Stat
The ISI's Journal for the Rapid
Dissemination of Statistics Research
(wileyonlinelibrary.com) DOI: 10.100X/sta.0000

Adaptively-Tuned Particle Swarm Optimization with Application to Spatial Design

Matthew Simpson[a]*, Christopher K. Wikle[a], Scott H. Holan[a]

Received 00 Month Year; Accepted 00 Month Year

Particle swarm optimization (PSO) algorithms are a class of heuristic optimization algorithms that are especially attractive for complex optimizations problems. We propose using PSO in order to solve spatial design problems, e.g. choosing new locations to add to an existing monitoring network. To this end, we introduce a new class of PSO algorithms, called adaptively-tuned PSO, that perform well in a wide variety of circumstances. In order to illustrate these algorithms, we apply them to a common spatial design problem: choosing new locations to add to an existing monitoring network. Specifically, we consider a network in the Houston, TX area for monitoring ambient ozone levels, which have been linked to out-of-hostpital cardiac arrest rates (Ensor et al., 2013).

Keywords: optimization; particle swarm; geostatistics; kriging; optimal design, spatial prediction

1. Introduction

PSO refers to a large class of heuristic optimization algorithms that use an analogy with animal flocking behavior in order to construct more robust optimization algorithms than many alternatives (Clerc & Kennedy, 2002; Blum & Li, 2008; Clerc, 2010). This robustness makes them attractive for more difficult optimization problems, especially when near optimal solutions are tolerable. We introduce a new class of PSO algorithms, called adaptively tuned PSO (AT-PSO), which exploit an analogy with a class of adaptive Markov chain Monte Carlo algorithms in order to tune a crucial parameter of the PSO algorithm adaptively based on the state of the particle swarm. We show that the resulting algorithms tend to be superior to many PSO alternatives. Additionally, we illustrate several PSO algorithms by applying them to choosing a set of new monitoring locations for ozone in Harris County, Texas, where Houston is located. Ambient ozone levels have been linked to cardiac arrest (Ensor et al., 2013), so ozone monitoring is an essential tool for determining when and where populations are at risk. Section 2 introduces various PSO and AT-PSO

[a]Department of Statistics, University of Missouri,
146 Middlebush Hall, Columbia, MO 65211-6100
***Email: thematttsimpson@gmail.com**

Prepared using staauth.cls [Version: 2012/05/12 v1.00]

algorithms, and briefly discusses the results of an extended simulation study comparing them. Section 3 introduces the generic spatial design problem and the Houson area ozone problem as an instance of that problem, and compares several PSO algorithms and some alternatives to solving the problem. Finally, Section 4 discusses our results and concludes.

2. Particle swarm optimization

We briefly describe PSO here; refer to Blum & Li (2008) for a comprehensive introduction, Clerc (2010) for details, and Clerc (2011) for good default versions of the algorithm. Suppose that we wish to minimize some objective function $Q(\boldsymbol{\theta}) : \Re^D \to \Re$. Let $i = 1, 2, \ldots, n$ index a set of particles over time, $k = 1, 2, \ldots, K$, where in every period each particle consists of a location $\boldsymbol{\theta}_i(k) \in \Re^D$, a velocity $\boldsymbol{v}_i(k) \in \Re^D$, a personal best location $\boldsymbol{p}_i(k) \in \Re^D$, and a group best location $\boldsymbol{g}_i(k) \in \Re^D$. Here we mean "best" in the sense of minimizing Q, so $Q(\boldsymbol{p}_i(k)) \geq Q(\boldsymbol{\theta}_i(l))$ for any $k \geq l$. The group best location is defined with respect to some neighborhood $\mathcal{N}_i$ of particle i; that is, $\boldsymbol{g}_i(k) = \arg\min_{\{\boldsymbol{p}_j(k) | j \in \mathcal{N}_i\}} Q(\boldsymbol{p}_j(k))$. In the simplest case where the entire swarm is the neighborhood of each particle, $\boldsymbol{g}_i(k) \equiv \boldsymbol{g}(k) = \arg\min_{\{\boldsymbol{p}_j(k) | j \in 1:n\}} Q(\boldsymbol{p}_j(k))$. The generic PSO algorithm updates each particle i as follows:

