Radial Basis Function method for black-box optimization and application to architectural design

Alberto Costa
(joint work with Giacomo Nannicini)

Singapore University of Technology and Design, Singapore

June 27th, 2013

LIX - École Polytechnique (Palaiseau, France)

Outline

1 Preliminaries

Outline

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions

Outline

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions
3. Application: architectural design

Outline

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions
3. Application: architectural design
4. Conclusions

Where are we?

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions
3. Application: architectural design
4. Conclusions

The problem

Find the (global) optimum for this problem

$$\left.\begin{array}{lrcl} \min & f(x) & & \\ & Ax & \geq & b \\ & x & \in & [x^L, x^U] \\ & x & \in & \mathbb{Z}^q \times \mathbb{R}^{n-q}, \end{array}\right\} \text{ problem } U$$

The problem

Find the (global) optimum for this problem

$$\left.\begin{array}{lrcl} \min & f(x) & & \\ & Ax & \geq & b \\ & x & \in & [x^L, x^U] \\ & x & \in & \mathbb{Z}^q \times \mathbb{R}^{n-q}, \end{array}\right\} \text{ problem } U$$

- $f : \mathbb{R}^n \to \mathbb{R}$: objective function (continuous)

The problem

Find the (global) optimum for this problem

$$\left.\begin{array}{lrcl} \min & f(x) & & \\ & Ax & \geq & b \\ & x & \in & [x^L, x^U] \\ & x & \in & \mathbb{Z}^q \times \mathbb{R}^{n-q}, \end{array}\right\} \text{ problem } U$$

- $f : \mathbb{R}^n \to \mathbb{R}$: objective function (continuous)
- $x^L, x^U \in \mathbb{R}^n$: vectors of lower/upper bounds on the variables

The problem

Find the (global) optimum for this problem

$$\left.\begin{array}{lrcl} \min & f(x) & & \\ & Ax & \geq & b \\ & x & \in & [x^L, x^U] \\ & x & \in & \mathbb{Z}^q \times \mathbb{R}^{n-q}, \end{array}\right\} \text{ problem } U$$

- $f : \mathbb{R}^n \to \mathbb{R}$: objective function (continuous)
- $x^L, x^U \in \mathbb{R}^n$: vectors of lower/upper bounds on the variables
- $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $q \leq n$

The problem

Find the (global) optimum for this problem

$$\left.\begin{array}{rrcl} \min & f(x) & & \\ & Ax & \geq & b \\ & x & \in & [x^L, x^U] \\ & x & \in & \mathbb{Z}^q \times \mathbb{R}^{n-q}, \end{array}\right\} \text{ problem } U$$

- $f : \mathbb{R}^n \to \mathbb{R}$: objective function (continuous)
- $x^L, x^U \in \mathbb{R}^n$: vectors of lower/upper bounds on the variables
- $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $q \leq n$
- Ψ: feasible region.

Why black-box?

:(

- The analytical expression for f is unknown

Why black-box?

:(

- The analytical expression for f is unknown
- Function values only available through an oracle

Why black-box?

:(

- The analytical expression for f is unknown
- Function values only available through an oracle
- Function evaluations by means of the oracle is usually very time consuming!

Why black-box?

:(

- The analytical expression for f is unknown
- Function values only available through an oracle
- Function evaluations by means of the oracle is usually very time consuming!

Goal

- Find the global optimum with the lowest possible number of evaluations of the objective function through the oracle.

Surrogate model

We want a good "approximation" of the unknown objective function $f \Rightarrow$ surrogate model (or response surface)

Surrogate model

We want a good "approximation" of the unknown objective function $f \Rightarrow$ surrogate model (or response surface)

Some possibilities

- Kriging [Jones; JOGO, 2001]
- Efficient Global Optimization (EGO) [Jones et al.; JOGO, 1998]
- Radial Basis Function (RBF) [Gutmann; JOGO, 2001 - Holmström et al.; Optim. Eng., 2008]

Surrogate model

We want a good "approximation" of the unknown objective function $f \Rightarrow$ surrogate model (or response surface)

Some possibilities

- Kriging [Jones; JOGO, 2001]
- Efficient Global Optimization (EGO) [Jones et al.; JOGO, 1998]
- Radial Basis Function (RBF) [Gutmann; JOGO, 2001 - Holmström et al.; Optim. Eng., 2008]

We consider the RBF method.

Where are we?

1 Preliminaries

2 Radial Basis Function method
- 1. Selection of k points
- 2. Computation of the interpolant
- 3. Selection of next point (x_{k+1})
- 4. Research questions

3 Application: architectural design

4 Conclusions

General idea

1. Evaluate the function f in k points through the oracle

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method
3. Select the next point x_{k+1} where the function f should be evaluated

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method
3. Select the next point x_{k+1} where the function f should be evaluated
4. Evaluate the function f in x_{k+1} using the oracle

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method
3. Select the next point x_{k+1} where the function f should be evaluated
4. Evaluate the function f in x_{k+1} using the oracle
5. Compute the new RBF interpolant

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method
3. Select the next point x_{k+1} where the function f should be evaluated
4. Evaluate the function f in x_{k+1} using the oracle
5. Compute the new RBF interpolant
6. If some stopping condition is met, solve the original problem U using the current interpolant in place of the objective function f

General idea

1. Evaluate the function f in k points through the oracle
2. Compute the interpolant using the RBF method
3. Select the next point x_{k+1} where the function f should be evaluated
4. Evaluate the function f in x_{k+1} using the oracle
5. Compute the new RBF interpolant
6. If some stopping condition is met, solve the original problem U using the current interpolant in place of the objective function f
7. Othersiwe go to Step 3.

How to choose the initial set of k points?

First step of the method $\Rightarrow$ evaluate the function f in k points $x_1 \dots x_k$, $x_i \in \mathbb{R}^n$, obtaining the corresponding values $f(x_i)$.

How to choose the initial set of k points?

First step of the method $\Rightarrow$ evaluate the function f in k points $x_1 \dots x_k$, $x_i \in \mathbb{R}^n$, obtaining the corresponding values $f(x_i)$.

- How many points (i.e., what is the value of k)?

How to choose the initial set of k points?

First step of the method $\Rightarrow$ evaluate the function f in k points $x_1 \dots x_k$, $x_i \in \mathbb{R}^n$, obtaining the corresponding values $f(x_i)$.

- How many points (i.e., what is the value of k)?
- How to choose these k points (i.e., what are the x_i)?

Selection of number of points k

To use the RBF method the points must be π_1 unisolvent $\Rightarrow$ not all points belong to a common hyperplane.

Selection of number of points k

To use the RBF method the points must be π_1 unisolvent $\Rightarrow$ not all points belong to a common hyperplane.

