Adaptively Tuned Particle Swarm Optimization for Spatial Design

Matthew Simpson

Department of Statistics, University of Missouri
SAS Institute, Inc.

August 3, 2016

Joint work with Christopher Wikle and Scott H. Holan

Research supported by the NSF-Census Research Network

Overview of the Talk

1. What is particle swarm optimization (PSO)? (Blum and Li, 2008; Clerc, 2010, 2012)

2. New adaptively-tuned PSO algorithms.

3. Using (adaptively-tuned) PSO for spatial design.

4. Example adding locations to an existing monitoring network.

Particle Swarm Optimization — Intuition

Put a "swarm" of particles in the search space:

> Don't search alone, pay attention to what your neighbors are doing!

Particle Swarm Optimization — Intuition

Put a "swarm" of particles in the search space:

> Don't search alone, pay attention to what your neighbors are doing!

Click!

Best for complex objective functions which are cheap to compute, and when near-optimal solutions are useful.

Particle Swarm Optimization

Goal: minimize some objective function $Q(\boldsymbol{\theta}) : \mathbb{R}^D \to \mathbb{R}$.

Populate $\mathbb{R}^D$ with n particles. Define particle i in period k by:

- a location $\boldsymbol{\theta}_i(k) \in \mathbb{R}^D$;
- a velocity $\boldsymbol{v}_i(k) \in \mathbb{R}^D$;
- a personal best location $\boldsymbol{p}_i(k) \in \mathbb{R}^D$;
- a neighborhood (group) best location $\boldsymbol{g}_i(k) \in \mathbb{R}^D$.

Particle Swarm Optimization

Goal: minimize some objective function $Q(\boldsymbol{\theta}) : \mathbb{R}^D \to \mathbb{R}$.

Populate $\mathbb{R}^D$ with n particles. Define particle i in period k by:

- a location $\boldsymbol{\theta}_i(k) \in \mathbb{R}^D$;
- a velocity $\boldsymbol{v}_i(k) \in \mathbb{R}^D$;
- a personal best location $\boldsymbol{p}_i(k) \in \mathbb{R}^D$;
- a neighborhood (group) best location $\boldsymbol{g}_i(k) \in \mathbb{R}^D$.

Basic PSO: update particle i from k to $k+1$ via:

- For $j = 1, 2, \ldots, D$:

$$\begin{aligned} v_{ij}(k+1) &= \omega v_{ij}(k) + \mathrm{U}(0, \phi_1) \times \{p_{ij}(k) - \theta_{ij}(k)\} \\ &\qquad + \mathrm{U}(0, \phi_2) \times \{g_{ij}(k) - \theta_{ij}(k)\} \\ &= \text{inertia} + \text{cognitive} + \text{social}, \\ \theta_{ij}(k+1) &= \theta_{ij}(k) + v_{ij}(k+1), \end{aligned}$$

- Then update personal and group best locations.

PSO — Parameters

Inertia parameter: ω.

- Controls the particle's tendency to keep moving in the same direction.

Cognitive correction factor: ϕ_1.

- Controls the particle's tendency to move toward its personal best.

Social correction factor: ϕ_2.

- Controls the particle's tendency to move toward its neighborhood best.

Default choices:

- $\omega = 0.7298$, $\phi_1 = \phi_2 = 1.496$ (Clerc and Kennedy, 2002).
- $\omega = 1/(2\ln 2) \approx 0.721$, $\phi_1 = \phi_2 = 1/2 + \ln 2 \approx 1.193$ (Clerc, 2006).

PSO — Neighborhood Topologies

Sometimes it is useful to restrict the flow of information across the swarm — e.g. complicated objective functions with many local optima.

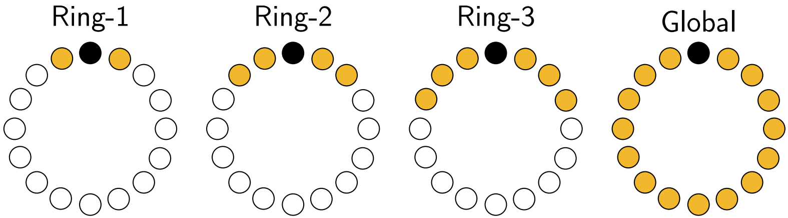
Particles are only informed by their *neighbors*.

PSO — Neighborhood Topologies

Sometimes it is useful to restrict the flow of information across the swarm — e.g. complicated objective functions with many local optima.

Particles are only informed by their *neighbors*.

Easy to visualize example: Ring-k neighborhood topology.



Each particle is informed by k neighbors to the left and k to the right, *no matter where they are in the search space*.

Stochastic Star Topology, and Other Bells and Whistles

We use the stochastic star neighborhood topology (Miranda et al., 2008).

- Each particle informs itself and m random particles.
  $\rightarrow$ sampled with replacement once during initialization.
- On average each particle is informed by m particles.
- A small number of particles will be informed by many particles.

Stochastic Star Topology, and Other Bells and Whistles

We use the stochastic star neighborhood topology (Miranda et al., 2008).

- Each particle informs itself and m random particles.
  $\rightarrow$ sampled with replacement once during initialization.
- On average each particle is informed by m particles.
- A small number of particles will be informed by many particles.

Many variants available (Clerc, 2012), (Simpson et al., 2017, appendix).

- Handling search space constraints.
- Coordinate free velocity updates.
- Parallelization.
- Asynchronous updates.
- Redraw neighborhoods.

Bare Bones PSO (BBPSO)

Developed by Kennedy (2003).