$$
\begin{aligned}
\boldsymbol{v}_i(k+1) &= \omega \boldsymbol{v}_i(k) + \phi_1 \boldsymbol{r}_{1i}(k) \circ \{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\} + \phi_2 \boldsymbol{r}_{2i}(t) \circ \{\boldsymbol{g}_i(k) - \boldsymbol{\theta}_i(k)\}, \\
\boldsymbol{\theta}_i(k+1) &= \boldsymbol{\theta}_i(k) + \boldsymbol{v}_i(k+1), \\
\boldsymbol{p}_i(k+1) &= \begin{cases} \boldsymbol{p}_i(k) & \text{if } \; Q(\boldsymbol{p}_i(k)) \leq Q(\boldsymbol{\theta}_i(t+1)) \\ \boldsymbol{\theta}_i(k+1) & \text{otherwise,} \end{cases} \\
\boldsymbol{g}_i(k+1) &= \arg \min_{\{\boldsymbol{p}_j(k+1) | j \in \mathcal{N}_i\}} Q(\boldsymbol{p}_j(k+1)),
\end{aligned} \tag{1}
$$

where $\circ$ denotes the Hadamard product (element-wise product), $\boldsymbol{r}_{1i}(k)$ and $\boldsymbol{r}_{2i}(k)$ are each vectors of D random variates independently generated from the $U(0, 1)$ distribution, and $\omega > 0$, $\phi_1 > 0$, and $\phi_2 > 0$ are user-defined parameters. The term $\omega \boldsymbol{v}_i(k)$ controls the particle's tendency to keep moving in the direction it is already going, so ω is called the inertia parameter. Similarly $\phi_1 \boldsymbol{r}_{1i}(k) \circ \{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\}$ controls the particle's tendency to move towards its personal best location while $\phi_2 \boldsymbol{r}_{2i}(k) \circ \{\boldsymbol{g}_i(k) - \boldsymbol{\theta}_i(k)\}$ controls its tendency to move toward its group best location, so ϕ_1 and ϕ_2 are called the cognitive correction factor and social correction factor, respectively (Blum & Li, 2008). This version of PSO is equivalent to Clerc & Kennedy (2002)'s constriction type I particle swarm, though there are many other variants. One default choice sets $\omega = 0.7298$ and $\phi_1 = \phi_2 = 1.496$ (Clerc & Kennedy, 2002) while another sets $\omega = 1/(2 \ln 2) \approx 0.721$ and $\phi_1 = \phi_2 = 1/2 + \ln 2 \approx 1.193$ (Clerc, 2006).

Any PSO variant can also be combined with various neighborhood topologies that control how the particles communicate to each other. The default global topology allows each particle to see each other particle's previous best location for the social components of their respective velocity updates, but this can cause inadequate exploration and premature convergence. Alternative neighborhood topologies limit how many other particles each particle can communicate with. We use a variant of the stochastic star topology (Miranda et al., 2008). Each particle informs itself and k random particles from the swarm, sampled with replacement during initialization of the algorithm. On average this implies that each particle is informed by k particles, though a small number of particles will often be informed by many of the other particles

Bare bones PSO (BBPSO) is a variant of PSO introduced by Kennedy (2003) that strips away the velocity term. Let $\theta_{ij}(k)$ denote the jth coordinate of the position for the ith particle in period t, and similarly for $p_{ij}(k)$ and $g_{ij}(k)$. Then

the BBPSO algorithm obtains a new position coordinate θ_{ij} via

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

The updates of $\boldsymbol{p}_i(k)$ and $\boldsymbol{g}_i(k)$ are the same as in (1). We explain the essential ideas of PSO and BBPSO in this section, though several modifications can be made to improve performance. Appendix A details each of the modifactions we use, most of which come from Clerc (2011). In the next two subsections we introduce adaptively-tuned BBPSO and adaptively-tuned PSO respectively. Both of these algorithms can be combined with the various modifications we discuss in Appendix A.