Hence, among all the polynomial $ax + b$, $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ the only one that vanishes at all the points x_i is the zero polynomial.

Selection of number of points k

To use the RBF method the points must be π_1 unisolvent $\Rightarrow$ not all points belong to a common hyperplane.

Hence, among all the polynomial $ax + b$, $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ the only one that vanishes at all the points x_i is the zero polynomial.

In other words, the matrix P should have rank $n + 1$.

$$P = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_k^T & 1 \end{pmatrix}$$

Selection of number of points k

To use the RBF method the points must be π_1 unisolvent $\Rightarrow$ not all points belong to a common hyperplane.

Hence, among all the polynomial $ax + b$, $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ the only one that vanishes at all the points x_i is the zero polynomial.

In other words, the matrix P should have rank $n + 1$.

$$P = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_k^T & 1 \end{pmatrix}$$

This means that $k \geq n + 1$.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Issue: for high dimension $2^n \gg n + 1$, hence too many costly function evaluation.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Issue: for high dimension $2^n \gg n + 1$, hence too many costly function evaluation.

Reduced corner points

Select only $k = n + 1$ corner points.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Issue: for high dimension $2^n \gg n + 1$, hence too many costly function evaluation.

Reduced corner points

Select only $k = n + 1$ corner points.

Issue: this can prioritize a part of the domain.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Issue: for high dimension $2^n \gg n + 1$, hence too many costly function evaluation.

Reduced corner points

Select only $k = n + 1$ corner points.

Issue: this can prioritize a part of the domain.

Latin Hypercube Design

Select $k = n + 1$ random points following some rules.

Strategies for selecting the k points

Corner points

Select the $k = 2^n$ points on the corner of the domain.

Issue: for high dimension $2^n \gg n + 1$, hence too many costly function evaluation.

Reduced corner points

Select only $k = n + 1$ corner points.

Issue: this can prioritize a part of the domain.

Latin Hypercube Design

Select $k = n + 1$ random points following some rules.

Issue: points can be not feasible/not π_1 unisolvent.

What is a radial function?

A function $g : \mathbb{R}^n \to \mathbb{R}$ is called radial if there exists a function $\phi : [0, +\infty) \to \mathbb{R}$ such that

$$g(x) = \phi(\|x\|),$$

where $\|\cdot\|$ is the Euclidean norm.

What is a radial function?

A function $g : \mathbb{R}^n \to \mathbb{R}$ is called radial if there exists a function $\phi : [0, +\infty) \to \mathbb{R}$ such that

$$g(x) = \phi(\|x\|),$$

where $\|\cdot\|$ is the Euclidean norm.

Intuitively, the function depends from the distance of a point $x \in \mathbb{R}^n$ from the origin.

RBF interpolation: idea

Each point $x_i \in X$ "influences" its sorrounding in the same way in all directions, according to some assumed functional form $\phi(r)$.

RBF interpolation: idea

Each point $x_i \in X$ "influences" its sorrounding in the same way in all directions, according to some assumed functional form $\phi(r)$.

The function $\phi(r)$ only depends on the radial distance $r = \|x - x_i\|$ from the point x_i.

RBF interpolation: idea

Each point $x_i \in X$ "influences" its sorrounding in the same way in all directions, according to some assumed functional form $\phi(r)$.

The function $\phi(r)$ only depends on the radial distance $r = \|x - x_i\|$ from the point x_i.

Hence, a point can be determined as a linear combination of $\phi(\|x - x_i\|)$, $\forall x_i \in X$.

RBF interpolation - 1

Given the set of points $X = \{x_1 \dots x_k\}$, $x_i \in \mathbb{R}^n$ and a RBF function ϕ, the RBF interpolant of the points in X is defined as:

$$s_k(x) := \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x),$$

RBF interpolation - 1

Given the set of points $X = \{x_1 \dots x_k\}$, $x_i \in \mathbb{R}^n$ and a RBF function ϕ, the RBF interpolant of the points in X is defined as:

$$s_k(x) := \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x),$$

where $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ and p is a polynomial of degree $d \geq d_{\min}$. The value of $d_{\min}$ depends on the RBF function $\phi(r)$.

RBF interpolation - 1

Given the set of points $X = \{x_1 \dots x_k\}$, $x_i \in \mathbb{R}^n$ and a RBF function ϕ, the RBF interpolant of the points in X is defined as:

$$s_k(x) := \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x),$$

where $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ and p is a polynomial of degree $d \geq d_{\min}$. The value of $d_{\min}$ depends on the RBF function $\phi(r)$.

We may need $p(x)$ in order to ensure that the interpolant exists!

RBF interpolation - 2

The most common RBF functions are

$\phi(r)$		$d_{\min}$
r	(linear)	0
r^3	(cubic)	1
$r^2 \log r$	(thin splate spline)	1
$\sqrt{r^2+\gamma^2}$	(multiquadric)	0
$\frac{1}{\sqrt{r^2+\gamma^2}}$	(inverse multiquadric)	-1
$e^{-\gamma r^2}$	(Gaussian)	-1

RBF interpolation - 2

The most common RBF functions are

$\phi(r)$		$d_{\min}$
r	(linear)	0
r^3	(cubic)	1
$r^2 \log r$	(thin splate spline)	1
$\sqrt{r^2+\gamma^2}$	(multiquadric)	0
$\frac{1}{\sqrt{r^2+\gamma^2}}$	(inverse multiquadric)	-1
$e^{-\gamma r^2}$	(Gaussian)	-1

The parameter $\gamma > 0$ can be used to change the shape of these functions.

RBF interpolation - 3

The interpolant $s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x)$ must respect these conditions:

RBF interpolation - 3

The interpolant $s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x)$ must respect these conditions:

1. $\forall x_i \in X \quad s_k(x_i) = f(x_i)$

RBF interpolation - 3

The interpolant $s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x)$ must respect these conditions:

1. $\forall x_i \in X \quad s_k(x_i) = f(x_i)$
2. $\forall j \in \{1, \ldots, Q\} \quad \sum_{i=1}^{k} \lambda_i P_{ij} = 0,\ Q = \max(0, d_{\min} \cdot n + 1)$

RBF interpolation - 3

The interpolant $s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x)$ must respect these conditions:

1. $\forall x_i \in X \quad s_k(x_i) = f(x_i)$
2. $\forall j \in \{1, \ldots, Q\} \quad \sum_{i=1}^{k} \lambda_i P_{ij} = 0,\ Q = \max(0, d_{\min} \cdot n + 1)$

Condition (1) means that the interpolant value in the point x_i is $f(x_i)$, i.e., the same value given by the oracle.

