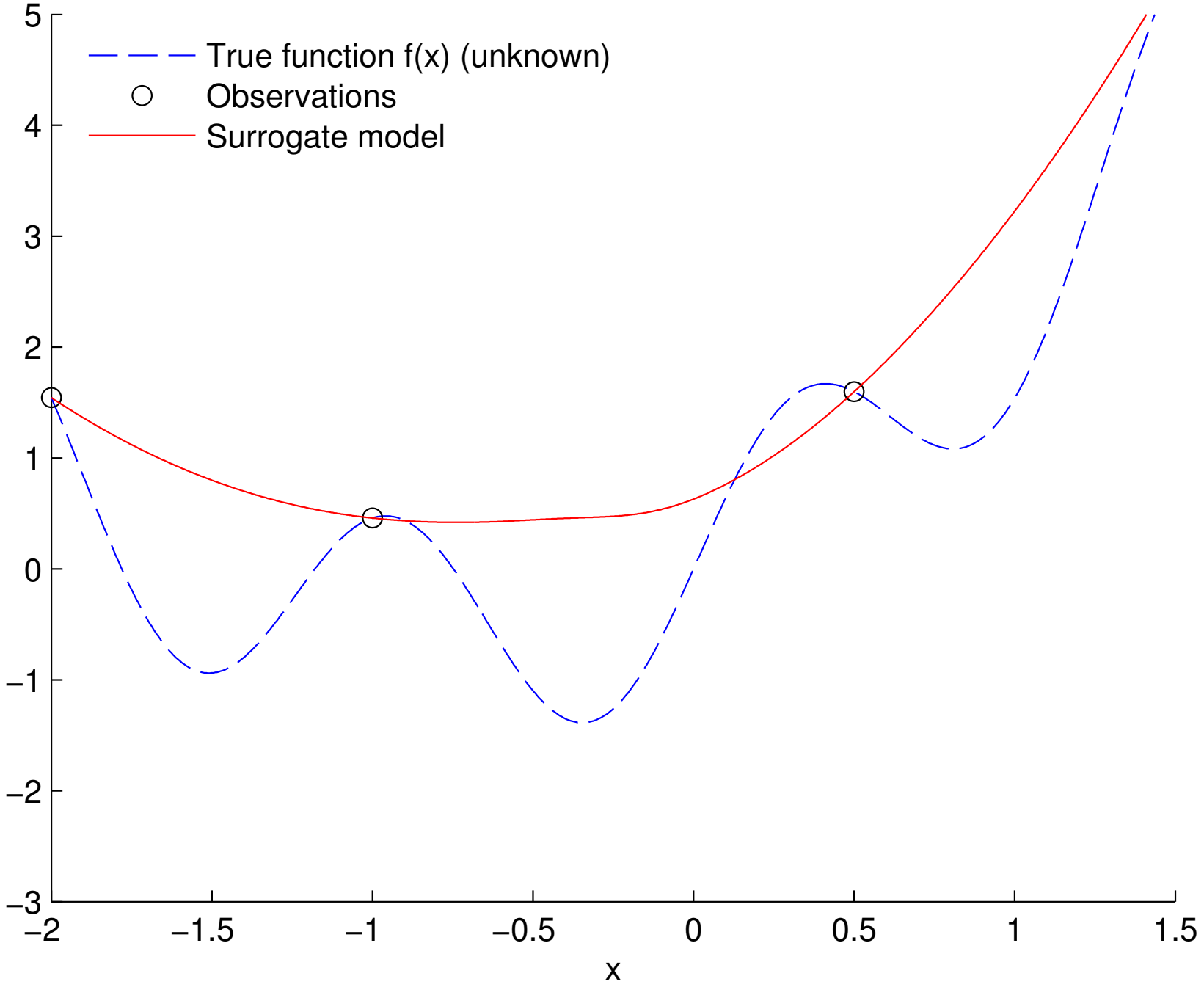
1. Use $[\mathbf{X}, f(\mathbf{X})]$ to build a surrogate $\hat{f}$ of the function f.
2. Find $\mathbf{x}_S \in \underset{\mathbf{x}}{\operatorname{argmin}}\ \hat{f}(\mathbf{x})$.
3. Evaluate $f(\mathbf{x}_S)$.
4. $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{x}_S$.
5. Go back to Step **1.**

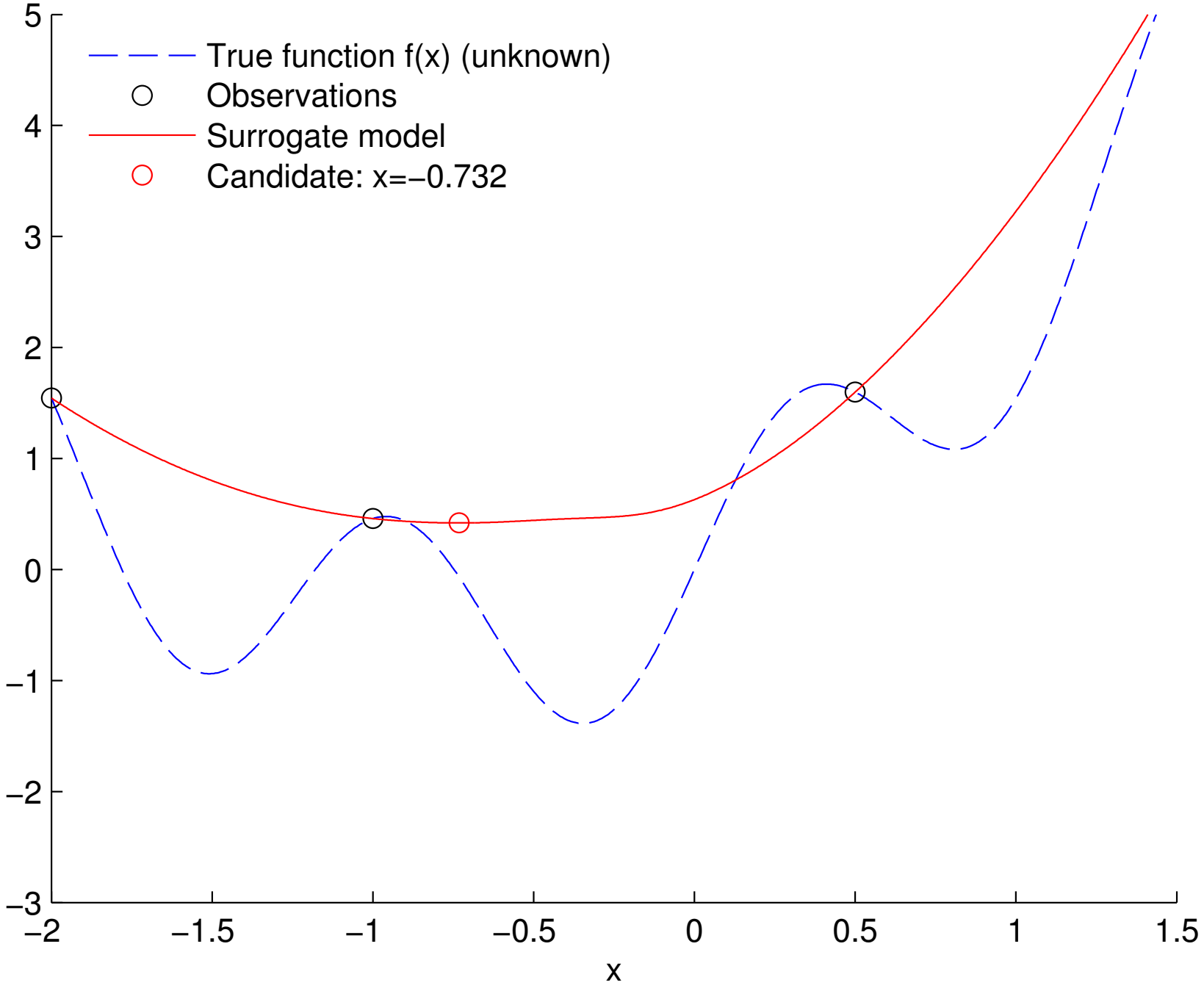
For constrained problems the same method can be used for constrained problems:

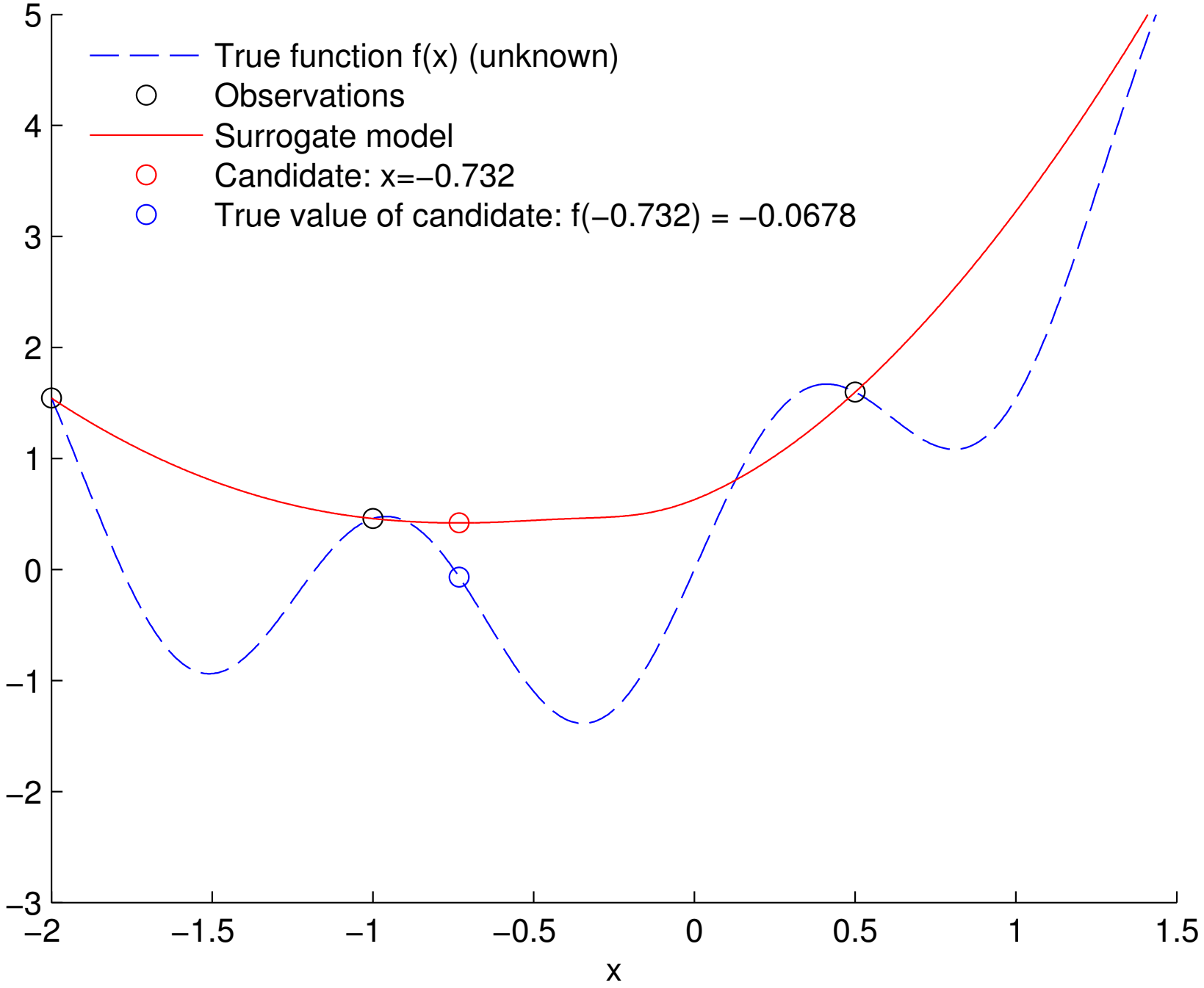
- Build the models of the constraints.
- $\mathbf{x}_S \leftarrow$ minimizer of $\hat{f}$ subject to the constraints $\hat{c}_j \leq 0$, $j = 1, 2, \ldots, m$.