Strips out the velocity term:

$$\theta_{ij}(k+1) \sim \mathrm{N}\left(\frac{p_{ij}(k) + g_{ij}(k)}{2}, |p_{ij}(k) - g_{ij}(k)|^2\right).$$

Mimics the behavior of standard PSO.

Easier to analyze, but tends to perform worse.

Adaptively Tuned BBPSO

Add flexibility to the scale parameter:

$$\theta_{ij}(k+1) \sim \mathrm{T}_{df}\left(\frac{p_{ij}(k) + g_{ij}(k)}{2}, \sigma^2(k)|p_{ij}(k) - g_{ij}(k)|^2\right).$$

with e.g. $df = 1$ by default.

- Larger $\sigma^2(k)$: more exploration.
- Smaller $\sigma^2(k)$: more exploitation.

How to choose $\sigma^2(k)$'s progression?

Adaptively Tuned BBPSO

Add flexibility to the scale parameter:

$$\theta_{ij}(k+1) \sim \mathrm{T}_{df}\left(\frac{p_{ij}(k) + g_{ij}(k)}{2}, \sigma^2(k)|p_{ij}(k) - g_{ij}(k)|^2\right).$$

with e.g. $df = 1$ by default.

- Larger $\sigma^2(k)$: more exploration.
- Smaller $\sigma^2(k)$: more exploitation.

How to choose $\sigma^2(k)$'s progression?

Analogy with adaptively tuned random walk Metropolis.
(Andrieu and Thoms, 2008)

Adaptively Tuned BBPSO — $\sigma^2(k)$'s progression

Define the improvement rate of the swarm in period k:

$$R(k) = \text{proportion of particles that improved on their personal best last period.}$$

Adaptively Tuned BBPSO — $\sigma^2(k)$'s progression

Define the improvement rate of the swarm in period k:

$$R(k) = \text{proportion of particles that improved on their personal best last period.}$$

Let R^* denote a target improvement rate, and
c denote an adjustment factor.

Update $\sigma^2(k)$ via:

$$\log \sigma^2(k+1) = \log \sigma^2(k) + c\{R(k+1) - R^*\}$$

Adaptively Tuned BBPSO — $\sigma^2(k)$'s progression

Define the improvement rate of the swarm in period k:

$$R(k) = \text{proportion of particles that improved on their personal best last period.}$$

Let R^* denote a target improvement rate, and
c denote an adjustment factor.

Update $\sigma^2(k)$ via:

$$\log \sigma^2(k+1) = \log \sigma^2(k) + c\{R(k+1) - R^*\}$$

Larger c, faster changes to $\sigma^2(k)$.

Adaptively Tuned BBPSO — $\sigma^2(k)$'s progression

Define the improvement rate of the swarm in period k:

$$R(k) = \text{proportion of particles that improved on their personal best last period.}$$

Let R^* denote a target improvement rate, and
c denote an adjustment factor.

Update $\sigma^2(k)$ via:

$$\log \sigma^2(k+1) = \log \sigma^2(k) + c\{R(k+1) - R^*\}$$

Larger c, faster changes to $\sigma^2(k)$.

Defaults: $R^* \in [0.3, 0.5]$, $c = 0.1$.

Adaptively Tuning PSO

In PSO larger $\omega \implies$ more exploration, smaller $\omega \implies$ more exploitation.

Adaptively Tuning PSO

In PSO larger $\omega \implies$ more exploration, smaller $\omega \implies$ more exploitation.

Idea: deterministic inertia PSO (DI-PSO)
$\rightarrow$ slowly decrease $\omega(k)$ over time (Eberhart and Shi, 2000).

- Hard to set appropriately for any given problem.

Adaptively Tuning PSO

In PSO larger $\omega \implies$ more exploration, smaller $\omega \implies$ more exploitation.

Idea: deterministic inertia PSO (DI-PSO)
$\rightarrow$ slowly decrease $\omega(k)$ over time (Eberhart and Shi, 2000).

- Hard to set appropriately for any given problem.

AT-PSO: tune $\omega(k)$ just like $\sigma^2(k)$ in AT-BBPSO.

$$\log \omega(k+1) = \log \omega(k) + c\{R(k+1) - R^*\}.$$

Adaptively Tuning PSO

In PSO larger $\omega \implies$ more exploration, smaller $\omega \implies$ more exploitation.

Idea: deterministic inertia PSO (DI-PSO)
$\rightarrow$ slowly decrease $\omega(k)$ over time (Eberhart and Shi, 2000).

- Hard to set appropriately for any given problem.

AT-PSO: tune $\omega(k)$ just like $\sigma^2(k)$ in AT-BBPSO.

$$\log \omega(k+1) = \log \omega(k) + c\{R(k+1) - R^*\}.$$

Same defaults: $R^* \in [0.3, 0.5]$, $c = 0.1$.

Adaptively Tuning PSO

In PSO larger $\omega \implies$ more exploration, smaller $\omega \implies$ more exploitation.

Idea: deterministic inertia PSO (DI-PSO)
$\rightarrow$ slowly decrease $\omega(k)$ over time (Eberhart and Shi, 2000).

- Hard to set appropriately for any given problem.

AT-PSO: tune $\omega(k)$ just like $\sigma^2(k)$ in AT-BBPSO.

$$\log \omega(k+1) = \log \omega(k) + c\{R(k+1) - R^*\}.$$

Same defaults: $R^* \in [0.3, 0.5]$, $c = 0.1$.

Similar PSO algorithm in spirit: Zhang et al. (2003).