RBF interpolation - 3

The interpolant $s_k(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - x_i\|) + p(x)$ must respect these conditions:

1. $\forall x_i \in X \quad s_k(x_i) = f(x_i)$
2. $\forall j \in \{1, \ldots, Q\} \quad \sum_{i=1}^{k} \lambda_i P_{ij} = 0,\ Q = \max(0, d_{\min} \cdot n + 1)$

Condition (1) means that the interpolant value in the point x_i is $f(x_i)$, i.e., the same value given by the oracle.

Condition (2) is used to ensure the existence of the interpolant. It guarantees the conditionally positive definiteness of the RBF (see `[Jakobsson et al.; Optim and Eng., 2010]` for details).

RBF interpolation - 4

This is equivalent to solve the following linear system:

$$\begin{pmatrix} \Phi & P \\ P^T & 0_{Q\times Q} \end{pmatrix} \begin{pmatrix} \lambda \\ h \end{pmatrix} = \begin{pmatrix} F \\ 0_Q \end{pmatrix}$$

RBF interpolation - 4

This is equivalent to solve the following linear system:

$$\begin{pmatrix} \Phi & P \\ P^T & 0_{Q\times Q} \end{pmatrix} \begin{pmatrix} \lambda \\ h \end{pmatrix} = \begin{pmatrix} F \\ 0_Q \end{pmatrix}$$

with:

$$\Phi = (\phi(\|x_i - x_j\|))_{i,j=1,\dots,k}\,,\; P = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_k^T & 1 \end{pmatrix},\; \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix},\; F = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix}$$

RBF interpolation - 4

This is equivalent to solve the following linear system:

$$\begin{pmatrix} \Phi & P \\ P^T & 0_{Q\times Q} \end{pmatrix} \begin{pmatrix} \lambda \\ h \end{pmatrix} = \begin{pmatrix} F \\ 0_Q \end{pmatrix}$$

with:

$$\Phi = (\phi(\|x_i - x_j\|))_{i,j=1,\dots,k}\,,\; P = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_k^T & 1 \end{pmatrix},\; \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix},\; F = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix}$$

$\Phi\lambda + Ph = F$ is condition (1); $P^T\lambda = 0$ is condition (2).

RBF interpolation - 5

- $d_{\min} = 1$ (i.e., cubic and thin splate RBF): $Q = n + 1$, h is a $n + 1$ dimension vector representing the coefficients of the polynomial $ax + b$ ($[a\, b] = h^T$).

RBF interpolation - 5

- $d_{\min} = 1$ (i.e., cubic and thin splate RBF): $Q = n + 1$, h is a $n + 1$ dimension vector representing the coefficients of the polynomial $ax + b$ ($[a\, b] = h^T$).
- $d_{\min} = 0$ (i.e., linear and multiquadric RBF): $Q = 1$, h represent the term b of the polynomial, and P is the all-one column vector of dimension k.

RBF interpolation - 5

- $d_{\min} = 1$ (i.e., cubic and thin splate RBF): $Q = n + 1$, h is a $n + 1$ dimension vector representing the coefficients of the polynomial $ax + b$ ($[a\, b] = h^T$).
- $d_{\min} = 0$ (i.e., linear and multiquadric RBF): $Q = 1$, h represent the term b of the polynomial, and P is the all-one column vector of dimension k.
- $d_{\min} = -1$ (i.e., inverse multiquadric and Gaussian RBF): $Q = 0$, h and P are removed (i.e., the system becomes $\Phi\lambda = F$).

RBF interpolation - 5

- $d_{\min} = 1$ (i.e., cubic and thin splate RBF): $Q = n + 1$, h is a $n + 1$ dimension vector representing the coefficients of the polynomial $ax + b$ ($[a\, b] = h^T$).
- $d_{\min} = 0$ (i.e., linear and multiquadric RBF): $Q = 1$, h represent the term b of the polynomial, and P is the all-one column vector of dimension k.
- $d_{\min} = -1$ (i.e., inverse multiquadric and Gaussian RBF): $Q = 0$, h and P are removed (i.e., the system becomes $\Phi\lambda = F$).

We consider the case of cubic and thin splate RBF (i.e., $d_{\min} = 1$), which seem to give best results for our application.

Next point

Where should we evaluate the function through the oracle now?

Next point

Where should we evaluate the function through the oracle now?

- Select a target value f_k^* (how to choose it is shown later)

Next point

Where should we evaluate the function through the oracle now?

- Select a target value f_k^* (how to choose it is shown later)
- Compute the point x_{k+1} such that the interpolant s_y which respects the conditions $\forall x_i \in X \; s_y(x_i) = f(x_i)$ and $s_y(x_{k+1}) = f_k^*$ is the best possible

Next point

Where should we evaluate the function through the oracle now?

- Select a target value f_k^* (how to choose it is shown later)
- Compute the point x_{k+1} such that the interpolant s_y which respects the conditions $\forall x_i \in X \, s_y(x_i) = f(x_i)$ and $s_y(x_{k+1}) = f_k^*$ is the best possible
- Evaluate the function through the oracle in x_{k+1} obtaining f_{k+1} and compute the new RBF interpolant s_{k+1} which respect the conditions $\forall x_i \in X \, s_{k+1}(x_i) = f(x_i)$ and $s_{k+1}(x_{k+1}) = f_k^*$ (solving a linear system as done before)

How to compute the point x_{k+1}? - 1

In order to find x_{k+1}, and consequently the best RBF interpolant s_y, we should minimize its bumpiness.

How to compute the point x_{k+1}? - 1

In order to find x_{k+1}, and consequently the best RBF interpolant s_y, we should minimize its bumpiness.

Why? Because the RBF interpolant has the property to be the less "bumpy" among all the functions which assume value $f(x_i)$ in x_i.

How to compute the point x_{k+1}? - 1

In order to find x_{k+1}, and consequently the best RBF interpolant s_y, we should minimize its bumpiness.

Why? Because the RBF interpolant has the property to be the less "bumpy" among all the functions which assume value $f(x_i)$ in x_i.

The bumpiness can be expressed as $(-1)^{d_{\min}+1}\lambda^T\Phi\lambda$.

How to compute the point x_{k+1}? - 2

It can be shown [Gutmann; JOGO, 2001] that the point x_{k+1} for which s_y is less bumpy is found by solving the optimization problem:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{(-1)^{d_{\min}+1} \mu_k(x) [s_k(x) - f_k^*]^2},$$

How to compute the point x_{k+1}? - 2

It can be shown [Gutmann; JOGO, 2001] that the point x_{k+1} for which s_y is less bumpy is found by solving the optimization problem:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{(-1)^{d_{\min}+1} \mu_k(x) [s_k(x) - f_k^*]^2},$$

where $\mu_k(x) = \dfrac{1}{\phi(0) - (u(x)\ \pi(x))^T \begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix}^{-1} (u(x)\ \pi(x))},$

with $u(x) = (\phi(\|x - x_1\|), \ldots, \phi(\|x - x_k\|))^T$ and $\pi(x)$ is $\begin{pmatrix} x^T & 1 \end{pmatrix}$ when $d_{\min} = 1$, 1 when $d_{\min} = 0$ and not used when $d_{\min} = -1$.