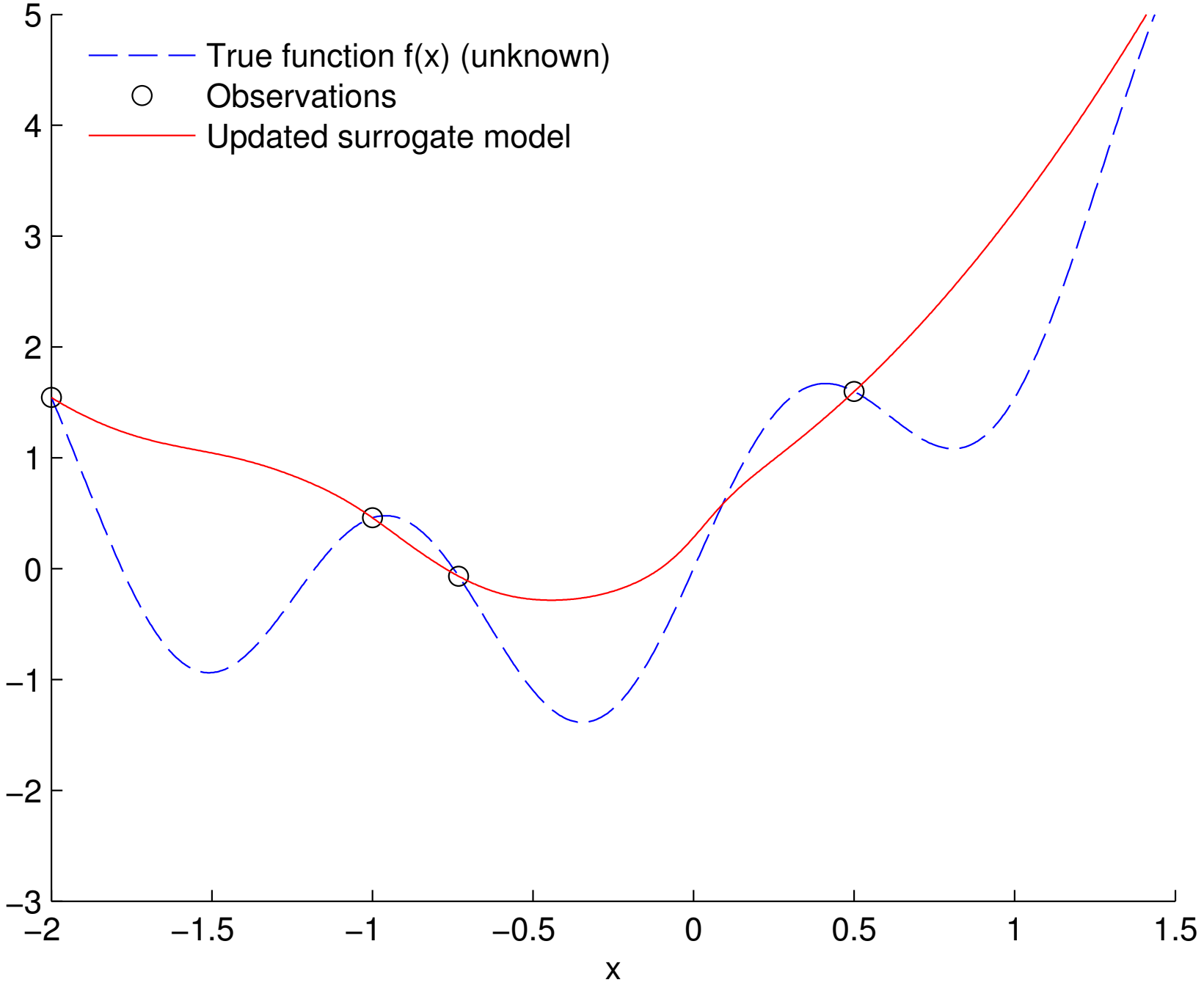

True function f(x) (unknown)
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

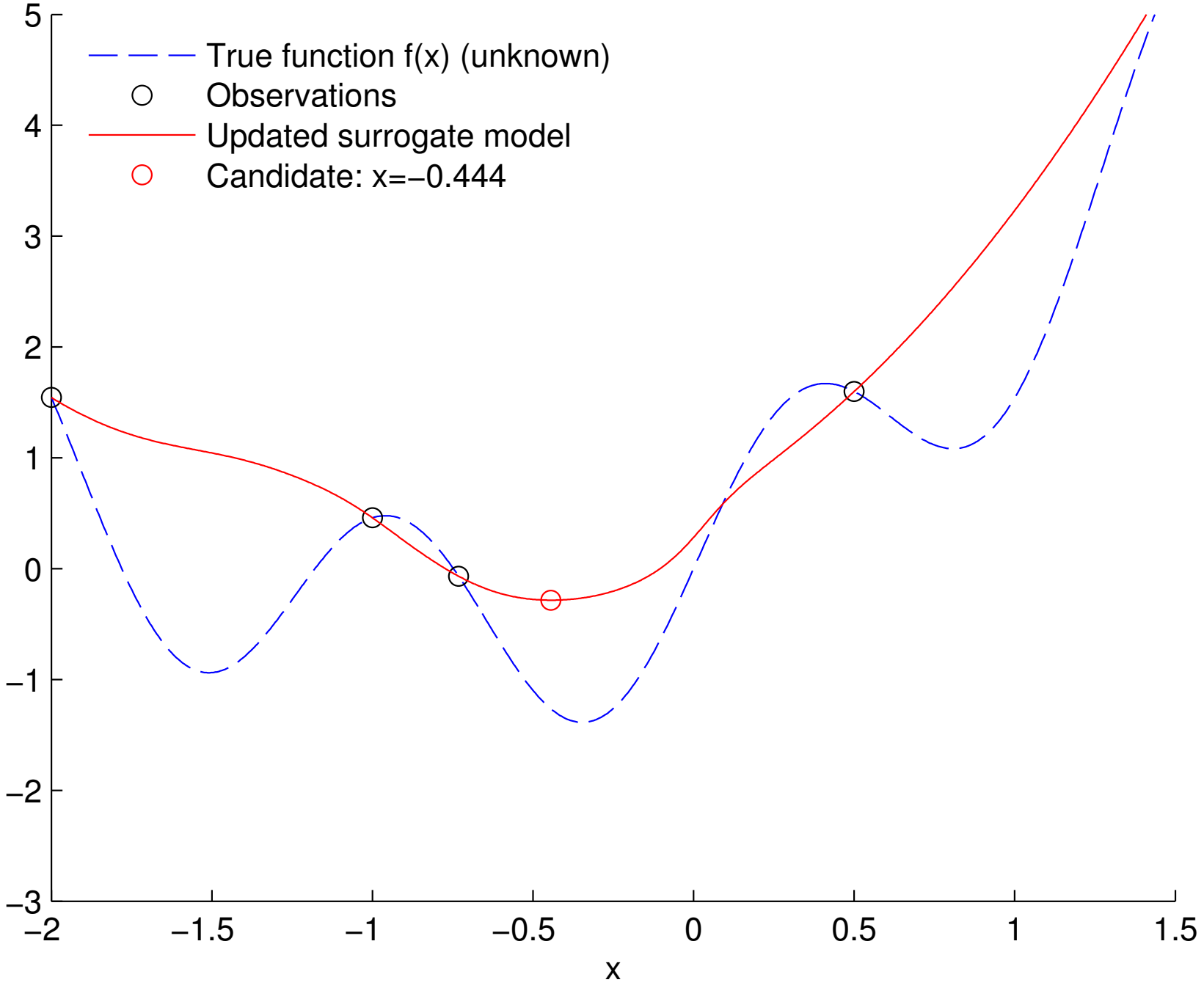

True function f(x) (unknown)
Observations
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

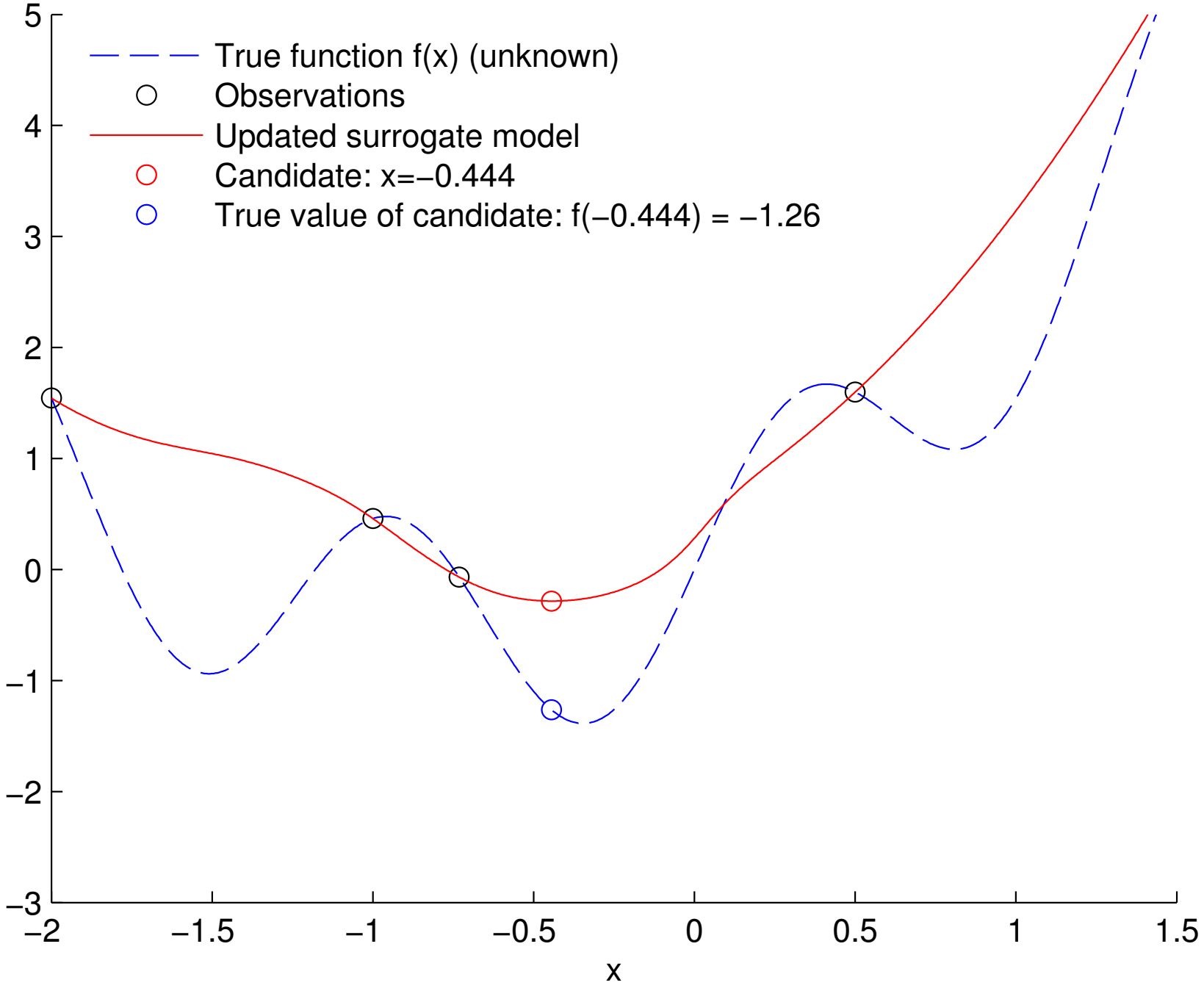

True function f(x) (unknown)
Observations
Surrogate model
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