2.1. Adaptively tuned BBPSO

BBPSO adapts the size effective search space of the swarm over time through the variance term, $|p_{ij}(k)-g_{ij}(k)|^2$. As the personal best locations of the swarm move closer together, these variances decrease and the swarm explores locations which are closer to known high value areas in the space. This behavior is desirable, but the adaptation is forced to occur only through personal and group best locations. In order to allow the algorithm to adapt in a more flexible manner, we modify the BBPSO variance to $\sigma^2(k)|p_{ij}(k)-g_{ij}(k)|^2$ and tune $\sigma^2(k)$ in a manner similar to adaptive random walk Metropolis MCMC algorithms (Andrieu & Thoms, 2008). Define the improvement rate of the swarm in period t as $R(k) = \#\{i : Q(\boldsymbol{p}_i(k)) > Q(\boldsymbol{p}_i(k-1))\}/n$ where $\#A$ is the number of members of the set A, and let R^* denote the target improvement rate. Then adaptively tuned BBPSO (AT-BBPSO) updates the swarm's personal best and group best locations as in (1), then updates particle locations as follows:

$$\begin{aligned}\theta_{ij}(k+1) &\sim t_{df}\left(\frac{p_{ij}(k)+g_{ij}(k)}{2}, \sigma^2(k)|p_{ij}(k)-g_{ij}(k)|^2\right),\\ \log\sigma^2(k+1) &= \log\sigma^2(k) + c\times\{R(k+1)-R^*\},\end{aligned} \quad (3)$$

where df is a user chosen degrees of freedom parameter and c is another user chosen parameter that controls the speed of adaptation. We use a t kernel instead of a Gaussian in order to allow for more flexibility, and we find $df = 1$ appears to combine well with adaptively tuning $\sigma^2(k)$. Setting the target rate from $R^* = 0.3$ to $R^* = 0.5$ tends to yield good AT-BBPSO algorithms, which is unsurprising given the connection to adaptive random walk Metropolis (Gelman et al., 1996). The parameter c controls the speed of adaptation so that larger values of c mean the algorithm adapts $\sigma^2(k)$ faster. We find that $c = 0.1$ works well, though anything within an order of maginitude yields similar results. We use $\sigma^2(0) = 1$ to initialize the algorithm at the standard BBPSO algorithm.

Both using a t kernel and adding a fixed scale parameter have been discussed in the BBPSO literature, but as Kennedy (2003) notes, something about setting $\sigma^2 = 1$ is special that causes the algorithm to work well. A similar BBPSO algorithm from Hsieh & Lee (2010) sets $\sigma^2 < 1$ for an initial set of iterations, then eventually sets $\sigma^2 = 1$. The authors alo suggest dynamically adjusting σ^2 in the early stage of the algorithm before they are set to one, but give no suggestion for how to do this. AT-BBPSO is able to adapt its value on the fly based on local knowledge about the objective function. If too much of the swarm is failing to find new personal best locations, AT-BBPSO proposes new locations closer to known high value areas. If too much of the swarm is improving, AT-BBPSO proposes bolder locations in an effort to make larger improvements. This ability to adapt to local information about the objective function allows AT-BBPSO to more quickly traverse the search space towards the global optimum, though by using local information AT-BBPSO does risk premature convergence to a local optimum. The adaptively tuned component of AT-BBPSO can also be combined with most BBPSO variants, some of which are outlined in Appendix A. In Appendix B we conduct a simulation study on several test functions that compares AT-BBPSO variants to other PSO and

BBPSO variants in order to justify the parameter settings discussed above and demonstrate AT-BBPSO variants are attractive PSO algorithms.