- ω is constant while ϕ_1 and ϕ_2 vary across time *and particle*.
- Can't use the same method to adapt ϕ_1 and ϕ_2.

Comparing AT-PSO/BBPSO to PSO/BBPSO

Tuning $\omega(k)/\sigma^2(k)$ allows the swarm to adjust the exploration / exploitation tradeoff based on local conditions.

- This has a tendency to speed up convergence.
- ...but convergence may be premature in multi-modal problems.

Comparing AT-PSO/BBPSO to PSO/BBPSO

Tuning $\omega(k)/\sigma^2(k)$ allows the swarm to adjust the exploration / exploitation tradeoff based on local conditions.

- This has a tendency to speed up convergence.
- ...but convergence may be premature in multi-modal problems.

Overview of results from a simulation study:

- BBPSO tends to perform poorly, but AT-BBPSO performs quite well.
- AT-BBPSO often the best for complex objective functions.
  $\rightarrow$ The T_{df} makes it fairly robust to many local optima.

Comparing AT-PSO/BBPSO to PSO/BBPSO

Tuning $\omega(k)/\sigma^2(k)$ allows the swarm to adjust the exploration / exploitation tradeoff based on local conditions.

- This has a tendency to speed up convergence.
- ...but convergence may be premature in multi-modal problems.

Overview of results from a simulation study:

- BBPSO tends to perform poorly, but AT-BBPSO performs quite well.
- AT-BBPSO often the best for complex objective functions.
  $\rightarrow$ The T_{df} makes it fairly robust to many local optima.
- AT-PSO performs better than PSO on "hard enough" problems...

Comparing AT-PSO/BBPSO to PSO/BBPSO

Tuning $\omega(k)/\sigma^2(k)$ allows the swarm to adjust the exploration / exploitation tradeoff based on local conditions.

- This has a tendency to speed up convergence.
- ...but convergence may be premature in multi-modal problems.

Overview of results from a simulation study:

- BBPSO tends to perform poorly, but AT-BBPSO performs quite well.
- AT-BBPSO often the best for complex objective functions.
  $\rightarrow$ The T_{df} makes it fairly robust to many local optima.
- AT-PSO performs better than PSO on “hard enough” problems...
- ...but has trouble with many local optima.

Spatial Design — Problem Setup

Goal: want to learn about the spatial field $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^2$.

Spatial Design — Problem Setup

Goal: want to learn about the spatial field $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^2$.

We observe a noisy signal of $Y(\boldsymbol{u})$ at N_s fixed monitoring locations, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_s}$, and assume:

$$Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$$

for all $\boldsymbol{u} \in \mathcal{D}$, and $\varepsilon(\boldsymbol{u}) \overset{iid}{\sim} \mathrm{N}(0, \tau^2)$.

Spatial Design — Problem Setup

Goal: want to learn about the spatial field $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^2$.

We observe a noisy signal of $Y(\boldsymbol{u})$ at N_s fixed monitoring locations, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_s}$, and assume:

$$Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$$

for all $\boldsymbol{u} \in \mathcal{D}$, and $\varepsilon(\boldsymbol{u}) \overset{iid}{\sim} \mathrm{N}(0, \tau^2)$.

Assume

$$Y(\boldsymbol{u}) \sim \mathrm{GP}(\boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}, C_{\boldsymbol{\phi}}(\cdot, \cdot))$$

where $\boldsymbol{x}(\boldsymbol{u})$ is a vector of covariates known at all locations $\boldsymbol{u} \in \mathcal{D}$.

Spatial Design — Problem Setup

Goal: want to learn about the spatial field $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^2$.

We observe a noisy signal of $Y(\boldsymbol{u})$ at N_s fixed monitoring locations, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_s}$, and assume:

$$Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$$

for all $\boldsymbol{u} \in \mathcal{D}$, and $\varepsilon(\boldsymbol{u}) \overset{iid}{\sim} \mathrm{N}(0, \tau^2)$.

Assume

$$Y(\boldsymbol{u}) \sim \mathrm{GP}(\boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}, C_{\phi}(\cdot, \cdot))$$

where $\boldsymbol{x}(\boldsymbol{u})$ is a vector of covariates known at all locations $\boldsymbol{u} \in \mathcal{D}$.

Where should we put an additional N_d monitoring locations, $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{N_d}$?

Spatial Design — Problem Setup

Goal: want to learn about the spatial field $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subset \mathbb{R}^2$.

We observe a noisy signal of $Y(\boldsymbol{u})$ at N_s fixed monitoring locations, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_s}$, and assume:

$$Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$$

for all $\boldsymbol{u} \in \mathcal{D}$, and $\varepsilon(\boldsymbol{u}) \overset{iid}{\sim} \mathrm{N}(0, \tau^2)$.

Assume

$$Y(\boldsymbol{u}) \sim \mathrm{GP}(\boldsymbol{x}(\boldsymbol{u})'\beta, C_\phi(\cdot,\cdot))$$

where $\boldsymbol{x}(\boldsymbol{u})$ is a vector of covariates known at all locations $\boldsymbol{u} \in \mathcal{D}$.

Where should we put an additional N_d monitoring locations, $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{N_d}$?

What if $\boldsymbol{\theta} = (\tau^2, \boldsymbol{\beta}, \phi)$ is unknown?.

Spatial Design — MSPE and Kriging

Sensible goal: choose new locations to minimize MSPE.

Spatial Design — MSPE and Kriging

Sensible goal: choose new locations to minimize MSPE.