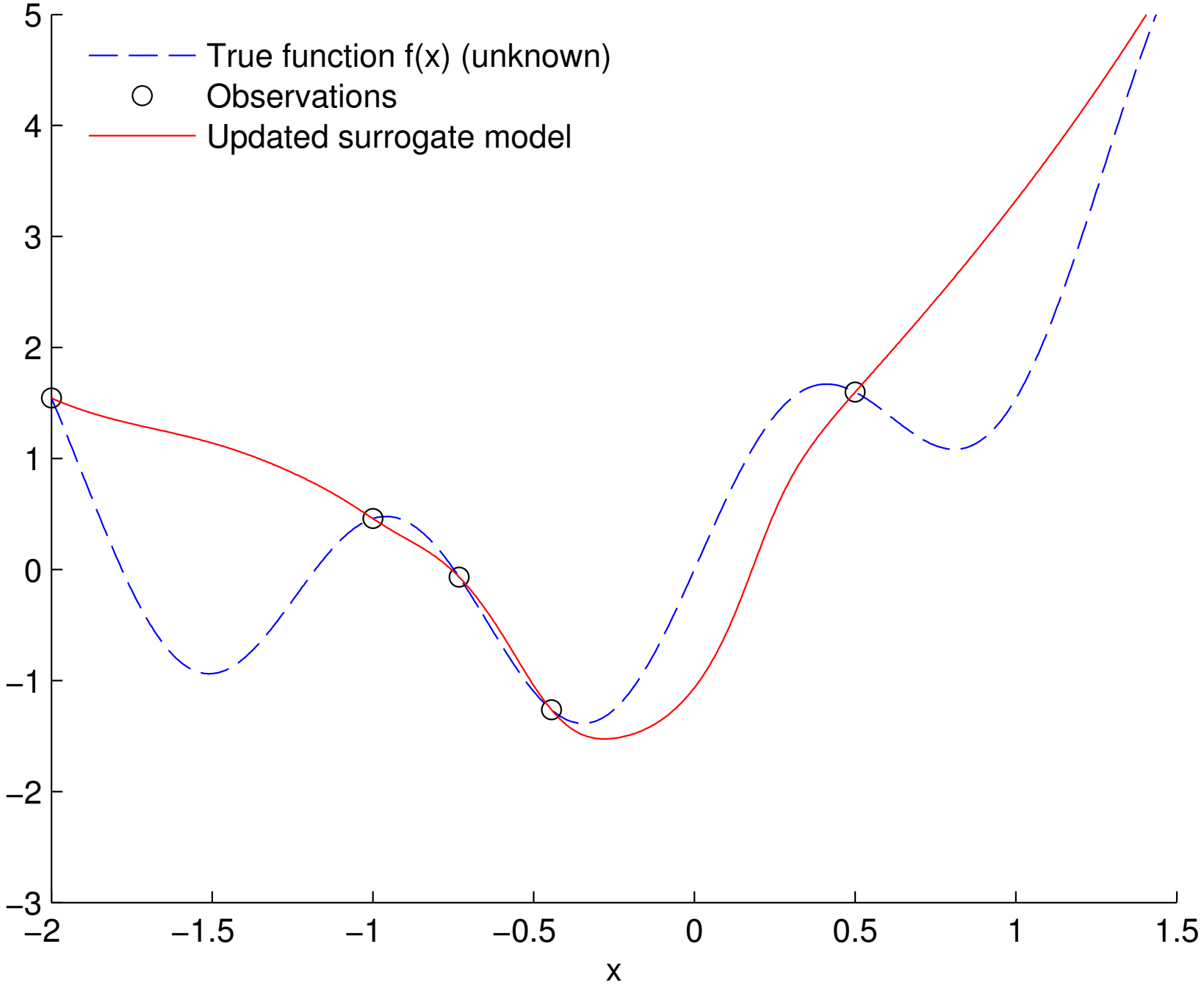

True function f(x) (unknown)
Observations
Surrogate model
Candidate: x=−0.732
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

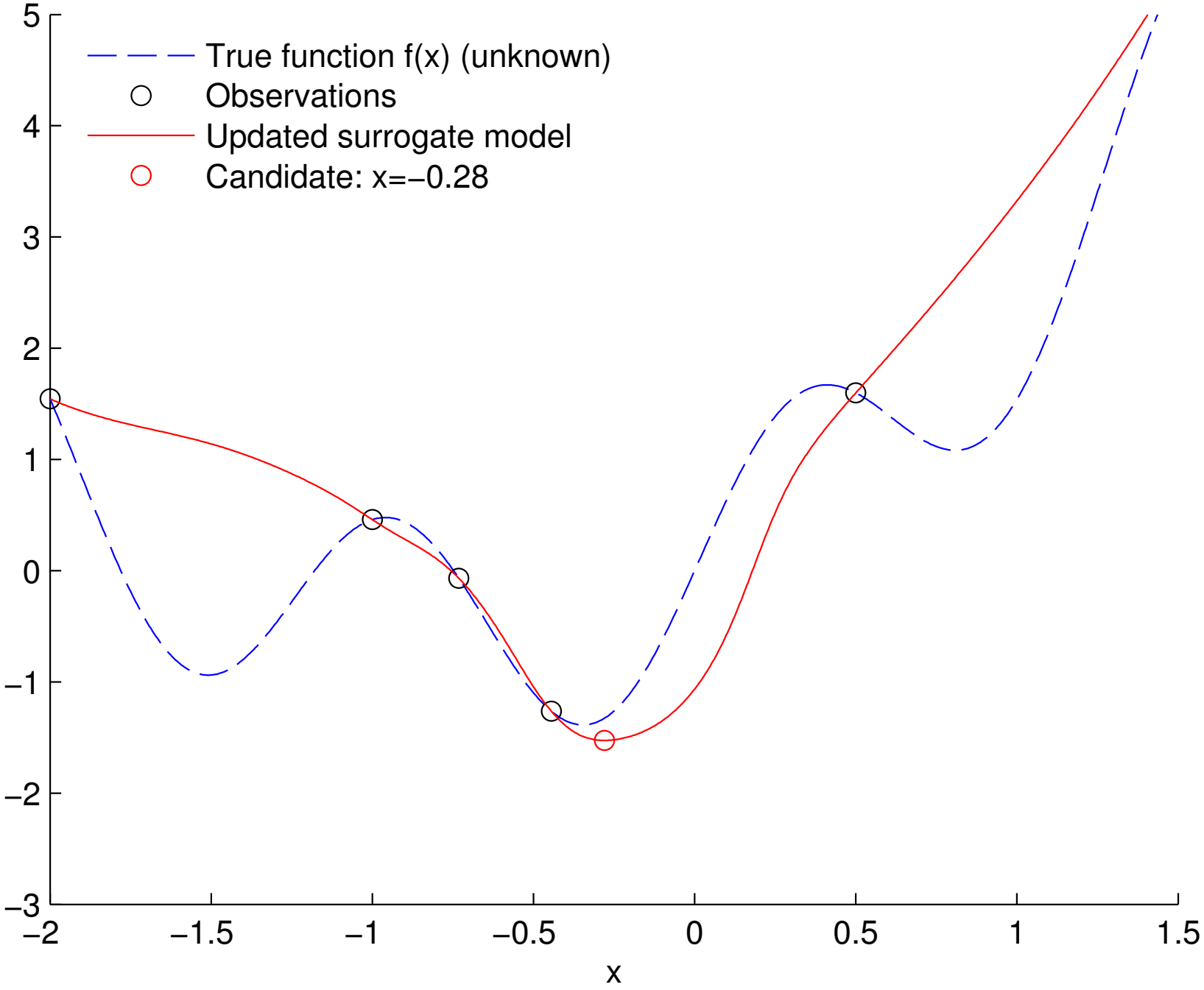

True function f(x) (unknown)
Observations
Surrogate model
Candidate: x=−0.732
True value of candidate: f(−0.732) = −0.0678
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

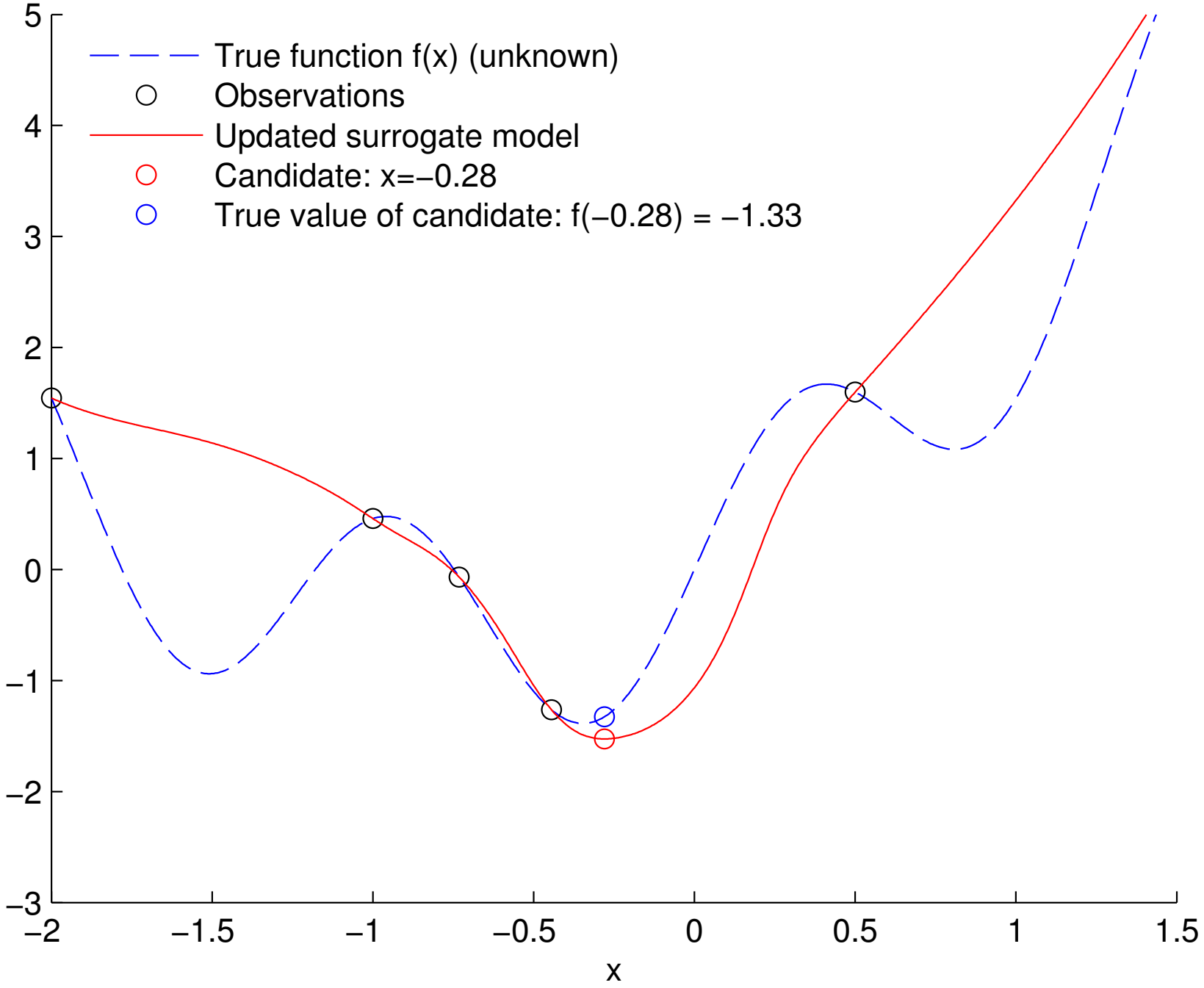

True function f(x) (unknown)
Observations
Updated surrogate model
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