How to compute the target value f_k^*? - 1

Let $y^* := \arg\min_{x \in \Psi} s_k(x)$ and $f_{\min} := \min_{i=1,\dots,k} f(x_i)$.

How to compute the target value f_k^*? - 1

Let $y^* := \arg\min_{x \in \Psi} s_k(x)$ and $f_{\min} := \min_{i=1,\ldots,k} f(x_i)$.

In order to assure the convergence of the method, a cyclic strategy of length $\kappa + 2$ for selecting f_k^* is employed:

How to compute the target value f_k^*? - 1

Let $y^* := \arg\min_{x \in \Psi} s_k(x)$ and $f_{\min} := \min_{i=1,\ldots,k} f(x_i)$.

In order to assure the convergence of the method, a cyclic strategy of length $\kappa + 2$ for selecting f_k^* is employed:

- Step -1 (InfStep): Choose $f_k^* = -\infty$.

How to compute the target value f_k^*? - 1

Let $y^* := \arg\min_{x \in \Psi} s_k(x)$ and $f_{\min} := \min_{i=1,\ldots,k} f(x_i)$.

In order to assure the convergence of the method, a cyclic strategy of length $\kappa + 2$ for selecting f_k^* is employed:

- Step -1 (InfStep): Choose $f_k^* = -\infty$.
- Step $h = 0, \ldots, \kappa - 1$ (Global search): Choose $f_k^* = s_k(y^*) - (1 - h/\kappa)(\max_{i=1,\ldots,k} f(x_i) - s_k(y^*))$.

How to compute the target value f_k^*? - 1

Let $y^* := \arg\min_{x \in \Psi} s_k(x)$ and $f_{\min} := \min_{i=1,\ldots,k} f(x_i)$.

In order to assure the convergence of the method, a cyclic strategy of length $\kappa + 2$ for selecting f_k^* is employed:

- Step -1 (InfStep): Choose $f_k^* = -\infty$.
- Step $h = 0, \ldots, \kappa - 1$ (Global search): Choose $f_k^* = s_k(y^*) - (1 - h/\kappa)(\max_{i=1,\ldots,k} f(x_i) - s_k(y^*))$.
- Step κ (Local search): If $s_k(y^*) < f_{\min} - 10^{-10}|f_{\min}|$ accept y^* as the new sample point x_{k+1}. Otherwise choose $f_k^* = f_{\min} - 10^{-2}|f_{\min}|$.

How to compute the target value f_k^*? - 2

In general $f_k^* \in [-\infty, s_k(y^*)]$

How to compute the target value f_k^*? - 2

In general $f_k^* \in [-\infty, s_k(y^*)]$

- If $f_k^* = -\infty$ (InfStep), the problem of finding x_{k+1} can be rewritten as:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{\mu_k(x)}$$

How to compute the target value f_k^*? - 2

In general $f_k^* \in [-\infty, s_k(y^*)]$

- If $f_k^* = -\infty$ (InfStep), the problem of finding x_{k+1} can be rewritten as:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{\mu_k(x)}$$

- If $f_k^* = s_k(y^*)$, the problem of finding x_{k+1} can be rewritten as:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{s_k(x)}$$

How to compute the target value f_k^*? - 2

In general $f_k^* \in [-\infty, s_k(y^*)]$

- If $f_k^* = -\infty$ (InfStep), the problem of finding x_{k+1} can be rewritten as:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{\mu_k(x)}$$

- If $f_k^* = s_k(y^*)$, the problem of finding x_{k+1} can be rewritten as:

$$x_{k+1} = \arg\max_{x \in \Psi} \frac{1}{s_k(x)}$$

So there is no need to solve this problem.

Some questions

- How to choose the RBF function?

Some questions

- How to choose the RBF function?
- How to solve the MINLP problem $\min_{x \in \Psi} s_k(x)$?

Some questions

- How to choose the RBF function?
- How to solve the MINLP problem $\min_{x \in \Psi} s_k(x)$?
- How to solve the MINLP problem $\max_{x \in \Psi} \frac{1}{(-1)^{d_{\min}+1} \mu_k(x) [s_k(x) - f_k^*]^2}$?

Some questions

- How to choose the RBF function?
- How to solve the MINLP problem $\min_{x \in \Psi} s_k(x)$?
- How to solve the MINLP problem $\max_{x \in \Psi} \frac{1}{(-1)^{d_{\min}+1} \mu_k(x)[s_k(x) - f_k^*]^2}$?

Some questions

- How to choose the RBF function?
- How to solve the MINLP problem $\min_{x \in \Psi} s_k(x)$?
- How to solve the MINLP problem $\max_{x \in \Psi} \frac{1}{(-1)^{d_{\min}+1} \mu_k(x)[s_k(x) - f_k^*]^2}$?

Last 2 questions: still under investigation. So far local search methods are employed. Is it worth to find the global optimum?

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points
3. Compute the RBF interpolant s_j using these $k-1$ points

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points
3. Compute the RBF interpolant s_j using these $k-1$ points
4. Set error $\leftarrow$ error $+ |s_j(x_j) - f_j|$

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points
3. Compute the RBF interpolant s_j using these $k-1$ points
4. Set error $\leftarrow$ error $+ |s_j(x_j) - f_j|$
5. Set $j \leftarrow j+1$

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points
3. Compute the RBF interpolant s_j using these $k-1$ points
4. Set error $\leftarrow$ error $+ |s_j(x_j) - f_j|$
5. Set $j \leftarrow j+1$
6. If $j \leq k$ goto step 2; otherwise terminate.

RBF choice: cross validation - 1

One way to decide which one is the best RBF for a given problem is to use the cross validation on the initial k points.

1. Select a RBF function; set $i \leftarrow 1$ and error $\leftarrow 0$
2. Remove the $j-th$ point among the k initial points
3. Compute the RBF interpolant s_j using these $k-1$ points
4. Set error $\leftarrow$ error $+ |s_j(x_j) - f_j|$
5. Set $j \leftarrow j+1$
6. If $j \leq k$ goto step 2; otherwise terminate.

We choose the RBF function which provides the smallest value of error.

RBF choice: cross validation - 2

To do this, we should solve k times the system to find the parameters of the RBF. Can we do better?