2.2. Adaptively-tuned PSO

In AT-BBPSO variants, the parameter $\sigma(k)$ partially controls the effective size of the swarm's search area, and we increase or decrease $\sigma(k)$ and consequently the search area depending on how much of the swarm is finding new personal best locations. In standard PSO the inertia parameter, denoted by ω in (1), is roughly analogous to σ^2 in BBPSO. It controls the effective size of the swarm's search area by controlling how the magnitude of the velocities evolve over time. In AT-BBPSO we use an analogy with tuning a random walk Metropolis-Hastings MCMC algorithm in order to build intuition about how to tune $\sigma(k)$. The analogy is much weaker in this case; nonetheless, the same mechanism works well to tune $\omega(k)$. Formally, AT-PSO updates personal and group best locations as in (1) and updates $\omega(k)$ and particle locations as follows:

$$\begin{aligned}
\boldsymbol{v}_i(k+1) &= \omega(k+1)\boldsymbol{v}_i(k) + \phi_1 \boldsymbol{r}_{1i}(k) \circ \{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\} + \phi_2 \boldsymbol{r}_{2i}(k) \circ \{\boldsymbol{g}_i(k) - \boldsymbol{\theta}_i(k)\}, \\
\boldsymbol{\theta}_i(k+1) &= \boldsymbol{\theta}_i(k) + \boldsymbol{v}_i(k+1), \\
\log \omega(k+1) &= \log \omega(k) + c \times \{R(k+1) - R^*\},
\end{aligned} \tag{4}$$

where $R(k)$ is the improvement rate of the swarm in iteration t, R^* is the target improvement rate, and c controls how much $R(k)$ changes on a per iteration basis. Again, we find that target rates from $R^* = 0.3$ to 0.5 work well for AT-PSO. The value of c controls the speed of adaptation. We use $c = 0.1$ as a default value and in simulations (not reported here), we find that the gains from optimizing c appear to be small. The adaptive tuning piece of this algorithm can be combined with almost any PSO variant, and Appendix A describes the various modifications to the standard PSO algorithm we employ, both in conjuction with AT and without it.

The idea of time-varying $\omega(k)$ has been in the PSO literature for some time. An early suggestion was to set $\omega(0) = 0.9$ and deterministically decrease it until it reaches $\omega(k) = 0.4$ after the maximum number of iterations allowed (Eberhart & Shi, 2000). In particular, Tuppadung & Kurutach (2011) suggest defining $\omega(k)$ via the parameterized inertia weight function $\omega(k) = \frac{1}{1+\left(\frac{t}{\alpha}\right)^{\beta}}$ where α and β are user-defined parameters. Roughly, α controls how low $\omega(k)$ can go and β controls how fast it gets there, so α and β can be thought of as intercept and slope parameters respectively. The suggestion in Tuppadung & Kurutach (2011) is to set α to a small fraction of the total amount of iterations in which the algorithm is allowed to run (e.g., 10% or 20%), and set β between one and four. We call this type of PSO algorithm deterministic inertia PSO (DI-PSO).

DI-PSO tends to improve on standard PSO if $\omega(k)$'s progression is set appropriately, but this can be difficult and often depends on the problem. A priori it may not be clear exactly which approach is best for any given problem, so an automatic method is desirable. A major strength of AT-PSO relative to DI-PSO and standard PSO is that AT-PSO can increase $\omega(k)$ when information from the swarm suggests there is an unexplored high value region of the space. Just like in AT-BBPSO, this mechanism provides a way for the swarm to adapt its behavior on the fly based on local conditions and speed up convergence by allowing the particles that do improve to make larger improvements, but it can also cause premature convergence to a local optimum. While DI-PSO monotonically decreases $\omega(k)$ toward some minimum value, AT-PSO typically oscillates $\omega(k)$ so that the algorithm alternates between exploring and exploiting more, relative to standard PSO. Appendix B contains an extended simulation study comparing a variety of these PSO and BBPSO algorithms on a suite of test functions that demonstrates some of the behavior detailed above and shows that AT-PSO is an attractive PSO algorithm.