When τ^2 and ϕ are known, the universal kriging predictor is:

$$\widehat{Y}_{uk}(\boldsymbol{u};\boldsymbol{d}) = \boldsymbol{x}(\boldsymbol{u})'\widehat{\boldsymbol{\beta}}_{gls} + \boldsymbol{c}_Y(\boldsymbol{u})'\boldsymbol{C}_Z^{-1}(\boldsymbol{Z} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}_{gls})$$

(Cressie and Wikle, 2011) where

$$\begin{aligned}
\boldsymbol{X} &= (\boldsymbol{x}(\boldsymbol{s}_1), \ldots, \boldsymbol{x}(\boldsymbol{s}_{N_s}), \boldsymbol{x}(\boldsymbol{d}_1), \ldots, \boldsymbol{x}(\boldsymbol{d}_{N_d}))', \\
\boldsymbol{Y} &= (Y(\boldsymbol{s}_1), \ldots, Y(\boldsymbol{s}_{N_s}), Y(\boldsymbol{d}_1), \ldots, Y(\boldsymbol{d}_{N_d}))', \\
\boldsymbol{Z} &= (Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_{N_s}), Z(\boldsymbol{d}_1), \ldots, Z(\boldsymbol{d}_{N_d}))', \\
\boldsymbol{C}_Z &= \mathrm{Cov}(\boldsymbol{Z}) = \tau^2\mathbf{I} + \mathrm{Cov}(\boldsymbol{Y}), \\
\boldsymbol{c}_Y &= \mathrm{Cov}(Y(\boldsymbol{u}), \boldsymbol{Y}), \\
\widehat{\boldsymbol{\beta}}_{gls} &= (\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{Z}.
\end{aligned}$$

Spatial Design — Kriging Variances

Kriging MSPE: $\mathrm{E}\left\{Y(\boldsymbol{u}) - \widehat{Y}_{uk}(\boldsymbol{u})\right\}^2 = \sigma^2_{uk}(\boldsymbol{u};\boldsymbol{d}) =$

$$C_\phi(\boldsymbol{u},\boldsymbol{u}) - \boldsymbol{c}_Y(\boldsymbol{u})'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})$$
$$+ \{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}'(\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X})^{-1}\{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}$$

Spatial Design — Kriging Variances

Kriging MSPE: $\quad \mathrm{E}\left\{Y(\boldsymbol{u}) - \widehat{Y}_{uk}(\boldsymbol{u})\right\}^2 = \sigma^2_{uk}(\boldsymbol{u};\boldsymbol{d}) =$

$$
\begin{aligned}
&C_\phi(\boldsymbol{u},\boldsymbol{u}) - \boldsymbol{c}_Y(\boldsymbol{u})'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u}) \\
&+ \{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}'(\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X})^{-1}\{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}
\end{aligned}
$$

What about when τ^2 and ϕ are estimated?

Kriging MSPE: $\mathrm{E}\left\{Y(\boldsymbol{u}) - \widehat{Y}_{uk}(\boldsymbol{u})\right\}^2 = \sigma^2_{uk}(\boldsymbol{u};\boldsymbol{d}) =$

$$C_\phi(\boldsymbol{u},\boldsymbol{u}) - \boldsymbol{c}_Y(\boldsymbol{u})'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u}) \\ + \{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}'(\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X})^{-1}\{\boldsymbol{x}(\boldsymbol{u}) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{u})\}$$

What about when τ^2 and ϕ are estimated?
$\rightarrow$ Parameter uncertainty universal kriging MSPE:

$$\mathrm{E}\left\{Y(\boldsymbol{u}) - \widehat{Y}_{uk}(\boldsymbol{u})\right\}^2 \approx \sigma^2_{puk}(\boldsymbol{u};\boldsymbol{d},\widehat{\boldsymbol{\theta}}) = \sigma^2_{uk}(\boldsymbol{u};\boldsymbol{d},\widehat{\boldsymbol{\theta}}) + \text{ stuff},$$

depending on the MLE ($\widehat{\boldsymbol{\theta}}$), FI matrix, and gradient of $\widehat{Y}_{uk}$ wrt $\boldsymbol{\theta}$.
(Zimmerman and Cressie, 1992; Abt, 1999)

Spatial Design — Design Criteria

Ideal design criteria: choose design points to minimize...

- Mean/total MSPE: $\overline{Q}_{puk}(\boldsymbol{d}) = \int_{\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u}) d\boldsymbol{u}$
- Maximum MSPE: $Q^*_{puk}(\boldsymbol{d}) = \max_{\boldsymbol{u} \in \mathcal{D}} \sigma^2_{puk}(\boldsymbol{u})$

Ideal design criteria: choose design points to minimize...

- Mean/total MSPE: $\overline{Q}_{puk}(\boldsymbol{d}) = \int_{\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u}) d\boldsymbol{u}$
- Maximum MSPE: $Q^*_{puk}(\boldsymbol{d}) = \max_{\boldsymbol{u}\in\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u})$

This is computationally infeasible.

Spatial Design — Design Criteria

Ideal design criteria: choose design points to minimize...

- Mean/total MSPE: $\overline{Q}_{puk}(\boldsymbol{d}) = \int_{\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u})d\boldsymbol{u}$
- Maximum MSPE: $Q^*_{puk}(\boldsymbol{d}) = \max_{\boldsymbol{u}\in\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u})$

This is computationally infeasible.

Realistic criteria: approximate with a grid of target points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_t}$:

- Minimize $\overline{Q}_{puk}(\boldsymbol{d}) = \sum_{i=1}^{N_t} \sigma^2_{puk}(\boldsymbol{r}_i)$
- Minimize $Q^*_{puk}(\boldsymbol{d}) = \max_{i=1,2,\ldots,N_t} \sigma^2_{puk}(\boldsymbol{r}_i)$

Ideal design criteria: choose design points to minimize...