RBF choice: cross validation - 2

To do this, we should solve k times the system to find the parameters of the RBF. Can we do better?

Recall that the bumpiness can be expressed as $(-1)^{d_{\min}+1}\lambda^T\Phi\lambda$

RBF choice: cross validation - 2

To do this, we should solve k times the system to find the parameters of the RBF. Can we do better?

Recall that the bumpiness can be expressed as $(-1)^{d_{\min}+1}\lambda^T\Phi\lambda$

It is possible to find the RBF by solving a mathematical programming problem instead of solving a system.

Mathematical programming formulation - 1

The problem of finding the parameters of the interpolant can be modeled as follows:

$$\begin{aligned} \min \quad & f(x) & & \\ \text{s.t.:} \quad & \Phi\lambda + Ph & = & F \\ & P^T\lambda & = & 0_{n+1}, \end{aligned}$$

where the objective function $f(x)$ can be any function.

Mathematical programming formulation - 1

The problem of finding the parameters of the interpolant can be modeled as follows:

$$\begin{aligned} \min \quad & f(x) & & \\ \text{s.t.:} \quad & \Phi\lambda + Ph & = & F \\ & P^T\lambda & = & 0_{n+1}, \end{aligned}$$

where the objective function $f(x)$ can be any function.

Why any function and not the bumpiness $(-1)^{d_{\min}+1}\lambda^T\Phi\lambda$?

Mathematical programming formulation - 1

The problem of finding the parameters of the interpolant can be modeled as follows:

$$\begin{aligned} \min \quad & f(x) & & \\ \text{s.t.:} \quad & \Phi\lambda + Ph & = & F \\ & P^T\lambda & = & 0_{n+1}, \end{aligned}$$

where the objective function $f(x)$ can be any function.

Why any function and not the bumpiness $(-1)^{d_{\min}+1}\lambda^T\Phi\lambda$?

Because there is only 1 feasible point for this problem (the same obtained by solving the linear system), with the property to minimize the bumpiness.

Mathematical programming formulation - 2

Can we modify the model to take into account the cross validation?

Mathematical programming formulation - 2

Can we modify the model to take into account the cross validation?

$$
\begin{aligned}
\min \quad & (-1)^{d_{\min}+1} \lambda^T \Phi \lambda & & \\
\text{s.t.:} \quad & \Phi\lambda + Ph + \xi &=& F \\
& P^T \lambda &=& 0_{n+1} \\
\forall i \in \{1,\dots,k\} \setminus \{j\} \quad & 0 \le \xi_i &\le& 0 \\
& -\infty \le \xi_j &\le& +\infty \\
& \lambda_j &=& 0.
\end{aligned}
$$

Mathematical programming formulation - 2

Can we modify the model to take into account the cross validation?

$$
\begin{array}{lrcl}
\min & (-1)^{d_{\min}+1}\lambda^T\Phi\lambda & & \\
\text{s.t.:} & \Phi\lambda + Ph + \xi & = & F \\
 & P^T\lambda & = & 0_{n+1} \\
\forall i \in \{1,\ldots,k\}\setminus\{j\} & 0 \leq \xi_i & \leq & 0 \\
 & -\infty \leq \xi_j & \leq & +\infty \\
 & \lambda_j & = & 0.
\end{array}
$$

$\lambda_j = 0$, so the $j-th$ point is removed from the objective function and has no effect on the constraint $P^T\lambda = 0_{n+1}$

Mathematical programming formulation - 2

Can we modify the model to take into account the cross validation?

$$
\begin{array}{lrcl}
\min & (-1)^{d_{\min}+1}\lambda^T\Phi\lambda & & \\
\text{s.t.:} & \Phi\lambda + Ph + \xi & = & F \\
 & P^T\lambda & = & 0_{n+1} \\
\forall i \in \{1,\dots,k\}\setminus\{j\} & 0 \le \xi_i & \le & 0 \\
 & -\infty \le \xi_j & \le & +\infty \\
 & \lambda_j & = & 0.
\end{array}
$$

$\lambda_j = 0$, so the $j-th$ point is removed from the objective function and has no effect on the constraint $P^T\lambda = 0_{n+1}$

ξ_j free, so the constraint $\Phi\lambda + Ph + \xi = F$ involving F_j is deactivated.

Reformulation - 1

The model can be reformulated exactly (same optimal solutions) changing the objective function.

Reformulation - 1

The model can be reformulated exactly (same optimal solutions) changing the objective function.

$$\lambda^T \Phi \lambda = \lambda^T (F - Ph - \xi) = \lambda^T F - \lambda^T P h - \lambda^T \xi.$$

Reformulation - 1

The model can be reformulated exactly (same optimal solutions) changing the objective function.

$$\lambda^T \Phi \lambda = \lambda^T (F - Ph - \xi) = \lambda^T F - \lambda^T P h - \lambda^T \xi.$$

$$P^T \lambda = \lambda^T P = 0_{n+1}$$

Reformulation - 1

The model can be reformulated exactly (same optimal solutions) changing the objective function.

$$\lambda^T \Phi \lambda = \lambda^T (F - Ph - \xi) = \lambda^T F - \lambda^T P h - \lambda^T \xi.$$

$$P^T \lambda = \lambda^T P = 0_{n+1}$$

Moreover, since $\forall i \neq j \ \xi_i = 0$ and $\lambda_j = 0$, we have

$$\lambda^T \xi = \sum_{i=1,\, i \neq j}^{k} \lambda_i \xi_i + \lambda_j \xi_j = 0.$$

Reformulation - 2

The resulting model is the following;

$$
\begin{aligned}
\min \quad & (-1)^{d_{\min}+1}\lambda^T F & & \\
\text{s.t.:} \quad & \Phi\lambda + Ph + \xi &=& F \\
& P^T\lambda &=& 0_{n+1} \\
\forall i \in \{1,\dots,k\}\setminus\{j\} \quad & 0 \le \xi_i &\le& 0 \\
& -\infty \le \xi_j &\le& +\infty \\
& \lambda_j &=& 0.
\end{aligned}
$$

Reformulation - 2

The resulting model is the following;

$$
\begin{array}{rrcl}
\min & (-1)^{d_{\min}+1}\lambda^T F & & \\
\text{s.t.:} & \Phi\lambda + Ph + \xi & = & F \\
 & P^T\lambda & = & 0_{n+1} \\
\forall i \in \{1,\dots,k\}\setminus\{j\} & 0 \le \xi_i & \le & 0 \\
 & -\infty \le \xi_j & \le & +\infty \\
 & \lambda_j & = & 0.
\end{array}
$$

We can use the same model, that is a LP, and only change the bounds for some variables (i.e., ξ_j and λ_j) for cross validation. A solver could solve the k problems needed for the cross validation effciently by warm-starting the dual simplex method.