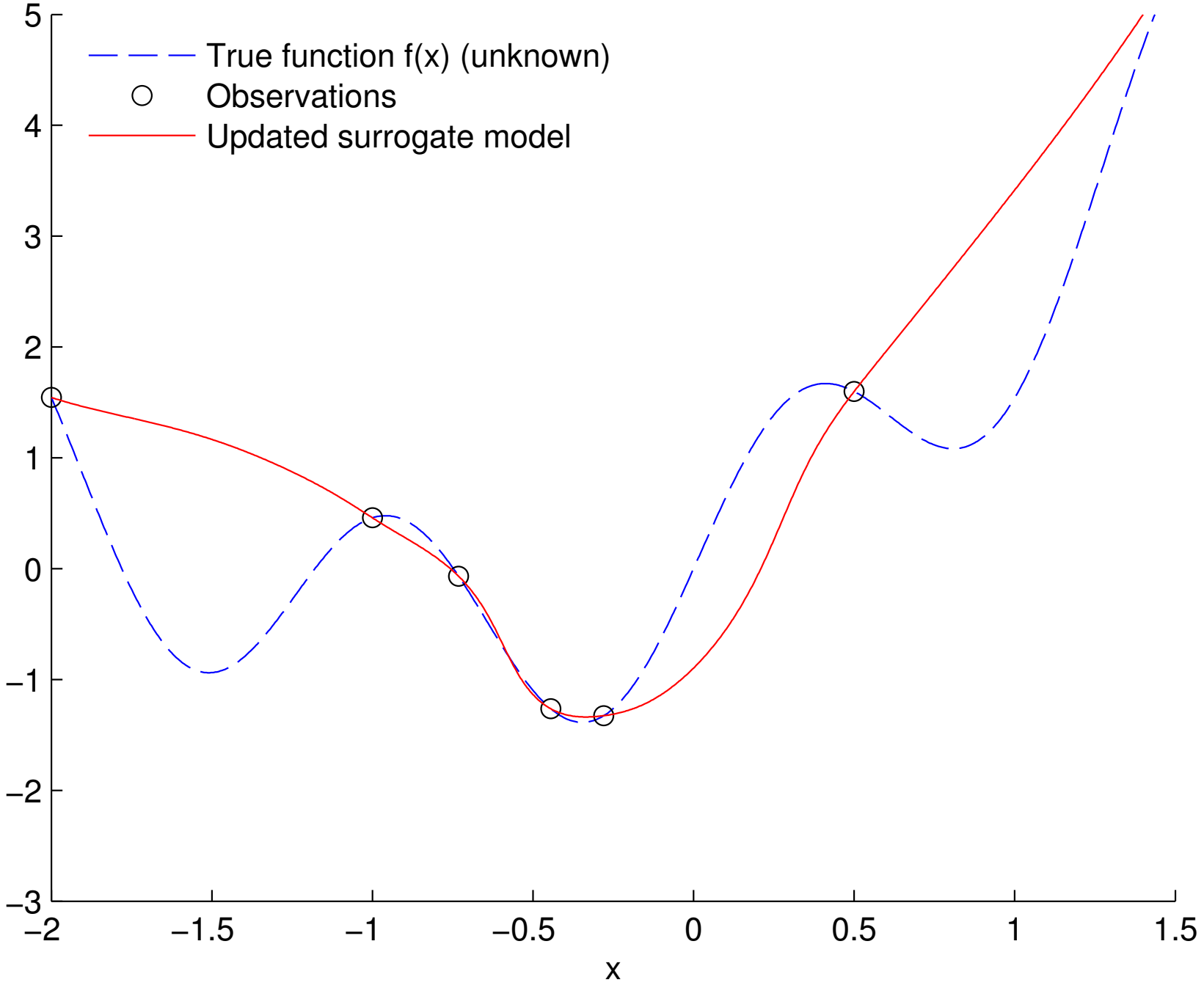

True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.444
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

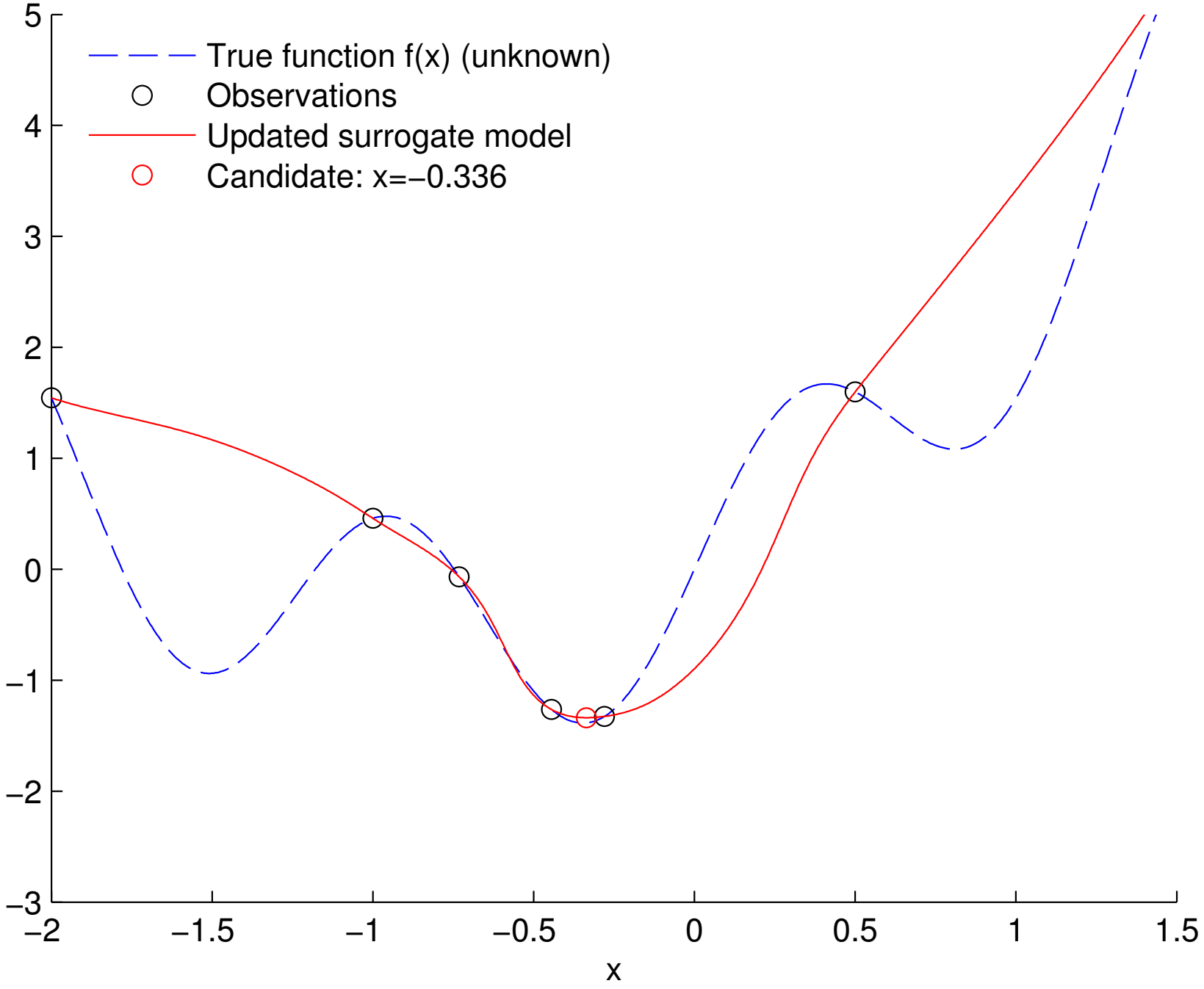

True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.444
True value of candidate: f(−0.444) = −1.26
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

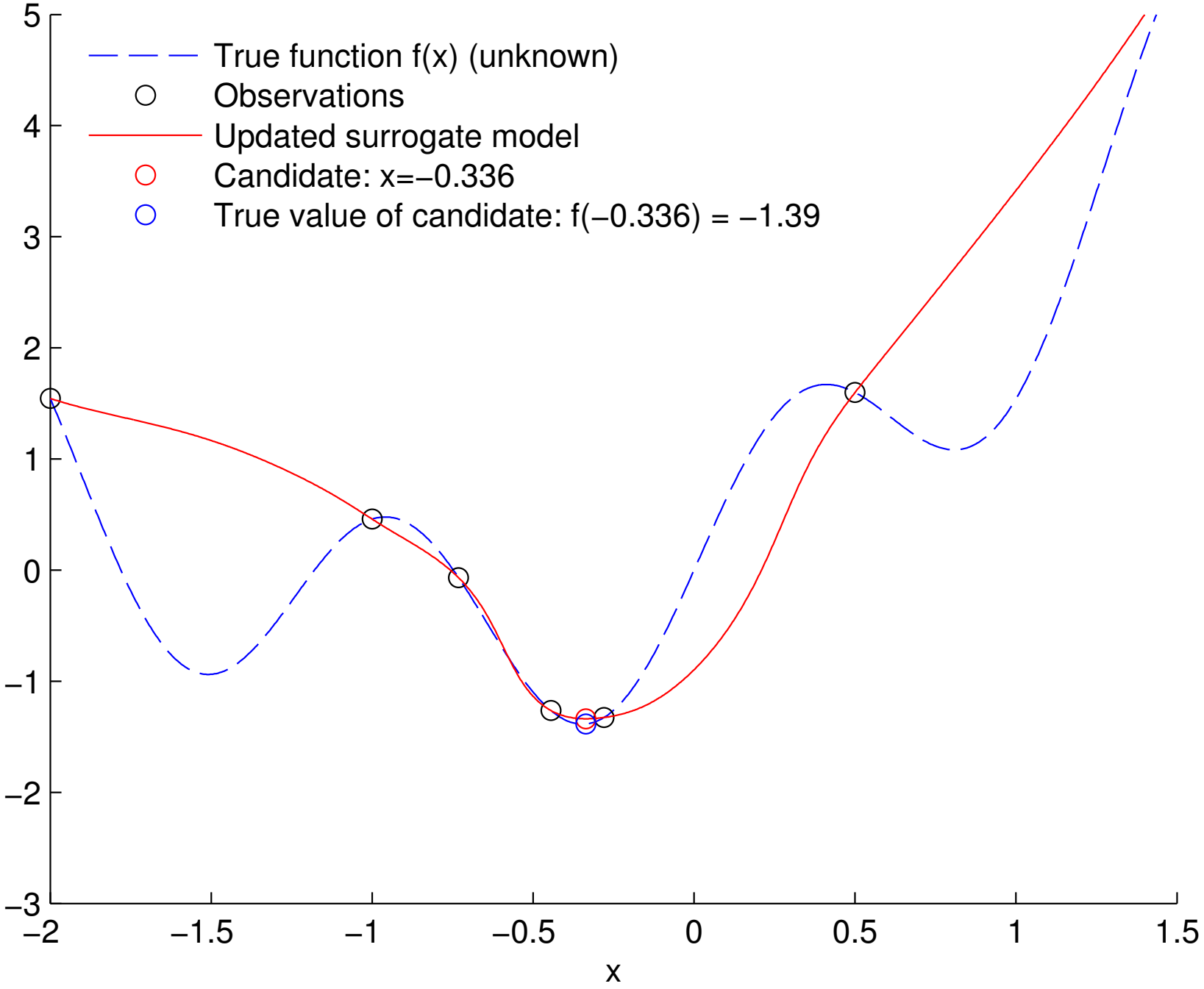

True function f(x) (unknown)
Observations
Updated surrogate model
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