- Mean/total MSPE: $\overline{Q}_{puk}(\boldsymbol{d}) = \int_{\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u}) d\boldsymbol{u}$
- Maximum MSPE: $Q^*_{puk}(\boldsymbol{d}) = \max_{\boldsymbol{u}\in\mathcal{D}} \sigma^2_{puk}(\boldsymbol{u})$

This is computationally infeasible.

Realistic criteria: approximate with a grid of target points $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_t}$:

- Minimize $\overline{Q}_{puk}(\boldsymbol{d}) = \sum_{i=1}^{N_t} \sigma^2_{puk}(\boldsymbol{r}_i)$
- Minimize $Q^*_{puk}(\boldsymbol{d}) = \max_{i=1,2,\ldots,N_t} \sigma^2_{puk}(\boldsymbol{r}_i)$

Optimzation for Spatial Design

The spatial design optimization problem cannot be solved in closed form.

Optimzation for Spatial Design

The spatial design optimization problem cannot be solved in closed form.

Usual solution: grid up the space and use an exchange algorithm. (Nychka and Saltzman, 1998; Wikle and Royle, 1999, 2005)

Optimzation for Spatial Design

The spatial design optimization problem cannot be solved in closed form.

Usual solution: grid up the space and use an exchange algorithm. (Nychka and Saltzman, 1998; Wikle and Royle, 1999, 2005)

In principle, *any* location is a valid design point, not just the grid.
$\Longrightarrow$ why not use PSO?

Optimzation for Spatial Design

The spatial design optimization problem cannot be solved in closed form.

Usual solution: grid up the space and use an exchange algorithm. (Nychka and Saltzman, 1998; Wikle and Royle, 1999, 2005)

In principle, *any* location is a valid design point, not just the grid.
$\Longrightarrow$ why not use PSO?

- Near-optimal solutions are nearly as valuable as optimal solutions.
- Objective function is cheap in universal kriging.
- More expensive in kriging with parameter uncertainty, but doable.
- Highly multi-modal objective function (e.g. switch two locations).

Optimzation for Spatial Design

The spatial design optimization problem cannot be solved in closed form.

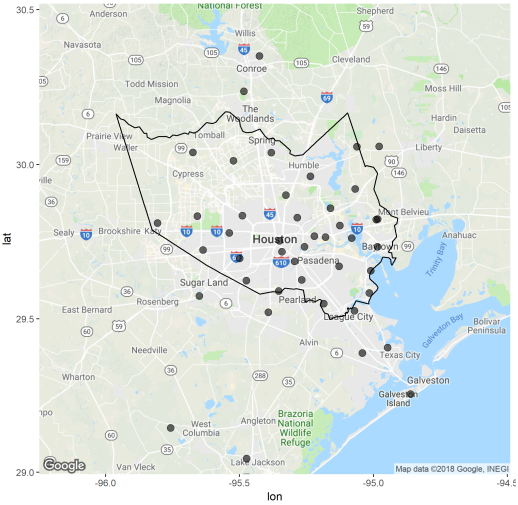
Usual solution: grid up the space and use an exchange algorithm. (Nychka and Saltzman, 1998; Wikle and Royle, 1999, 2005)

In principle, *any* location is a valid design point, not just the grid.
$\implies$ why not use PSO?

- Near-optimal solutions are nearly as valuable as optimal solutions.
- Objective function is cheap in universal kriging.
- More expensive in kriging with parameter uncertainty, but doable.
- Highly multi-modal objective function (e.g. switch two locations).

Genetic algorithms also reasonable, e.g. Hamada et al. (2001).

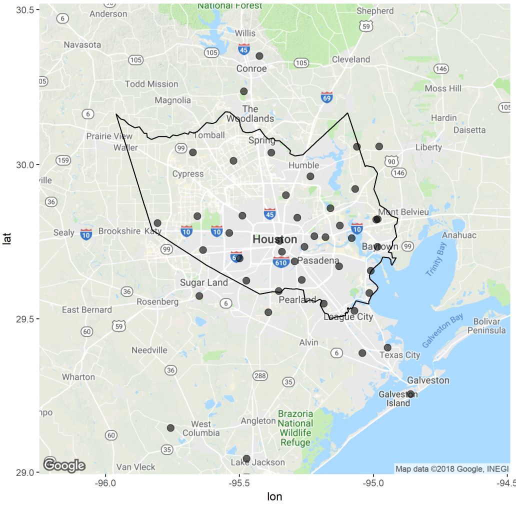
Example: Ozone Monitoring in Harris County, TX



Map via ggmap (Kahle and Wickham, 2013).

- Ozone concentration is associated with increased risk of cardiac arrest (Ensor et al., 2013).

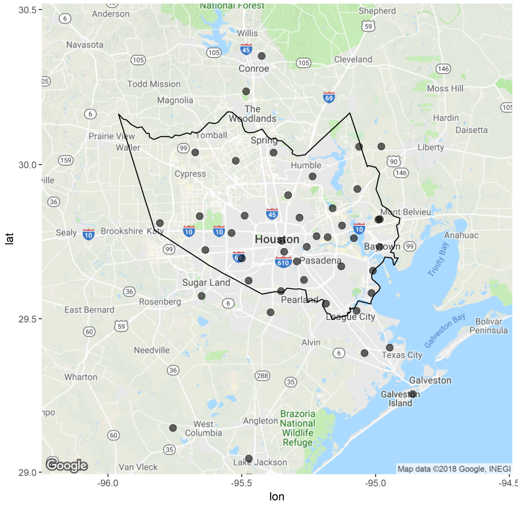
Example: Ozone Monitoring in Harris County, TX



Map via ggmap (Kahle and Wickham, 2013).