Another similar algorithm comes from Zhang et al. (2003). In their algorithm, ω is constant while ϕ_1 and ϕ_2 not only vary across time, but also across the swarm. Then each particle adapts its values of ϕ_1 and ϕ_2 based on how much it improves on its personal best location. The key difference is that the adaptation is local to each specific particle while in AT-PSO the adaptation is global. AT-PSO takes into account less information by ignoring local conditions of the particles, but the adaptation scheme is simpler.

3. The Spatial Design Problem

Suppose we are interested predicting some spatially indexed response variable $Y(\boldsymbol{u})$, $\boldsymbol{u} \in \mathcal{D} \subseteq \Re^2$ at a set of target locations $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_{N_t} \in \mathcal{D}$. Let $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{N_s} \in \mathcal{D}$ denote a set of N_s fixed sampling locations within the spatial domain. The design problem is to add N_d new sampling locations in order to optimize the amount we learn about $Y(\boldsymbol{u})$ at the target locations. Let $\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{N_d} \in \mathcal{D}$ denote a set of candidate design points and suppose that $Y(\boldsymbol{u})$ is a geostatistical process with mean function $\mu(\boldsymbol{u}) = \boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}$ for some covariate $\boldsymbol{x}(\boldsymbol{u})$ known at every point in $\mathcal{D}$ and some covariance function $C(\boldsymbol{u}, \boldsymbol{v})$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{D}$. Not all covariates can be known a priori at every point in the spatial domain; however, covariates that are known functions of the location satisfy this constraint. Once the design points are selected, we observe $Z(\boldsymbol{d}_i)$ for $i = 1, 2, \ldots, N_d$ and $Z(\boldsymbol{s}_i)$ for $i = 1, 2, \ldots, N_s$ where $Z(\boldsymbol{u}) = Y(\boldsymbol{u}) + \varepsilon(\boldsymbol{u})$ and $\varepsilon(\boldsymbol{u})$ is mean zero white noise with variance τ^2, representing measurement error. Typically $\boldsymbol{\beta}$, τ^2, and $C(.,.)$ are unknown and must be estimated, though for now we will treat them as known.

To completely specify the problem we need to define an informative design criterion. Intuitively, the larger the mean square prediction error (MSPE), i.e. the kriging variance, is at each of the target locations, the less information we have about $Y(\boldsymbol{u})$ at those locations. A common design criterion is to optimize function of these variances, e.g. to minimize the mean kriging variance or the maximum kriging variance over all target locations. These criteria are somewhat naive since they they ignore parameter uncertainty, and taking that into account will often change the optimal design (Zimmerman, 2006).

3.1. Universal Kriging

In universal kriging, $C(.,.)$ and τ^2 are treated as known while $\boldsymbol{\beta}$ is treated as a parameter that neeeds to be estimated. Let $\boldsymbol{Z}$ be the vector of $Z(\boldsymbol{s}_i)$s and $Z(\boldsymbol{d}_i)$s, $\boldsymbol{X}$ denote the corresponding stacked $\boldsymbol{x}(\boldsymbol{s}_i)'$s and $\boldsymbol{x}(\boldsymbol{d}_i)'$s, $\boldsymbol{C}_Z = \text{cov}(\boldsymbol{Z})$ where $\text{cov}[Z(\boldsymbol{u}), Z(\boldsymbol{v})] = C(\boldsymbol{u}, \boldsymbol{v}) + \sigma^2_\varepsilon 1(\boldsymbol{u} = \boldsymbol{v})$, and $\boldsymbol{c}_Y(\boldsymbol{t}_i) = \text{cov}[Y(\boldsymbol{t}_i), \boldsymbol{Z}]$ where $\text{cov}[Y(\boldsymbol{t}_i), Z(\boldsymbol{u})] = C(\boldsymbol{t}_i, \boldsymbol{u})$. The universal kriging predictor of $Y(\boldsymbol{t}_i)$ is $\widehat{Y}_{uk}(\boldsymbol{t}_i; \boldsymbol{d}) = \boldsymbol{x}(\boldsymbol{t}_i)'\widehat{\boldsymbol{\beta}}_{gls} + \boldsymbol{c}_Y(\boldsymbol{t}_i)'\boldsymbol{C}_Z^{-1}(\boldsymbol{Z} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}_{gls})$ and the MSPE of $\widehat{Y}_{uk}(\boldsymbol{t}_i)$ is