Where are we?

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions
3. Application: architectural design
4. Conclusions

The project

This project deals with the design of a new contemporary church in Singapore optimizing the use of daylight (with S. Chaltiel and T. Schroepfer, SUTD).

The project

This project deals with the design of a new contemporary church in Singapore optimizing the use of daylight (with S. Chaltiel and T. Schroepfer, SUTD).

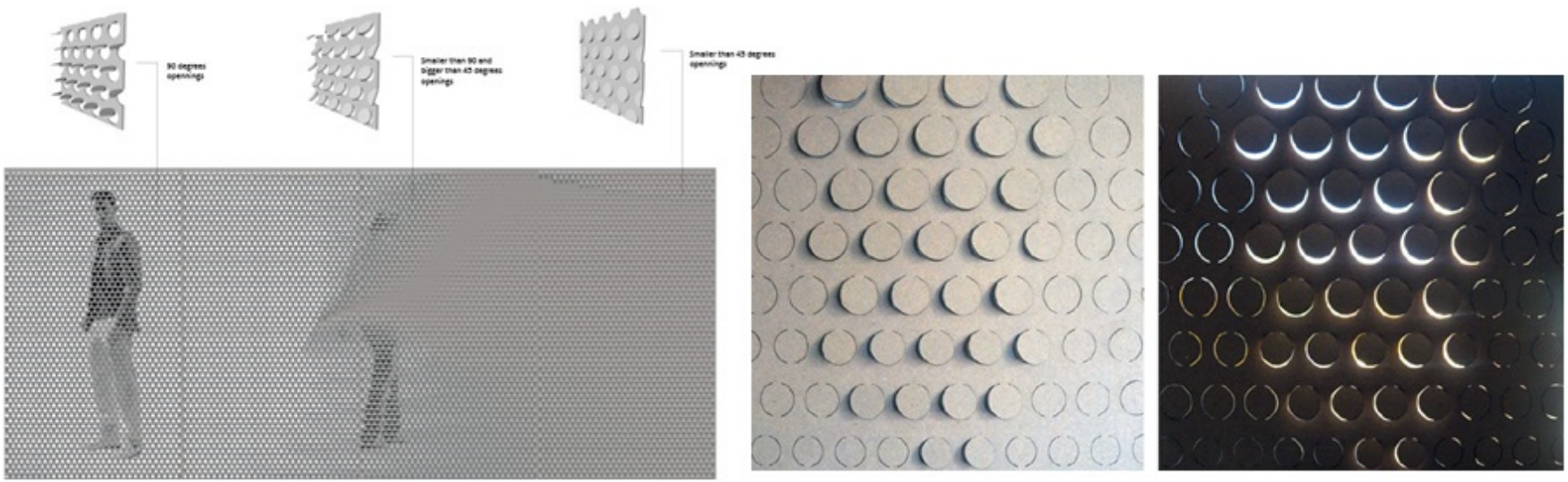
This work is based on the exploration of a performative facade aiming at offering natural light quality.

The project

This project deals with the design of a new contemporary church in Singapore optimizing the use of daylight (with S. Chaltiel and T. Schroepfer, SUTD).

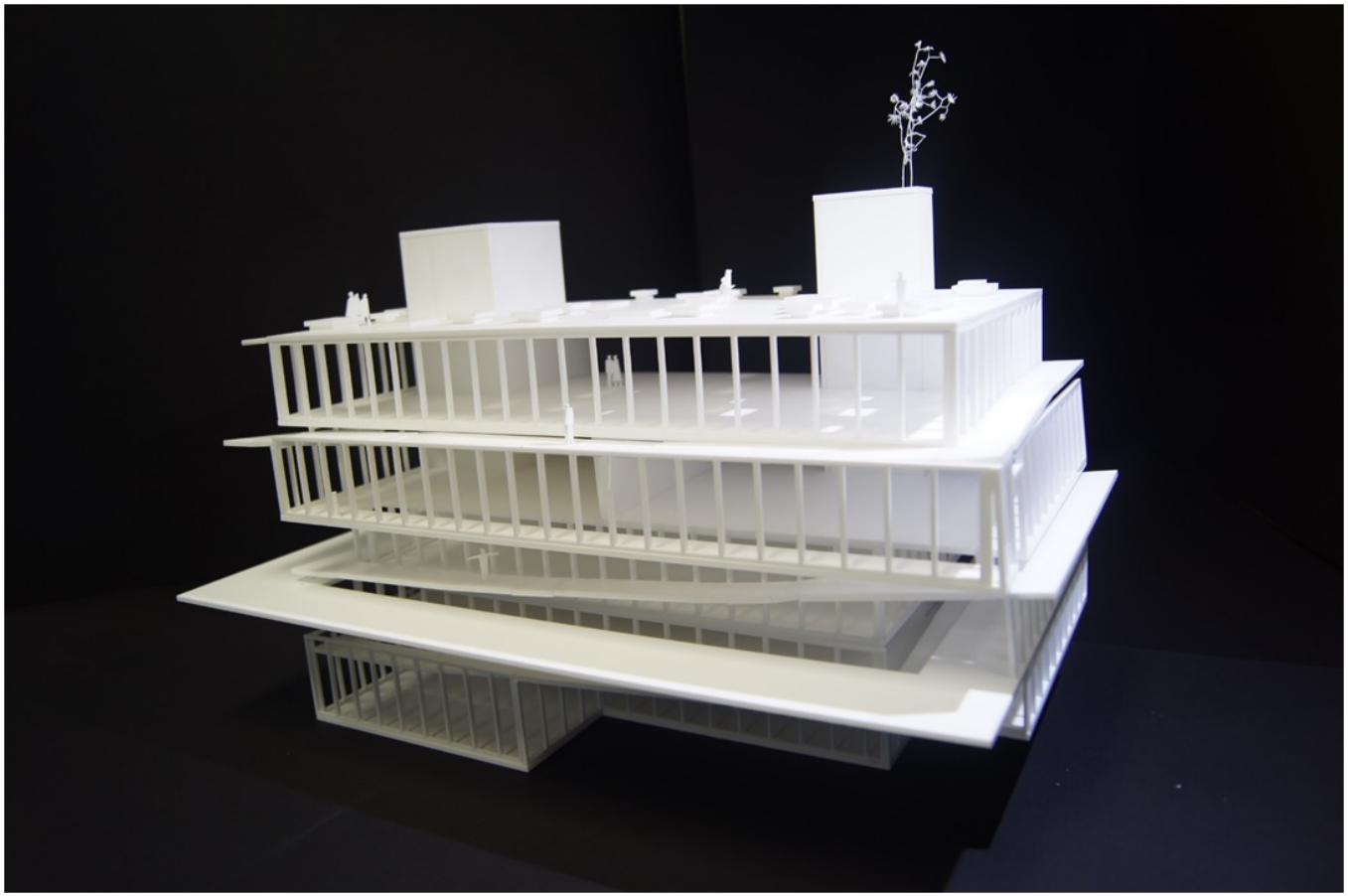
This work is based on the exploration of a performative facade aiming at offering natural light quality.