- Ozone concentration is associated with increased risk of cardiac arrest (Ensor et al., 2013).
- In August 2016, there were 44 active monitoring locations near Houston, TX.

Example: Ozone Monitoring in Harris County, TX



Map via ggmap (Kahle and Wickham, 2013).

- Ozone concentration is associated with increased risk of cardiac arrest (Ensor et al., 2013).
- In August 2016, there were 44 active monitoring locations near Houston, TX.
- Harris County, TX, contains 33 of these locations.

Hypothetical Design Problem and Data

Want to learn more about ozone concentrations in Harris County.

Where to put 100 new monitoring locations within the county?

Note: nearby locations outside of the county still useful for estimation.

Hypothetical Design Problem and Data

Want to learn more about ozone concentrations in Harris County.

Where to put 100 new monitoring locations within the county?

Note: nearby locations outside of the county still useful for estimation.

Data from the Texas Commission on Environmental Quality (TCEQ)

- Monitoring locations measure several air quality indicators.
- Ozone: daily maximum eight-hour ozone concentration (DM8) in parts per billion.
  $\rightarrow$ maximum of all contiguous 8-hour means for that day.
- Some locations have missing data.

Model and Design Criteria

Model:

- Linear mean function in spatial coordinates: $\boldsymbol{x}(\boldsymbol{u})' = (u_1, u_2)$.

Model and Design Criteria

Model:

- Linear mean function in spatial coordinates: $\boldsymbol{x}(\boldsymbol{u})' = (u_1, u_2)$.
- Exponential covariance function:

$$C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\psi)$$

Model and Design Criteria

Model:

- Linear mean function in spatial coordinates: $\boldsymbol{x}(\boldsymbol{u})' = (u_1, u_2)$.
- Exponential covariance function:

$$C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\psi)$$

- Estimate $\boldsymbol{\theta} = (\tau^2, \beta_0, \beta_1, \beta_2, \sigma^2, \psi)$ via maximum likelihood.

Model and Design Criteria

Model:

- Linear mean function in spatial coordinates: $\boldsymbol{x}(\boldsymbol{u})' = (u_1, u_2)$.
- Exponential covariance function:

$$C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\psi)$$

- Estimate $\boldsymbol{\theta} = (\tau^2, \beta_0, \beta_1, \beta_2, \sigma^2, \psi)$ via maximum likelihood.

Design Criteria:

- Mean MSPE w/ parameter uncertainty $\overline{Q}_{puk}(\boldsymbol{d})$.
- Maximum MSPE w/ parameter uncertainty $Q^*_{puk}(\boldsymbol{d})$.

Model and Design Criteria

Model:

- Linear mean function in spatial coordinates: $\boldsymbol{x}(\boldsymbol{u})' = (u_1, u_2)$.
- Exponential covariance function:

$$C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\psi)$$

- Estimate $\boldsymbol{\theta} = (\tau^2, \beta_0, \beta_1, \beta_2, \sigma^2, \psi)$ via maximum likelihood.

Design Criteria:

- Mean MSPE w/ parameter uncertainty $\overline{Q}_{puk}(\boldsymbol{d})$.
- Maximum MSPE w/ parameter uncertainty $Q^*_{puk}(\boldsymbol{d})$.
- Approximate each with a grid of 1229 points in Harris County.

Algorithm	$\overline{Q}_{puk}$	Q^*_{puk}
Uniform	16.40	26.80
PSO1	**14.40**	**20.63**
PSO2	14.45	**21.03**
PSO1-CF	15.53	23.54
PSO2-CF	15.77	23.16
AT1-PSO1	**14.38**	**20.57**
AT1-PSO2	14.56	23.18
AT1-PSO1-CF	15.96	23.33
AT1-PSO2-CF	15.60	24.02
AT2-PSO1	**14.42**	**21.13**
AT2-PSO2	**14.32**	22.11
AT2-PSO1-CF	15.85	24.00
AT2-PSO2-CF	15.95	23.63
AT1-BBPSO	14.53	22.28
AT1-BBPSOxp	15.87	22.19
AT1-BBPSO-CF	14.65	21.33
AT1-BBPSOxp-CF	14.84	22.34
AT2-BBPSO	14.65	23.49
AT2-BBPSOxp	15.21	23.25
AT2-BBPSO-CF	14.63	21.92
AT2-BBPSOxp-CF	14.52	22.76
GA-11	**14.40**	21.19
GA-21	15.20	23.21
GA-12	14.45	**20.84**
GA-22	15.26	22.61

Results:

- Uniform: uniformly sample new monitoring locations.
  GA: genetic algorithm.
- Bolded: top 5 for that design criterion (column).