$$\begin{aligned}\sigma^2_{uk}(\boldsymbol{t}_i; \boldsymbol{d}) = C(\boldsymbol{t}_i, \boldsymbol{t}_i) - \boldsymbol{c}_Y(\boldsymbol{t}_i)'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i) \\ + [\boldsymbol{x}(\boldsymbol{t}_i) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i)]'[\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X}]^{-1}[\boldsymbol{x}(\boldsymbol{t}_i) - \boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{c}_Y(\boldsymbol{t}_i)].\end{aligned}$$

where $\widehat{\boldsymbol{\beta}}_{gls} = [\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{X}]^{-1}\boldsymbol{X}'\boldsymbol{C}_Z^{-1}\boldsymbol{Z}$ is the generalized least squares estimate of $\boldsymbol{\beta}$ (Cressie & Wikle, 2011, Section 4.1.2). To avoid clutter we drop the explicit dependence on $\boldsymbol{d}$ in these equations, but $\boldsymbol{c}_Y(\boldsymbol{t}_i)$, $\boldsymbol{C}_Z$, $\boldsymbol{Z}$, $\boldsymbol{X}$, and $\widehat{\boldsymbol{\beta}}_{gls}$ all depend on $\boldsymbol{d}$. The mean universal kriging variance is given by $\overline{Q}_{uk}(\boldsymbol{d}) = \frac{1}{N_t}\sum_i^{N_t} \sigma^2_{uk}(\boldsymbol{t}_i; \boldsymbol{d})$ while the maximum universal kriging variance is given by $Q^*_{uk}(\boldsymbol{d}) = \max_{i=1,2\ldots,N_t} \sigma^2_{uk}(\boldsymbol{t}_i; \boldsymbol{d})$. Zimmerman (2006) finds that the optimal design under both criteria is highly dependent on the class of mean function of the geostatistical process. In practice we are often interested in predicting at the entire spatial domain rather than a finite set of target locations. This changes the mean and maximum kriging variances to $\overline{Q}_{uk}(\boldsymbol{d}) = \frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} \sigma^2_{uk}(\boldsymbol{u}; \boldsymbol{d})d\boldsymbol{u}$ and $Q^*_{uk}(\boldsymbol{d}) = \max_{\boldsymbol{u}\in\mathcal{D}} \sigma^2_{uk}(\boldsymbol{u}; \boldsymbol{d})$ respectively. We can approximate both of these with a large but finite sample of target locations from $\mathcal{D}$ or a large fixed grid in $\mathcal{D}$.

3.2. Parameter Uncertainty Kriging

Assuming that all covariance function parameters are known, the MSPE from kriging at an arbitrary location $\boldsymbol{u}$ is $\sigma^2_{uk}(\boldsymbol{u})$. This underestimates the MSPE when those parameters must be estimated. An approximation of the correct MSPE, which we call the parameter uncertainty kriging variance (PUK variance), is given by (Zimmerman & Cressie, 1992; Abt, 1999)

$$\mathrm{E}[Y(\boldsymbol{u}) - \widehat{Y}_{uk}(\boldsymbol{u})]^2 \approx \sigma^2_{puk}(\boldsymbol{u}; \boldsymbol{d}, \widehat{\boldsymbol{\theta}}) = \sigma^2_{uk}(\boldsymbol{u}; \boldsymbol{d}, \widehat{\boldsymbol{\theta}}) + \mathrm{tr}[\boldsymbol{A}(\boldsymbol{u}; \boldsymbol{d}, \widehat{\boldsymbol{\theta}}) \boldsymbol{I}^{-1}(\boldsymbol{d}, \widehat{\boldsymbol{\theta}})]$$