Facade



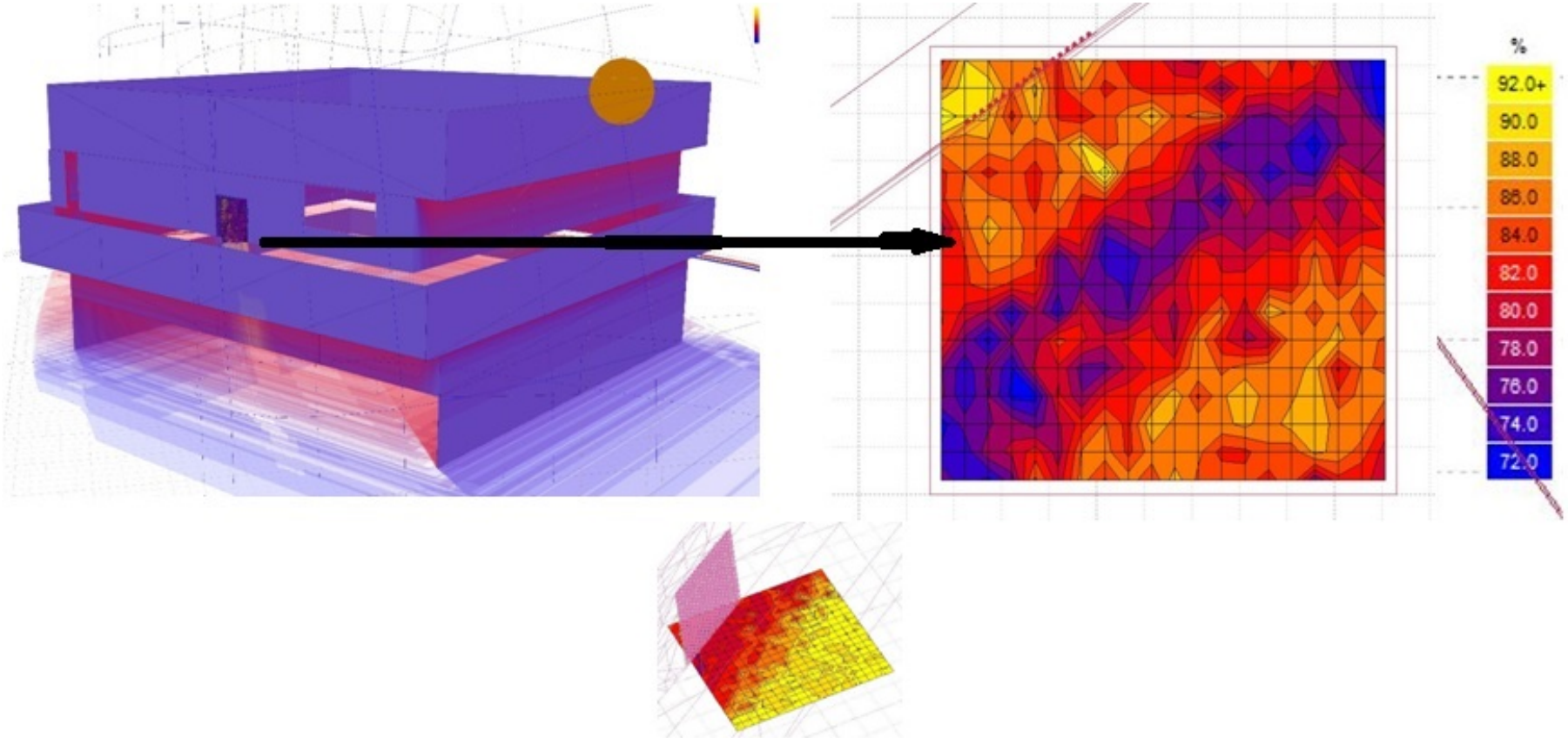
The envelope of the building is a powerful shadowing and filtering light device, depending on the angle of the circled louver.

The church



Simulations

Simulation of the lighthing on the floor of a room (using Ecotect (Autodesk)).



Objective functions

How can we decide about the quality of a solution given by the simulator?

Objective functions

How can we decide about the quality of a solution given by the simulator?

Different criteria

- Satisfaction to maximize
- Uniformity (neighbors) to minimize
- Uniformity (mean) to minimize
- Uniformity (variance) to minimize

Satisfaction - 1

For each area p of the building that we consider (e.g., a room), we define a satisfaction function $s_p(x) : \mathbb{R}^+ \to [0, 1]$.

Satisfaction - 1

For each area p of the building that we consider (e.g., a room), we define a satisfaction function $s_p(x) : \mathbb{R}^+ \to [0, 1]$.

- x_i: value of illuminance (expressed in lux) at a point $i \in p$
- $s_p(x_i)$: the corresponding satisfaction.

Satisfaction - 1

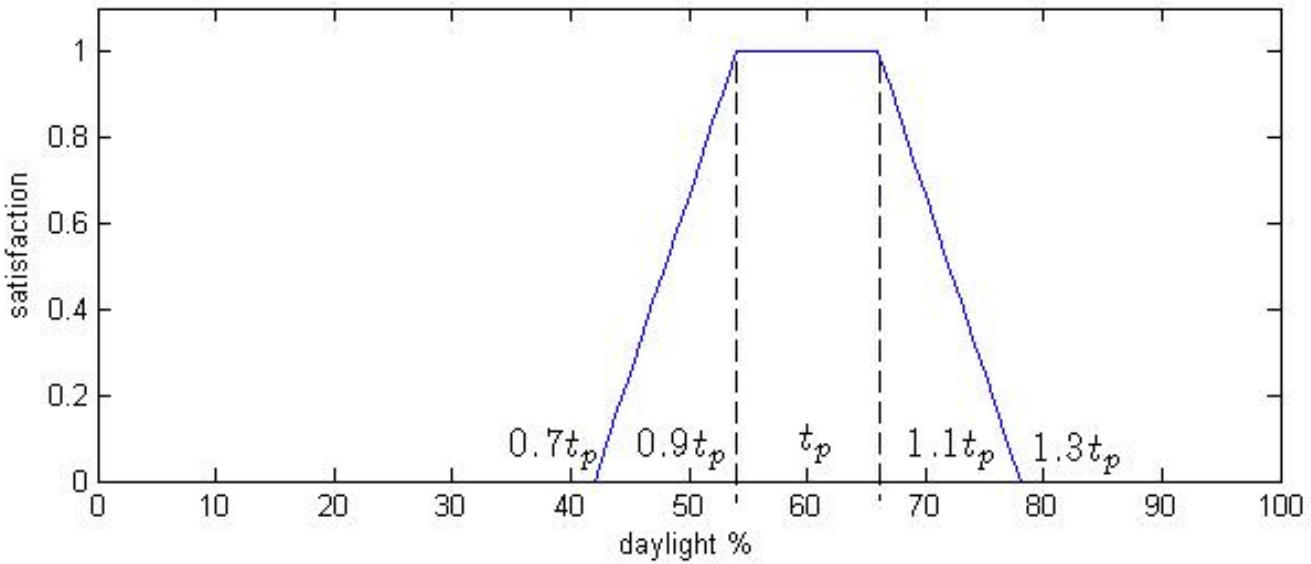
For each area p of the building that we consider (e.g., a room), we define a satisfaction function $s_p(x) : \mathbb{R}^+ \to [0, 1]$.