Algorithm	$\overline{Q}_{puk}$	Q^*_{puk}
Uniform	16.40	26.80
PSO1	**14.40**	**20.63**
PSO2	14.45	**21.03**
PSO1-CF	15.53	23.54
PSO2-CF	15.77	23.16
AT1-PSO1	**14.38**	**20.57**
AT1-PSO2	14.56	23.18
AT1-PSO1-CF	15.96	23.33
AT1-PSO2-CF	15.60	24.02
AT2-PSO1	**14.42**	**21.13**
AT2-PSO2	**14.32**	22.11
AT2-PSO1-CF	15.85	24.00
AT2-PSO2-CF	15.95	23.63
AT1-BBPSO	14.53	22.28
AT1-BBPSOxp	15.87	22.19
AT1-BBPSO-CF	14.65	21.33
AT1-BBPSOxp-CF	14.84	22.34
AT2-BBPSO	14.65	23.49
AT2-BBPSOxp	15.21	23.25
AT2-BBPSO-CF	14.63	21.92
AT2-BBPSOxp-CF	14.52	22.76
GA-11	**14.40**	21.19
GA-21	15.20	23.21
GA-12	14.45	**20.84**
GA-22	15.26	22.61

Results:

- Uniform: uniformly sample new monitoring locations.
  GA: genetic algorithm.
- Bolded: top 5 for that design criterion (column).
- Objective function is simple enough that AT-BBPSO variants are less good.

Algorithm	$\overline{Q}_{puk}$	Q^*_{puk}
Uniform	16.40	26.80
PSO1	**14.40**	**20.63**
PSO2	14.45	**21.03**
PSO1-CF	15.53	23.54
PSO2-CF	15.77	23.16
AT1-PSO1	**14.38**	**20.57**
AT1-PSO2	14.56	23.18
AT1-PSO1-CF	15.96	23.33
AT1-PSO2-CF	15.60	24.02
AT2-PSO1	**14.42**	**21.13**
AT2-PSO2	**14.32**	22.11
AT2-PSO1-CF	15.85	24.00
AT2-PSO2-CF	15.95	23.63
AT1-BBPSO	14.53	22.28
AT1-BBPSOxp	15.87	22.19
AT1-BBPSO-CF	14.65	21.33
AT1-BBPSOxp-CF	14.84	22.34
AT2-BBPSO	14.65	23.49
AT2-BBPSOxp	15.21	23.25
AT2-BBPSO-CF	14.63	21.92
AT2-BBPSOxp-CF	14.52	22.76
GA-11	**14.40**	21.19
GA-21	15.20	23.21
GA-12	14.45	**20.84**
GA-22	15.26	22.61

Results:

- Uniform: uniformly sample new monitoring locations.
  GA: genetic algorithm.
- Bolded: top 5 for that design criterion (column).
- Objective function is simple enough that AT-BBPSO variants are less good.
- PSO and AT-PSO variants tend to be the best.

Algorithm	$\overline{Q}_{puk}$	Q^*_{puk}
Uniform	16.40	26.80
PSO1	**14.40**	**20.63**
PSO2	14.45	**21.03**
PSO1-CF	15.53	23.54
PSO2-CF	15.77	23.16
AT1-PSO1	**14.38**	**20.57**
AT1-PSO2	14.56	23.18
AT1-PSO1-CF	15.96	23.33
AT1-PSO2-CF	15.60	24.02
AT2-PSO1	**14.42**	**21.13**
AT2-PSO2	**14.32**	22.11
AT2-PSO1-CF	15.85	24.00
AT2-PSO2-CF	15.95	23.63
AT1-BBPSO	14.53	22.28
AT1-BBPSOxp	15.87	22.19
AT1-BBPSO-CF	14.65	21.33
AT1-BBPSOxp-CF	14.84	22.34
AT2-BBPSO	14.65	23.49
AT2-BBPSOxp	15.21	23.25
AT2-BBPSO-CF	14.63	21.92
AT2-BBPSOxp-CF	14.52	22.76
GA-11	**14.40**	21.19
GA-21	15.20	23.21
GA-12	14.45	**20.84**
GA-22	15.26	22.61

Results:

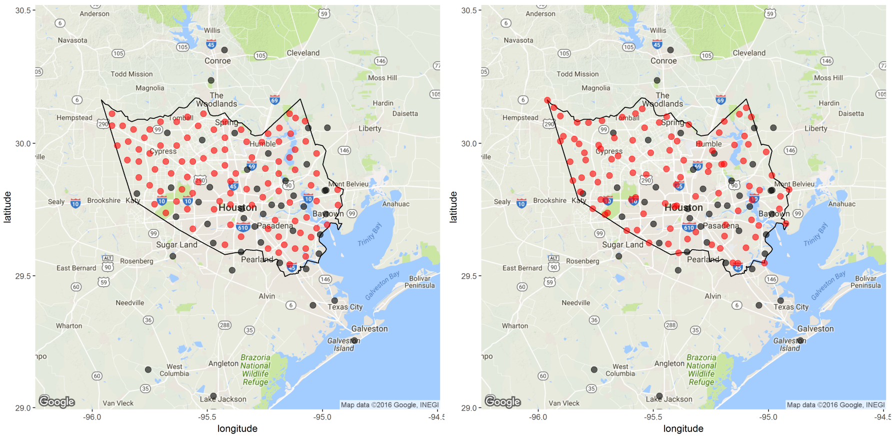
- Uniform: uniformly sample new monitoring locations.
  GA: genetic algorithm.
- Bolded: top 5 for that design criterion (column).
- Objective function is simple enough that AT-BBPSO variants are less good.
- PSO and AT-PSO variants tend to be the best.
- GAs are competitive.