where $\widehat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$ from previously observed data, $\boldsymbol{I}^{-1}$ is the inverse Fisher information (FI) matrix, and $\boldsymbol{A} = \mathrm{var}[\partial \widehat{Y}_{uk} / \partial \boldsymbol{\theta}]$. The ijth element of the FI matrix can be derived as $\mathrm{tr}\left(\boldsymbol{C}_Z^{-1} \frac{\partial \boldsymbol{C}_Z}{\partial \theta_i} \boldsymbol{C}_Z^{-1} \frac{\partial \boldsymbol{C}_Z}{\partial \theta_j}\right)$ while $\boldsymbol{A}$ can be derived using elementary matrix calculus. From these we define the mean and maximum PUK variances as $\overline{Q}_{puk}(\boldsymbol{d}) = \frac{1}{N_t} \sum_i^{N_t} \sigma^2_{puk}(\boldsymbol{t}_i; \boldsymbol{d})$ and $Q^*_{puk}(\boldsymbol{d}) = \max_{i=1,2\ldots,N_t} \sigma^2_{puk}(\boldsymbol{t}_i; \boldsymbol{d})$ respectively.

3.3. Houston Ozone Monitoring

The Texas Commission on Environmental Quality (TCEQ) publishes a variety of environmental data for Texas, including many environmental indicators that directly relate to public health. We focus on ozone in the Houston-Galveston-Brazoria area. The TCEQ measures ozone at a network of monitoring locations in this area and publishes daily maximum eight-hour ozone concentrations (DM8s) in parts per billion (ppb) for each monitoring location. DM8s are computed as follows. First, the TCEQ creates publishes an hourly average (in ppb) for each monitoring location. Then an eight hour average is constructed at that location for each contiguous eight-hour period where all eight measurements were present for a given day. The maximum of these eight-hour averages for a given day is the published DM8. Days with less than 18 valid eight-hour averages have no published DM8. A 20 ppb increase in DM8 has been associated with an increased risk of out-of-hospital cardiact arrest, with a relative risk estimate of 1.039 (Ensor et al., 2013, 95% CI 1.005, 1.073).

In August 2016 there were 44 active monitoring locations in the Houston-Galveston-Brazoria area. For each location $\boldsymbol{u}$ we compute the monthly average DM8, which we denote by $Z(\boldsymbol{u})$. At one location, MRM-3 Haden Road, there are two DM8 observations of 0 ppb in the month of August. We assume that these were data errors and omit them for the purposes of computing $Z(\boldsymbol{u})$ at that location. Of the 44 locations, one has 15 valid DM8 measurements of 31 possible valid measurements, another has 24 valid measurements, and the rest of the locations have at least 27 valid measurements.

The hypothetical design problem we consider is the addition of five new ozone monitoring locations to the Houston-Galveston-Brazoria monitoring network in Harris County, where Houston is located, with the goal of predicting ozone concentrations within Harris County. Of the 44 existing locations, 33 are already in Harris County, though the locations outside of the county are still useful for spatial prediction within Harris County.

Let $Z(\boldsymbol{u})$ denote the measured average DM8 as location $\boldsymbol{u}$ and let $Y(\boldsymbol{u})$ denote the true DM8. We assume that $Z(\boldsymbol{u})$ is a noisy Gaussian measurement of $Y(\boldsymbol{u})$; i.e., the data model is $Z(\boldsymbol{u}) \sim N[Y(\boldsymbol{u}), \tau^2]$. The parameter τ^2 represents variability added due to both measurement error from the instruments and potentially sampling error from measuring DM8 on less than the full 31 days in August. At the process level, we assume that $Y(\boldsymbol{u})$ is a geostatistical process with mean function $\mu(\boldsymbol{u}) = \boldsymbol{x}(\boldsymbol{u})'\boldsymbol{\beta}$ and exponential covariance function $C(\boldsymbol{u}, \boldsymbol{v}) = \sigma^2 \exp(-||\boldsymbol{u} - \boldsymbol{v}||/\psi)$. We assume that any fine scale variability is measurement error and captured in the data model. We considered several possible mean functions: constant in $\boldsymbol{u}$, linear in $\boldsymbol{u}$, and quadratic in $\boldsymbol{u}$. We fit each model using maximum likelihood and found that quadratic terms were unnecessary, but linear terms did significantly help explain variation in $Y(\boldsymbol{u})$.