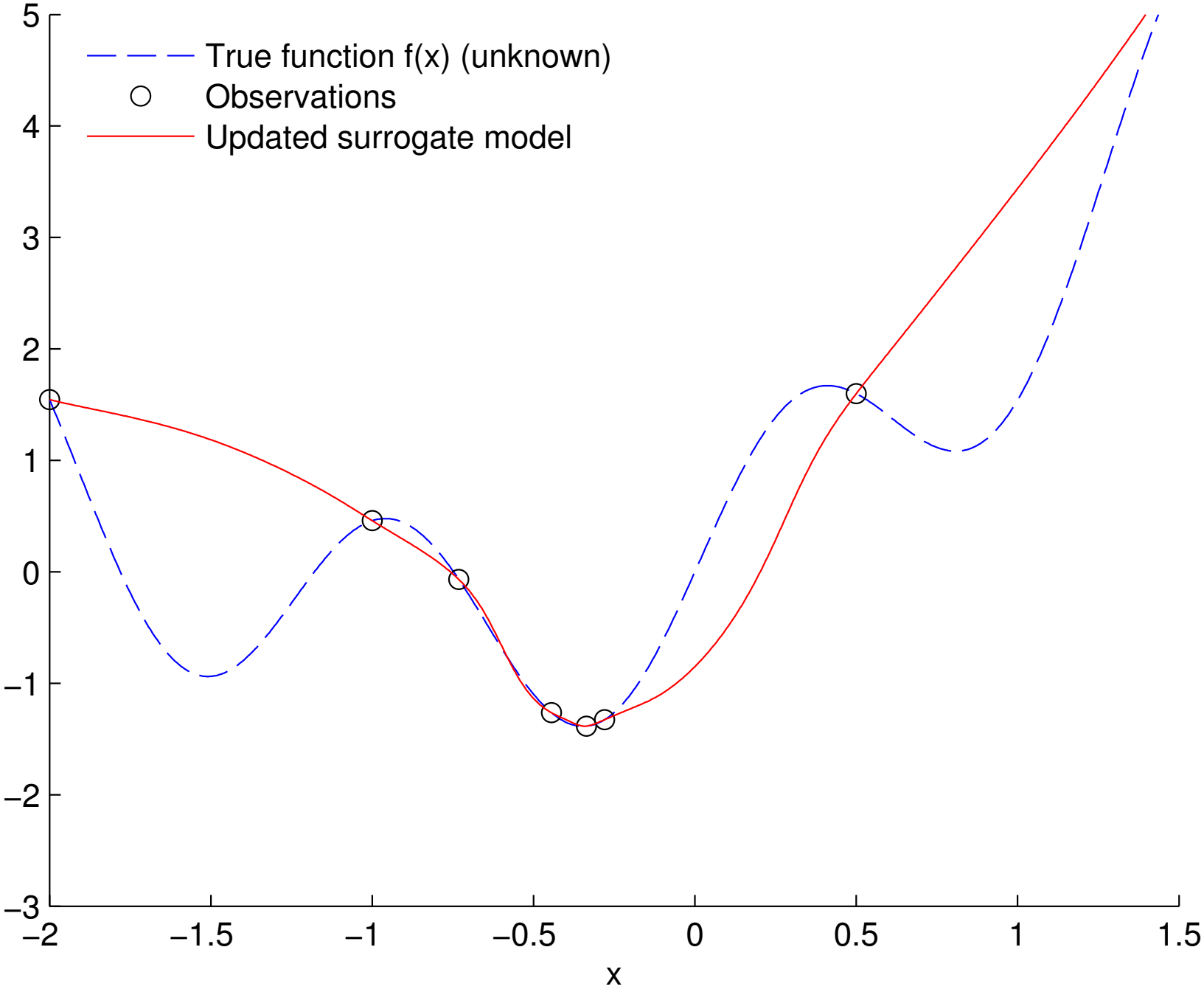

True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.28
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

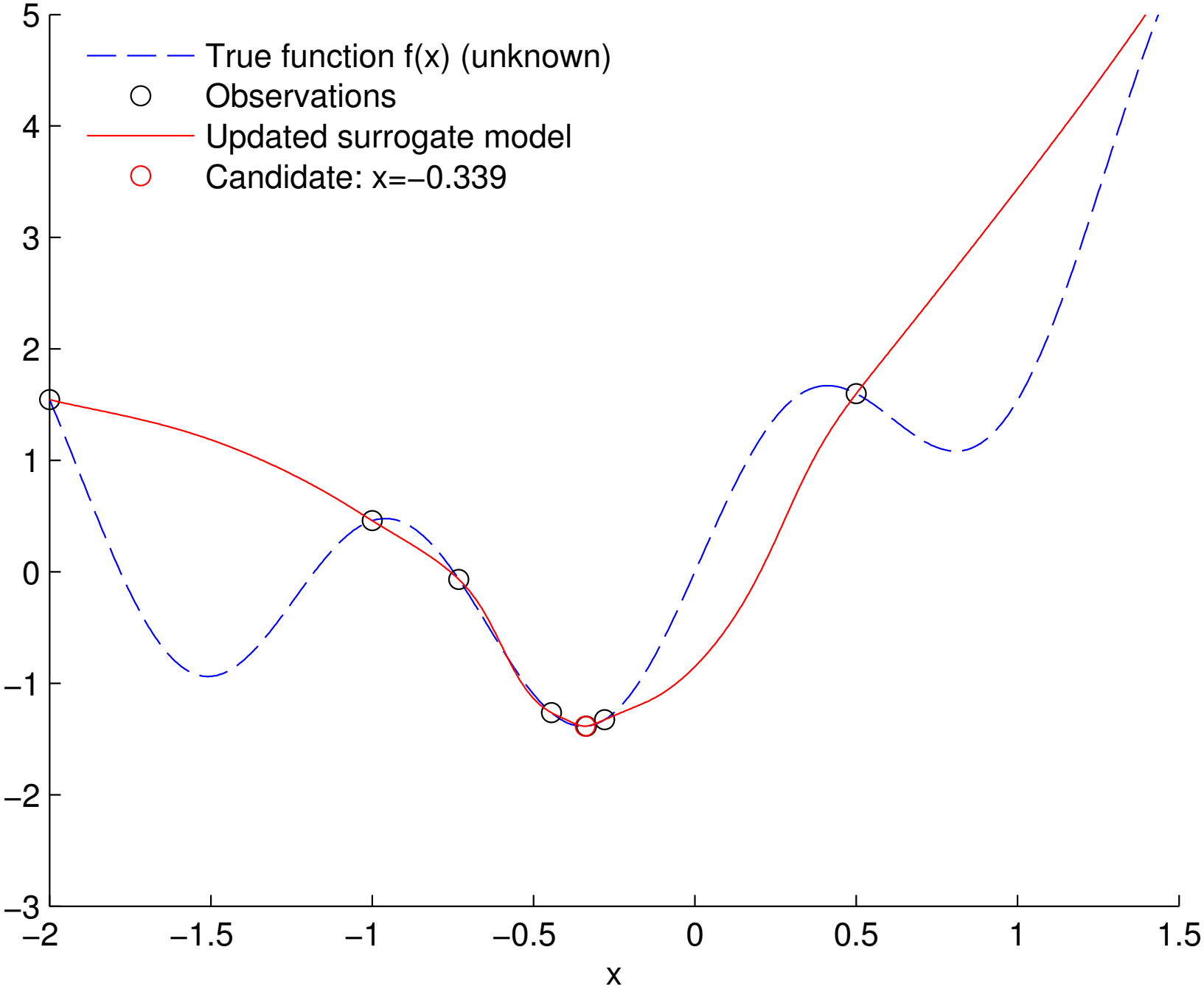

True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.28
True value of candidate: f(−0.28) = −1.33
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x


True function f(x) (unknown)
Observations
Updated surrogate model
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x


True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.336
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x


True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.336
True value of candidate: f(−0.336) = −1.39
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x


True function f(x) (unknown)
Observations
Updated surrogate model
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x


True function f(x) (unknown)
Observations
Updated surrogate model
Candidate: x=−0.339
5
4
3
2
1
0
−1
−2
−3
−2
−1.5
−1
−0.5
0
0.5
1
1.5
x

Surrogate-assisted optimization in MADS

1. Initialization:
   - Initial design (x_0).
   - Initial mesh and poll sizes (δ^0, Δ^0).
2. Search
   - Build the **surrogates** $\hat{f}$ and $\{\hat{c}_j\}_{j=1,2,\ldots,m}$.
   - $\mathbf{x}_S \leftarrow$ solution of the surrogate problem, projected on the current mesh.
   - If $\mathbf{x}_S$ is a success, repeat the search.
3. Poll
   - Construct the poll candidates.
   - Use the **surrogates** to order the poll candidates.
   - Evaluate the poll candidates *opportunistically*.
4. If no stopping criteria is met, go back to Step **2.**

Surrogate modeling techniques

- **Polynomial response surface (PRS):**

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{q} \alpha_j h_j(\mathbf{x}) \qquad \text{where } h_j(\mathbf{x}) \text{ is a polynomial of } \mathbf{x}$$

- **Radial basis function (RBF):**

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{p} \alpha_i \phi\Big(\|\mathbf{x} - \mathbf{x}_i\|_2\Big) \qquad \text{where } \phi(d) = \exp\left(-\frac{r_\phi^2 d^2}{d_{mean}^2}\right)$$

- **Kernel smoothing (KS):**

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{p} \phi\Big(\|\mathbf{x} - \mathbf{x}_i\|_2\Big)\; y(\mathbf{x}_i)}{\sum_{i=1}^{p} \phi\Big(\|\mathbf{x} - \mathbf{x}_i\|_2\Big)}$$

Ensemble of models

For each blackbox output (i.e. the objective and each constraint):

- Build an ensemble of surrogate models (Several PRS, RBF, KS, with various parameters).
- Compute the error for each model.
- Select the best model.

→ **Which error metric to use?** *(we will compare two candidates)*

Quadratic error

Root Mean Square Error (RMSE):

$$\mathcal{E}_{RMSE} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\Big(y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)\Big)^2}$$

$\rightarrow$ Quantifies the error on the training points but not the predictive accuracy outside of the training points.

Leave-one-out cross-validation

For each $\mathbf{x}_i \in \mathbf{X}$, build the model $\hat{y}^{(\text{-}i)}$ by leaving out the observation $[\mathbf{x}_i, y(\mathbf{x}_i)]$.

PRESS (Predicted REsidual Sum of Squares): **Method 1/2:**

$$\mathcal{E}_{PRESS} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\left(y(\mathbf{x}_i) - \hat{y}^{(\text{-}i)}(\mathbf{x}_i)\right)^2}$$

$\rightarrow$ Quantifies the predictive accuracy, but is the model really suited for surrogate-assisted optimization?