- x_i: value of illuminance (expressed in lux) at a point $i \in p$
- $s_p(x_i)$: the corresponding satisfaction.

Given a target value t_p of illuminance for the area p, the satisfaction is equal to 1 (i.e., 100%) in the range $t_p \pm 10\%$, 0 if more than or less than 30% from t_p. For the other points, the satisfaction is defined by a linear function.

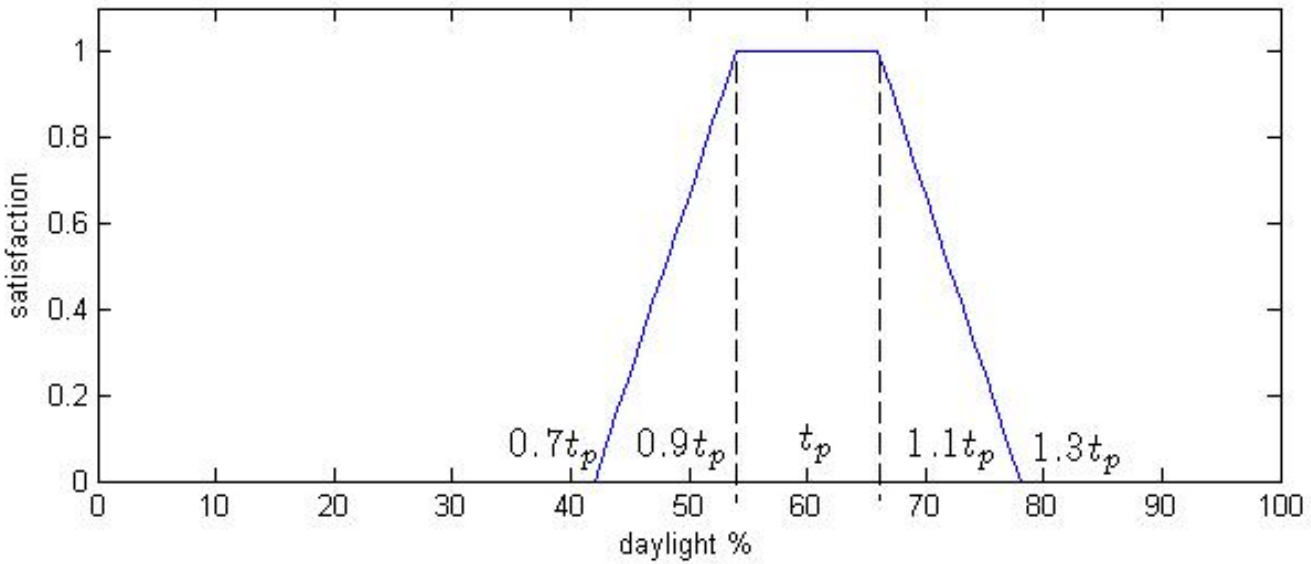
Satisfaction - 2

Example, $t_p = 60\%$



Satisfaction - 2

Example, $t_p = 60\%$



The average satisfaction over an area is the performance measure:

$$S(p) := \frac{1}{|p|} \sum_{i \in p} s_p(x_i).$$

Uniformity (neighbors)

In most situations we want to avoid having large differences of illuminance values over a small area. For example, in the same room having both very bright and very dark areas is usually undesirable.

Uniformity (neighbors)

In most situations we want to avoid having large differences of illuminance values over a small area. For example, in the same room having both very bright and very dark areas is usually undesirable.

We can minimize the average difference between the illuminance value of each point in p and those of its neighbors:

$$U_n(p) = \frac{\sum_{i \in p,\, j \in N_p(i) \,:\, i<j} |x_i - x_j|}{\sum_{i \in p,\, j \in N_p(i) \,:\, i<j} 1},$$

where $N_p(i)$ represents the neighbors of the point i in p.

Uniformity (mean)

We can minimize the average difference between the illuminance value of each point in p and the average illuminance in p ($\bar{x}_p$):

$$U_a(p) = \frac{1}{|p|} \sum_{i \in p} |x_i - \bar{x}_p|.$$

Uniformity (variance)

We can alternatively minimize the variance:

$$U_v(p) = \frac{1}{|p|} \sum_{i \in p} (x_i - \bar{x}_p)^2.$$

Where are we?

1. Preliminaries
2. Radial Basis Function method
   - 1. Selection of k points
   - 2. Computation of the interpolant
   - 3. Selection of next point (x_{k+1})
   - 4. Research questions
3. Application: architectural design
4. Conclusions

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).
- The cyclic strategy for the target value assure convergence.

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).
- The cyclic strategy for the target value assure convergence.

Future work and research questions

- How to solve efficiently the MINLP problems to decide the next evaluation point?

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).
- The cyclic strategy for the target value assure convergence.

Future work and research questions

- How to solve efficiently the MINLP problems to decide the next evaluation point?
- Which objective function is the best? (Ask to architects).

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).
- The cyclic strategy for the target value assure convergence.

Future work and research questions

- How to solve efficiently the MINLP problems to decide the next evaluation point?
- Which objective function is the best? (Ask to architects).
- Extend the problem (e.g., ventilation and temperature).

Key points

Final considerations

- The RBF method gives an indication about where to evaluate the function through the oracle (i.e., minimize bumpiness).
- Cross validation can be achieved solving LP (should be fast!).
- The cyclic strategy for the target value assure convergence.

Future work and research questions

- How to solve efficiently the MINLP problems to decide the next evaluation point?
- Which objective function is the best? (Ask to architects).
- Extend the problem (e.g., ventilation and temperature).
- Create a software to integrate to simulator.