Algorithm	$\overline{Q}_{puk}$	Q^*_{puk}
Uniform	16.40	26.80
PSO1	**14.40**	**20.63**
PSO2	14.45	**21.03**
PSO1-CF	15.53	23.54
PSO2-CF	15.77	23.16
AT1-PSO1	**14.38**	**20.57**
AT1-PSO2	14.56	23.18
AT1-PSO1-CF	15.96	23.33
AT1-PSO2-CF	15.60	24.02
AT2-PSO1	**14.42**	**21.13**
AT2-PSO2	**14.32**	22.11
AT2-PSO1-CF	15.85	24.00
AT2-PSO2-CF	15.95	23.63
AT1-BBPSO	14.53	22.28
AT1-BBPSOxp	15.87	22.19
AT1-BBPSO-CF	14.65	21.33
AT1-BBPSOxp-CF	14.84	22.34
AT2-BBPSO	14.65	23.49
AT2-BBPSOxp	15.21	23.25
AT2-BBPSO-CF	14.63	21.92
AT2-BBPSOxp-CF	14.52	22.76
GA-11	**14.40**	21.19
GA-21	15.20	23.21
GA-12	14.45	**20.84**
GA-22	15.26	22.61

Results:

- Uniform: uniformly sample new monitoring locations.
  GA: genetic algorithm.
- Bolded: top 5 for that design criterion (column).
- Objective function is simple enough that AT-BBPSO variants are less good.
- PSO and AT-PSO variants tend to be the best.
- GAs are competitive.
- With significantly fewer monitoring locations, PSO variants are the best.

Best designs found according to $\overline{Q}_{puk}$ (left) and Q^*_{puk} (right)



Optimal design is highly dependent on the mean function (Zimmerman, 2006).

Background map via ggmap (Kahle and Wickham, 2013).

Conclusions

- Introduced new classes of adaptively tuned PSO and BBPSO algorithms.

Conclusions

- Introduced new classes of adaptively tuned PSO and BBPSO algorithms.
- AT-BBPSO performed well on very difficult problems.

Conclusions

- Introduced new classes of adaptively tuned PSO and BBPSO algorithms.
- AT-BBPSO performed well on very difficult problems.
- AT-PSO performed well on difficult, but not too difficult problems.

Conclusions

- Introduced new classes of adaptively tuned PSO and BBPSO algorithms.
- AT-BBPSO performed well on very difficult problems.
- AT-PSO performed well on difficult, but not too difficult problems.
- For spatial design problems, standard PSO works well.

Conclusions

- Introduced new classes of adaptively tuned PSO and BBPSO algorithms.
- AT-BBPSO performed well on very difficult problems.
- AT-PSO performed well on difficult, but not too difficult problems.
- For spatial design problems, standard PSO works well.
- For large enough spatial design problems, AT-PSO is attractive.

Thank you!

References I

Abt, M. (1999). Estimating the prediction mean squared error in Gaussian stochastic processes with exponential correlation structure. *Scandinavian Journal of Statistics*, 26(4):563–578.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Blum, C. and Li, X. (2008). Swarm intelligence in optimization. In Blum, C. and Merkle, D., editors, *Swarm Intelligence: Introduction and Applications*, pages 43–85. Springer-Verlag, Berlin.

Clerc, M. (2006). Stagnation analysis in particle swarm optimisation or what happens when nothing happens. 17 pages. https://hal.archives-ouvertes.fr/hal-00122031.

Clerc, M. (2010). *Particle swarm optimization*. John Wiley & Sons.

Clerc, M. (2012). Standard particle swarm optimisation. 15 pages. https://hal.archives-ouvertes.fr/hal-00764996.

References II

Clerc, M. and Kennedy, J. (2002). The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73.

Cressie, N. and Wikle, C. K. (2011). *Statistics for Spatio-Temporal Data*. John Wiley & Sons, Hoboken, NJ.

Eberhart, R. C. and Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 84–88. IEEE.

Ensor, K. B., Raun, L. H., and Persse, D. (2013). A case-crossover analysis of out-of-hospital cardiac arrest and air pollution. *Circulation*, 127(11):1192–1199.

Hamada, M., Martz, H., Reese, C., and Wilson, A. (2001). Finding near-optimal Bayesian experimental designs via genetic algorithms. *The American Statistician*, 55(3):175–181.

References III

Kahle, D. and Wickham, H. (2013). ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161.

Kennedy, J. (2003). Bare bones particle swarms. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 80–87. IEEE.

Miranda, V., Keko, H., and Duque, A. J. (2008). Stochastic star communication topology in evolutionary particle swarms (EPSO). *International journal of computational intelligence research*, 4(2):105–116.

Nychka, D. and Saltzman, N. (1998). Design of air-quality monitoring networks. In Nychka, D., Piegorsch, W. W., and Cox, L. H., editors, *Case Studies in Environmental Statistics*, pages 51–76. Springer, New York.

Simpson, M., Wikle, C. K., and Holan, S. H. (2017). Adaptively tuned particle swarm optimization with application to spatial design. *Stat*, 6(1):145–159.

References IV

Wikle, C. K. and Royle, J. A. (1999). Space-time dynamic design of environmental monitoring networks. *Journal of Agricultural, Biological, and Environmental Statistics*, 4(4):489–507.

Wikle, C. K. and Royle, J. A. (2005). Dynamic design of ecological monitoring networks for non-Gaussian spatio-temporal data. *Environmetrics*, 16(5):507–522.

Zhang, W., Liu, Y., and Clerc, M. (2003). An adaptive PSO algorithm for reactive power optimization. In *IET Conference Proceedings*, pages 302–307. Institution of Engineering and Technology.

Zimmerman, D. L. (2006). Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652.

Zimmerman, D. L. and Cressie, N. (1992). Mean squared prediction error in the spatial linear model with estimated covariance parameters. *Annals of the Institute of Statistical Mathematics*, 44(1):27–43.