What is a good model for surrogate-assisted optimization

- Good model of the objective f: respects the **order** between two candidates:

$$f(\mathbf{x}) \leq f(\mathbf{x}') \Leftrightarrow \hat{f}(\mathbf{x}) \leq \hat{f}(\mathbf{x}') \text{ for all } \mathbf{x}, \mathbf{x}' \in \mathcal{X} \ .$$

- Good model of a constraint c_j: respects the **sign** of the function:

$$c_j(\mathbf{x}) \leq 0 \Leftrightarrow \hat{c}_j(\mathbf{x}) \leq 0 \text{ for all } \mathbf{x} \in \mathcal{X} \ .$$

Order error

Idea: quantify the violation of those two conditions

$$f(\mathbf{x}) \leq f(\mathbf{x}') \Leftrightarrow \hat{f}(\mathbf{x}) \leq \hat{f}(\mathbf{x}') \text{ for all } \mathbf{x}, \mathbf{x}' \in \mathcal{X}$$

$$c_j(\mathbf{x}) \leq 0 \Leftrightarrow \hat{c}_j(\mathbf{x}) \leq 0 \text{ for all } \mathbf{x} \in \mathcal{X} \ .$$

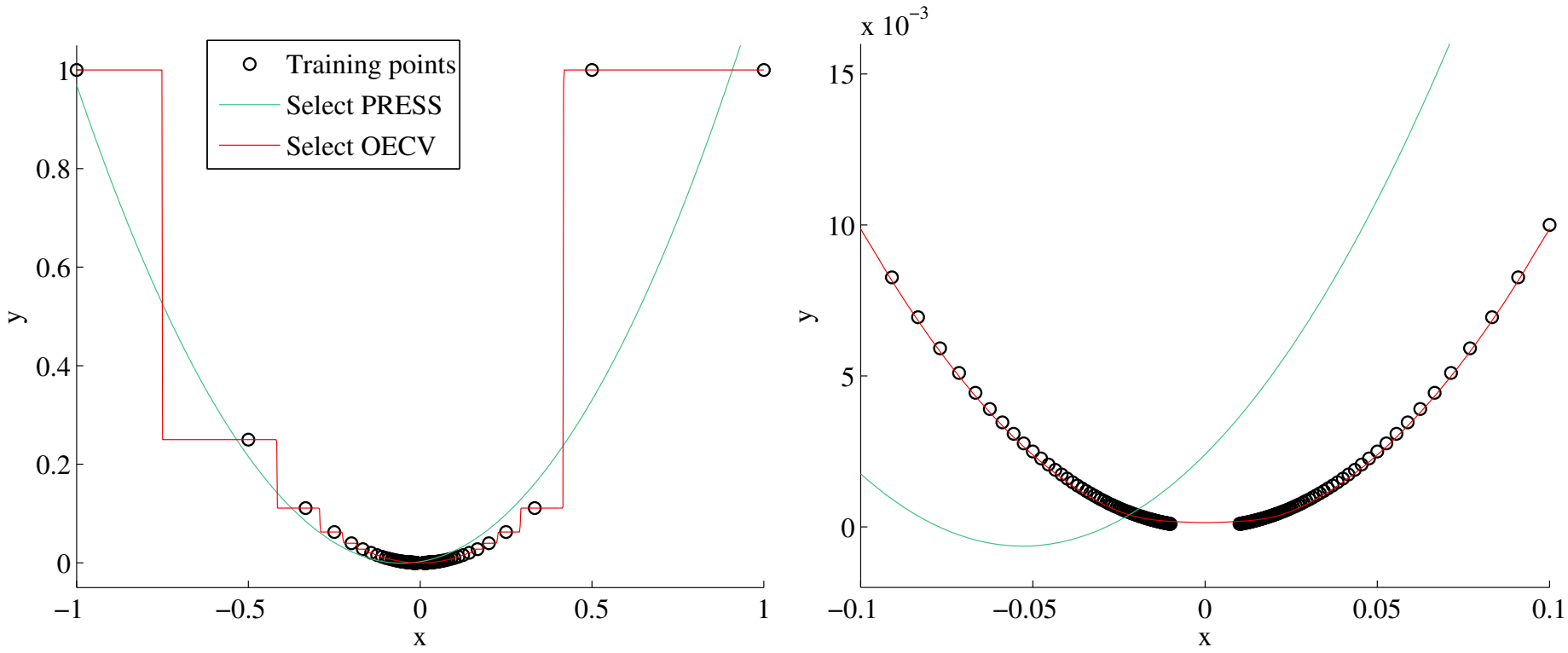
OECV (Order Error with Cross-Validation): **Method 2/2:**

$$\mathcal{E}_{OECV} = \begin{cases} \frac{1}{p^2} \sum_{i,j=1}^{p} \theta\Big(f(\mathbf{x}_i) - f(\mathbf{x}_j), \hat{f}^{(\text{-}i)}(\mathbf{x}_i) - \hat{f}^{(\text{-}j)}(\mathbf{x}_j) \Big) \\ \qquad\qquad\qquad\qquad \text{for the objective function} \\ \frac{1}{p} \sum_{i=1}^{p} \theta\Big(c(\mathbf{x}_i), \hat{c}^{(\text{-}i)}(\mathbf{x}_i) \Big) \\ \qquad\qquad\qquad\qquad \text{for a constraint function} \end{cases}$$

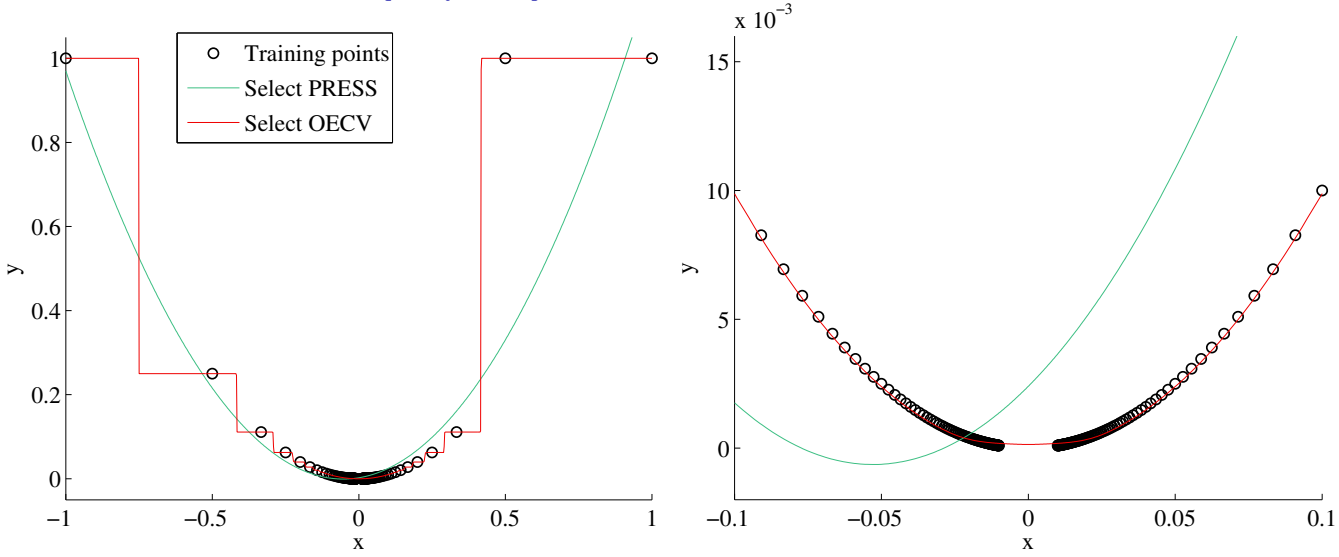
where $\theta(a, b) = (a \leq 0)$ XOR $(b \leq 0)$.

Artificial example (1/2)

$$\mathbf{X} = \{\pm 1/k, k = 1, 2, \ldots, 100\} \quad \text{and} \quad y(x) = \begin{cases} x^2 & \text{if } x \leq 1/2 \\ 1 & \text{otherwise.} \end{cases}$$



Artificial example (2/2)



- PRESS: Quadratic regression. OECV: KS with $r_\phi = 10$.
- Left figure: Quadratic regression is doing fine in general while KS looks weird.
- Right figure (zoom): KS optimizer is better.
- PRESS favors models that are generally good while OECV is more adapted to surrogate-assisted optimization.

Computational results

- Tests on two real applications from aeronautics.
- Compared methods:

Quad	—□—	MADS with local quadratic model search
Select PRESS	—▽—	MADS with Ensemble of surrogates & PRESS
Select OECV	—☆—	MADS with Ensemble of surrogates & OECV

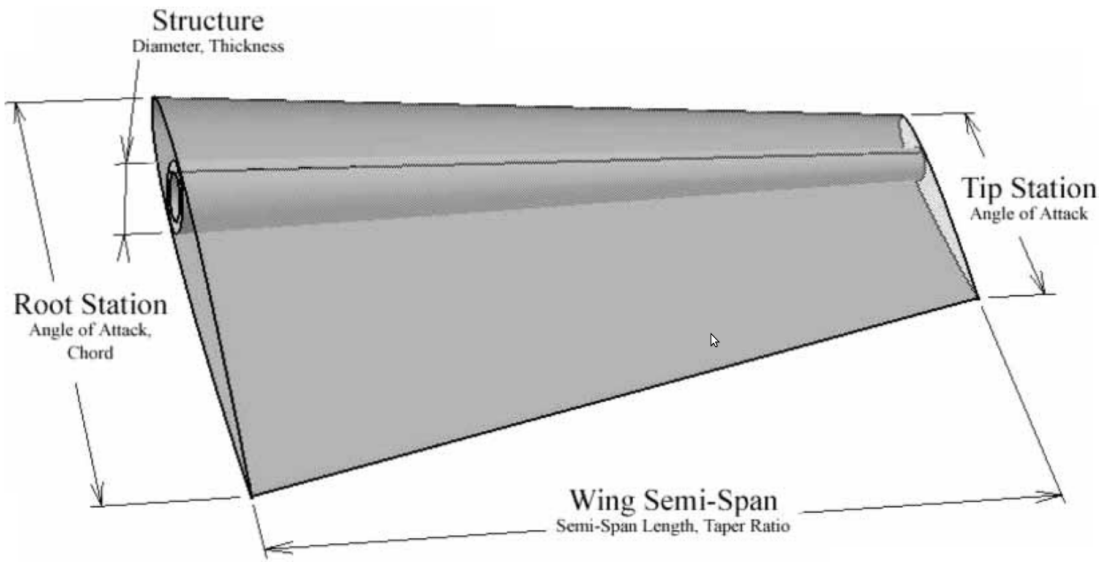
Test problem 1: MDO Simplified wing

Min: Wind drag

st: Shear stress $\leq$ 73,200 psi

Tensile stress $\leq$ 47,900 psi

Sum of the weights $\leq$ total lift

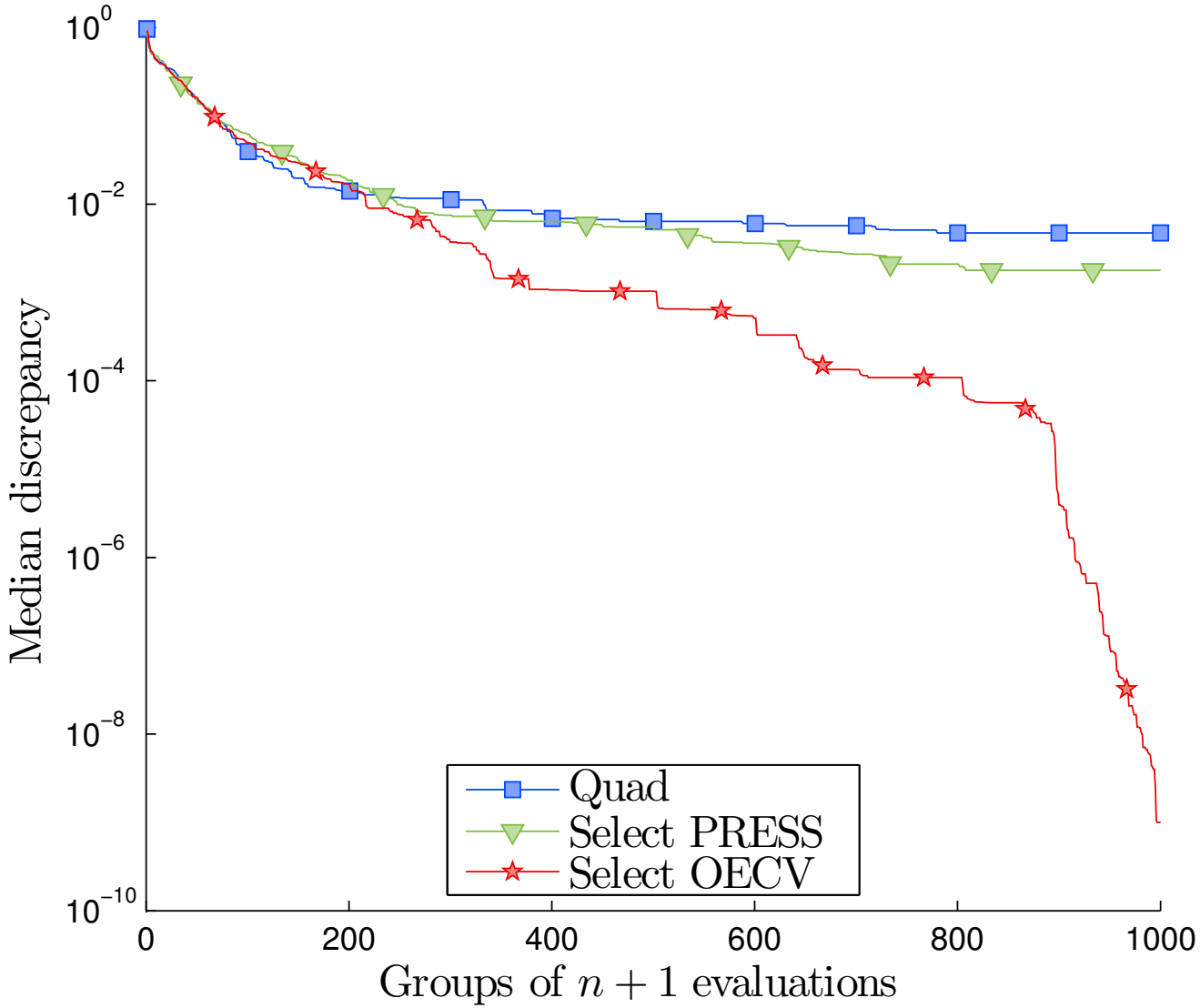


7 variables:

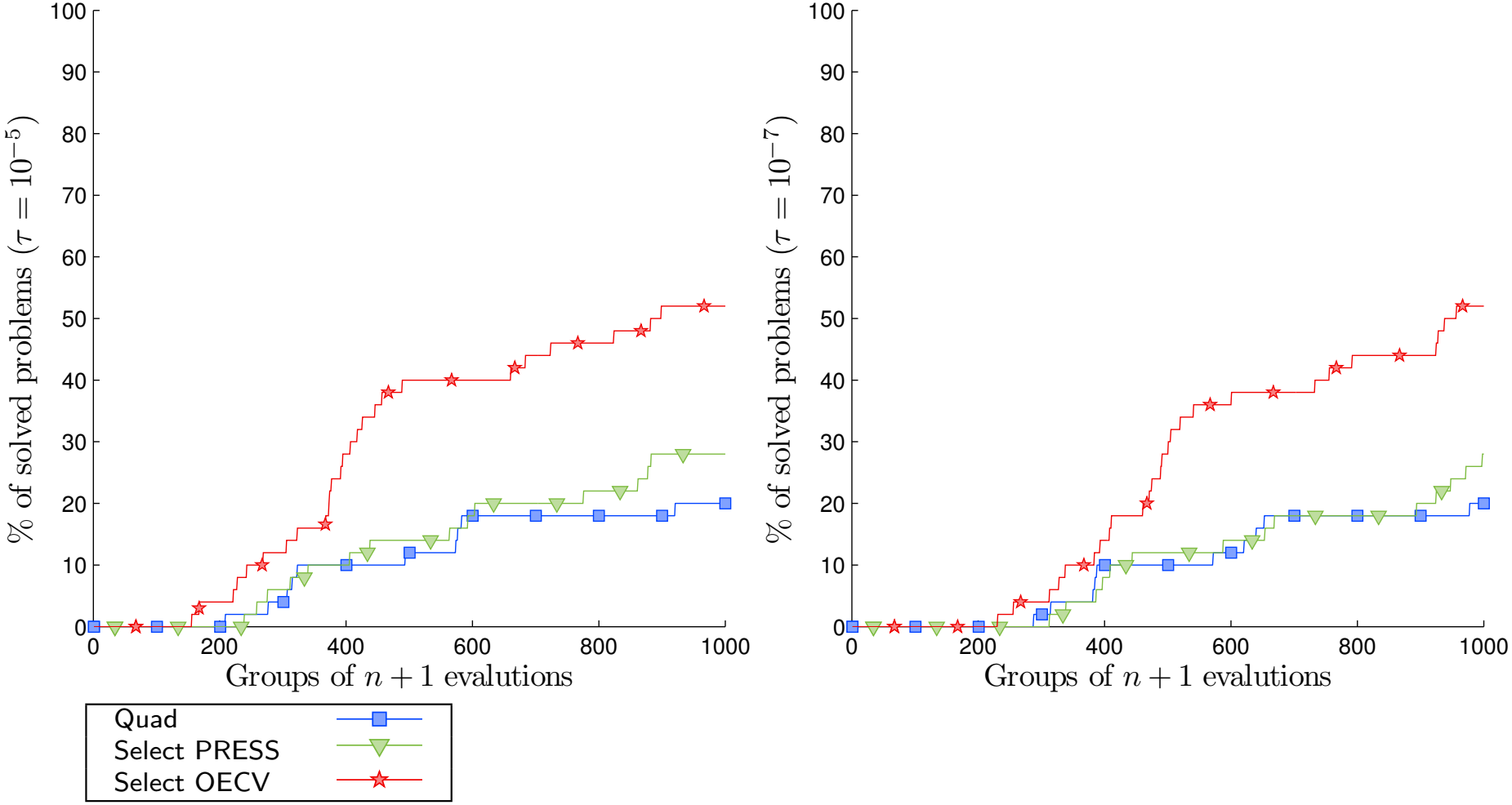
- Wing span
- Root chord
- Taper ratio
- Angle of attack at root
- Angle of attack at tip
- Tube external diameter
- Tube thickness

Decomposition of multidisciplinary optimization problems: Formulations and application to a simplified wing design, C. Tribes, J.F. Dubé and J.Y. Trépanier, Engineering Optimization, Vol. 37, No. 8, December 2005, 775–796

Test problem 1: MDO Simplified wing (50 runs)



Test problem 1: MDO Simplified wing (50 runs)



Test problem 2: MDO Aircraft range

Max: Aircraft Range

st: Stress < 1.09 ($\times 5$)

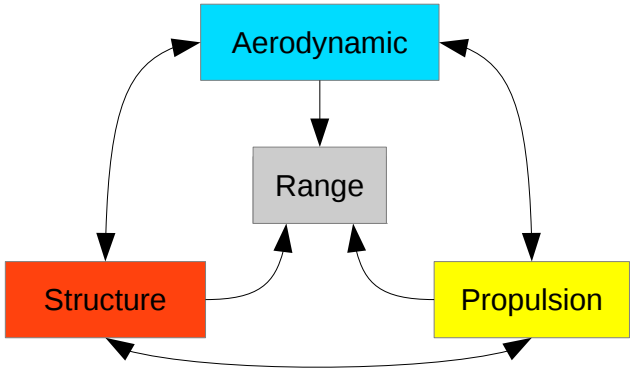
$0.96 <$ Wing Twist < 1.04

Pressure gradient < 1.04

$0.5 <$ Eng. Scale Factor < 1.5

Engine Temperature < 1.02

Throttle Setting $< T_{UA}$

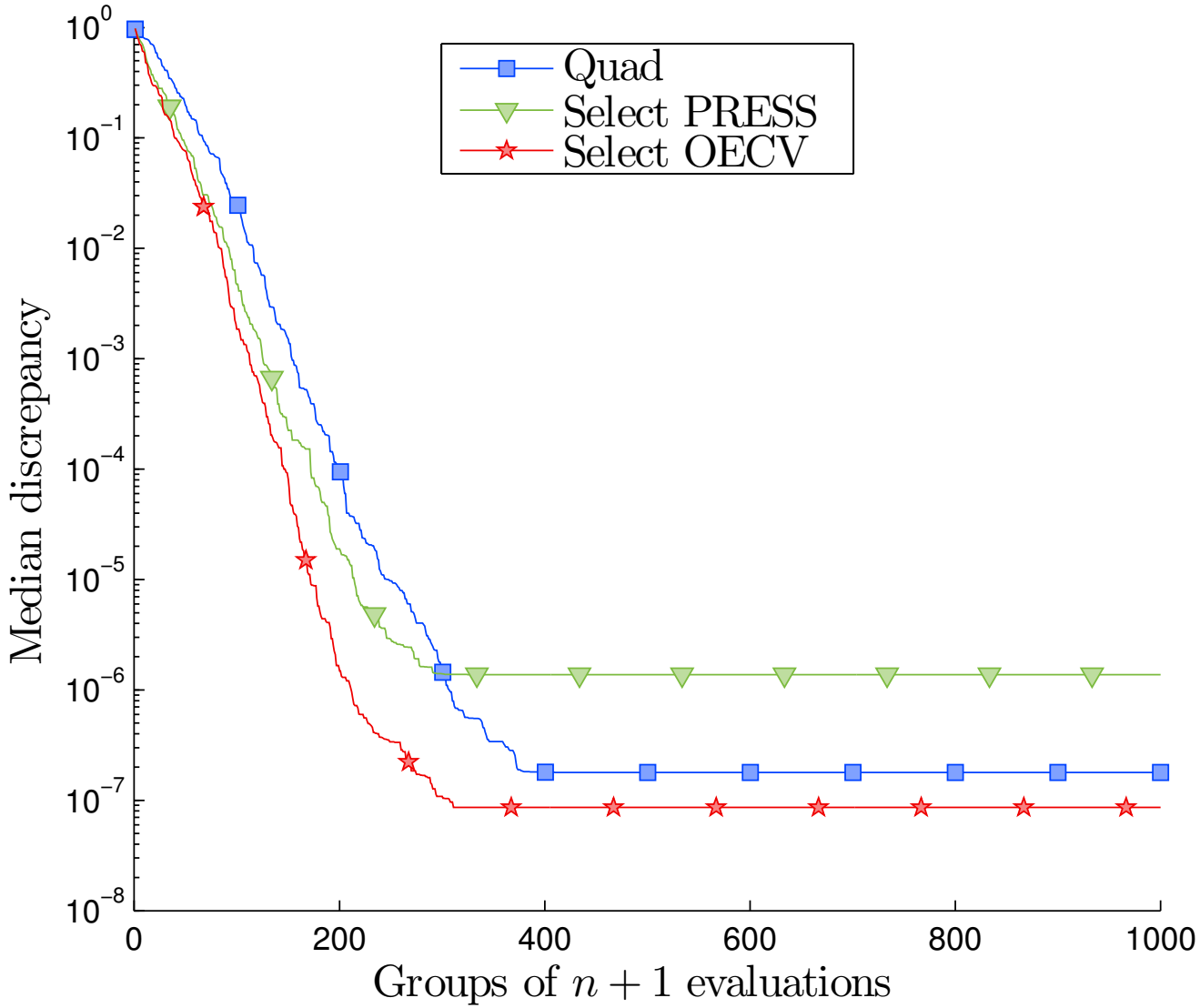


10 variables:

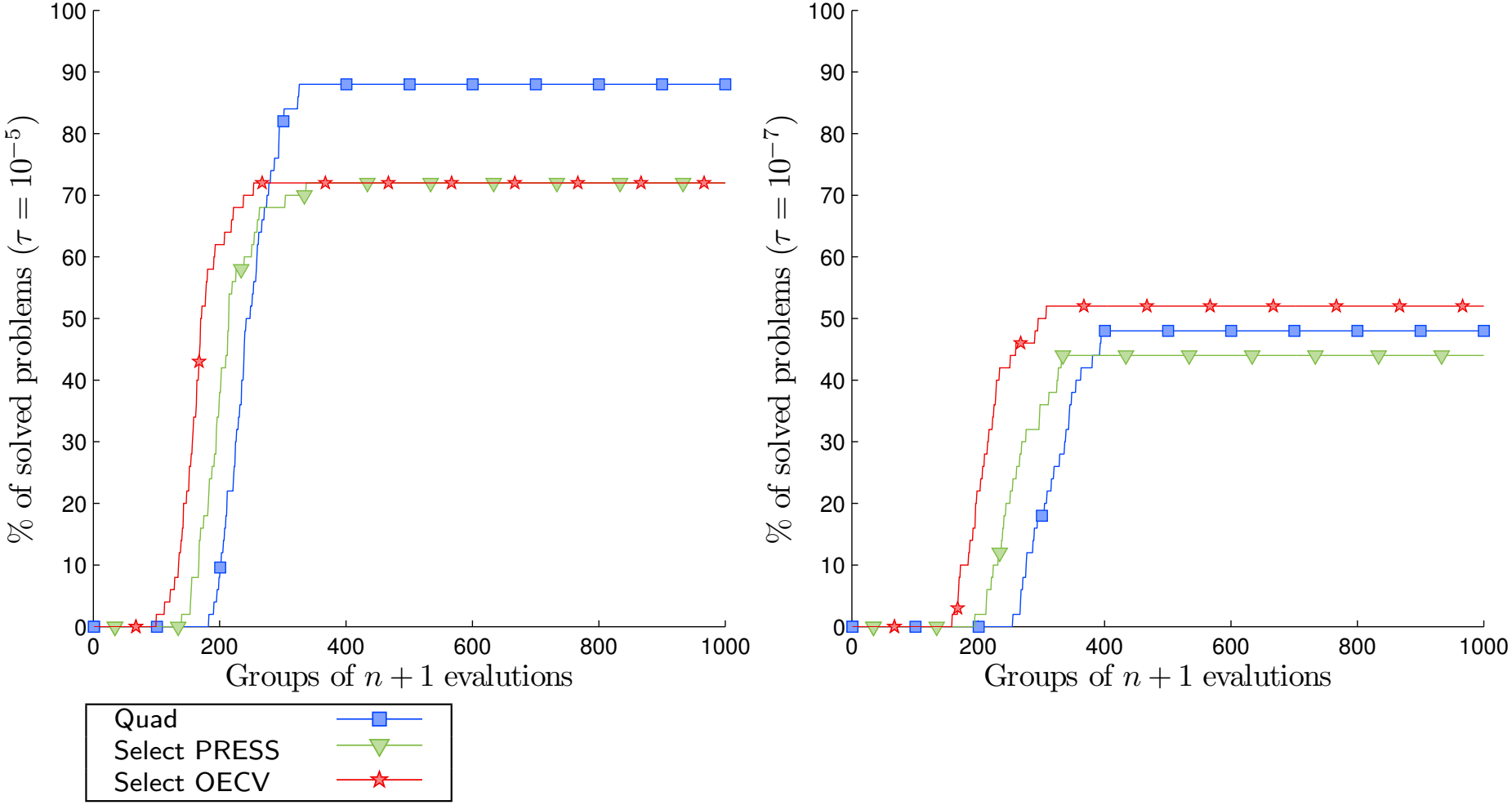
- Taper ratio
- Wingbox cross-section
- Thickness/chord
- Aspect ratio
- Wing surface area
- Wing sweep
- Skin friction coef.
- Throttle
- Altitude
- Mach number

NASA/CR-2001-211053, Multidisciplinary Aerospace Systems Optimization, Computational AeroSciences Project, S. Kodiyalam, Lockheed Martin Space Systems Company, Sunnyvale (Ca)

Test problem 2: MDO Aircraft range (50 runs)



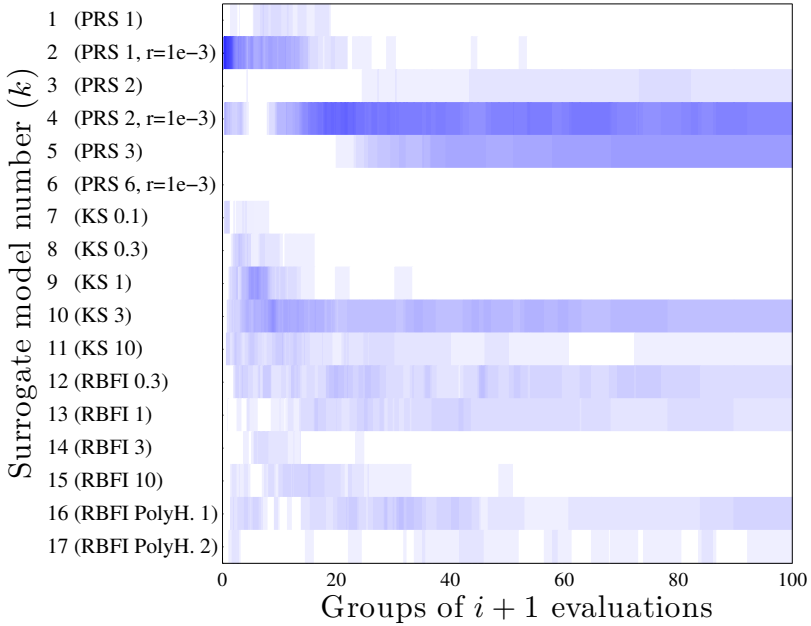
Test problem 2: MDO Aircraft range (50 runs)



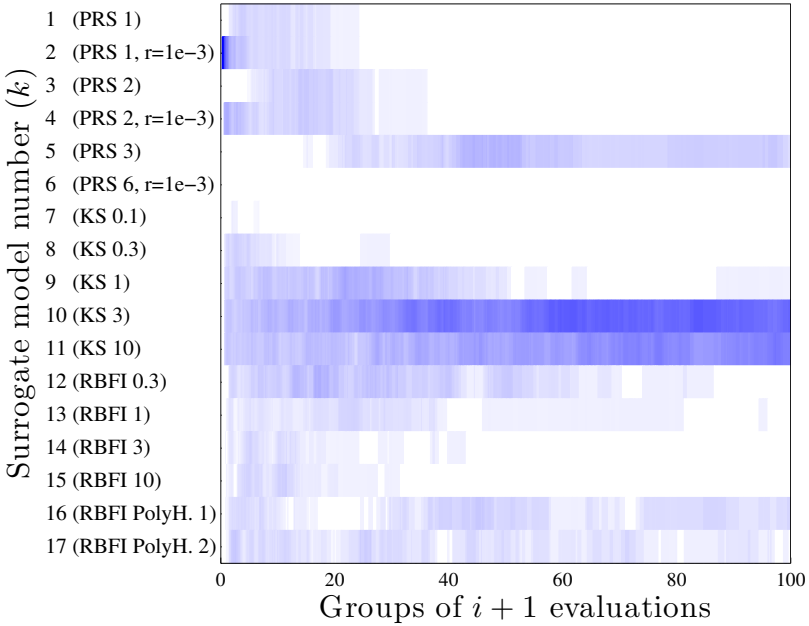
Model selection maps (example for one constraint)

Indicates how often Model k was selected during blackbox evaluation number $i(n+1)$ over 50 runs; darker tones indicate a model selected more frequently.

PRESS



OECV

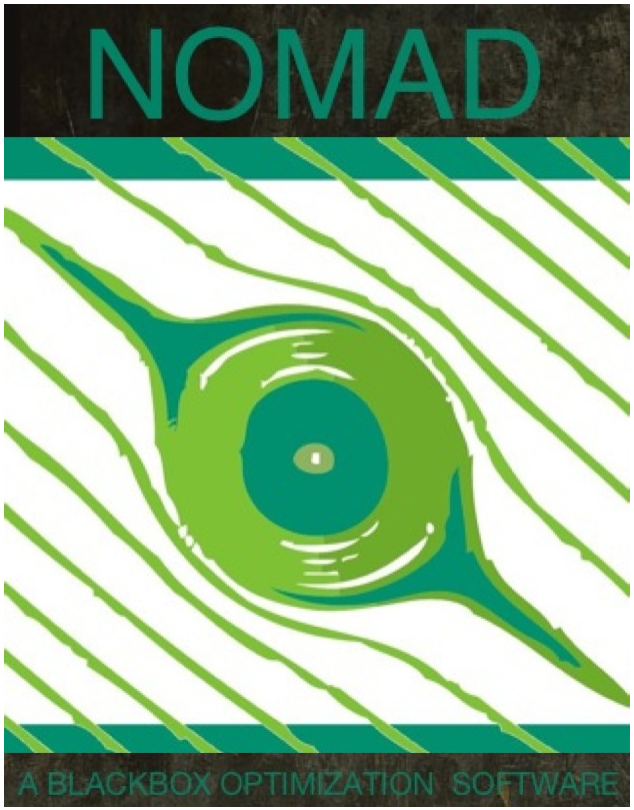


Discussion

- MADS + surrogate-assisted optimization: Guarantee of convergence + efficiency.
- We use **ensembles of surrogates** with **selection** based on a **metric**.
- We compare two metrics (PRESS and OECV):
  - Both based on **Cross-validation (CV)** that ensures a good quality of prediction outside of the training points.
  - Use either the **quadratic error (RMSE)** or the **order error (OE)**:
    - PRESS: CV + RMSE.
    - OECV: CV + OE.
- OECV is essentially designed to detect the best suited model for surrogate-assisted optimization.

NOMAD (Nonlinear Optimization with MADS)

- C++ implementation of MADS.
- Download at https://www.gerad.ca/nomad.



References I

Audet, C. and Dennis, Jr., J. (2006).
Mesh adaptive direct search algorithms for constrained optimization.
SIAM Journal on Optimization, 17(1):188–217.

Audet, C. and Kokkolaras, M. (2016).
Blackbox and derivative-free optimization: theory, algorithms and applications.
Optimization and Engineering, 17(1):1–2.

Audet, C., Kokkolaras, M., Le Digabel, S., and Talgorn, B. (2016).
Order-based error for managing ensembles of surrogates in derivative-free optimization.
Technical Report G-2016-36, Les cahiers du GERAD.

Clarke, F. (1983).
Optimization and Nonsmooth Analysis.
John Wiley & Sons, New York.
Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

References II

Kodiyalam, S. (2001).
Multidisciplinary aerospace systems optimization.
Technical Report NASA/CR-2001-211053, Lockheed Martin Space Systems Company, Computational AeroSciences Project, Sunnyvale, CA.

Le Digabel, S. (2011).
Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm.
ACM Transactions on Mathematical Software, 37(4):44:1–44:15.

Tribes, C., Dubé, J.-F., and Trépanier, J.-Y. (2005).
Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design.
Engineering Optimization, 37(8):775–796.