We use both PUK design criteria in order to choose the five new locations in Harris County: $\overline{Q}_{puk}(\boldsymbol{d})$ and $Q^*_{puk}(\boldsymbol{d})$, both defined in Section 3. We assume that the goal is the predict average DM8 in all of Harris County, so we approximate the continuous versions of $\overline{Q}_{puk}(\boldsymbol{d})$ and $Q^*_{puk}(\boldsymbol{d})$ with the finite sample versions using a grid of 1229 points, obtained by gridding up the smallest rectangle containing Harris County and throwing away all points outside of hte county. We try a variety of PSO algorithms in order to select the new locations. Since the design space only allows new monitoring locations within Harris County, we define $\overline{Q}_{puk}(\boldsymbol{d}) = Q^*_{puk}(\boldsymbol{d}) = \infty$ when any of the proposed locations are outside of Harris County. Existing sampling locations outside of Harris County are still used in the estimation of the model and in the construction of the PUK variances.

[THIS PARAGRAPH WILL BE PART OF EXPLAINING THE HOUSTON TABLE] In Table <ref> PSO refers to the standard PSO algorithm with the usual velocity update and all of the other modifications listed in Appendix A.1. The CF modifier indicates that the velocity update is coordinate-free – see Appendix A.1 for details. The DI modifier indicates that $\omega(k)$ is deterministically evolved with $\alpha = 0.2K$ and $\beta = 2$ as in Section 2.2, where recall K is the total number of PSO iterations. In simulations not reported here, we found those parameter values tended to yield the best DI-PSO algorithms. The AT modified indicates that $\omega(k)$ is adaptively tuned as described in Section 2.2. AT1 uses $R^* = 0.3$ and $AT2$ uses $R^* = 0.5$, while both use $c = 0.1$ and $\omega(0) = 1.2$. We use two sets of parameter values for all of the PSO algorithms: $\phi_1^{(1)} = \phi_2^{(1)} = 1.496$ and $\phi_1^{(2)} = \phi_2^{(2)} = \ln(2) + 1/2$, and when applicable the corresponding inertias are $\omega^{(1)} = 0.7298$ and $\omega^{(2)} = 1/(2\ln 2)$. All of the BBPSO algorithms we consider are AT, with AT1-BBPSO and AT2-BBPSO using the same parameter values as AT1-PSO and AT2-PSO. All PSO algorithms also use $df = 1$, and each of the modications detailed in Appendix A.2. Further, the modifier CF indicates that the BBPSO algorithm uses the coordinate-free variance update and xp indicates it has a 0.5 probability of moving any given coordinate of a particle to its personal best location on that coordinate, both described in Appendix A.2. All of the PSO and BBPSO algorithms use either the global neighborhood topology, or the variant of the stochastic star neighborhood topology discussed is Section 2 with either one or three informants. Further, each PSO algorithm has a swarm size of 40 and is run for $K = 500$ iterations. Additionally, we employ two other classes of alternative algorithms to compare to PSO. First, we use the genetic algorithm described in Hamada et al. (2001), with either one or two batches (B), and with two possible mutation rates (λ) and two possible mutation variances (μ). The genetic algorithms also use a population of 40, though they are only allowed to run for 250 iterations so that after the initialization, the genetic algorithms use the same number of objective function evaluations as each of the PSO algorithms. Finally, we use an exchange algorithm (Miller & Nguyen, 1994). The exchange algorithm is commonly used in spatial and spatio-temporal design problems (Wikle & Royle, 1999, 2005), and uses a discrete grid as the search space and operates by considering the neighbors of the current point. We use the target grid, $\boldsymbol{t}$, as the search space and consider tenth-order neighbors. Each algorithm was implemented in the R programming language (R Core Team, 2016).

[INSERT PSO RESULTS HERE... CURRENTLY RUNNING/PENDING TO BE RUN ON THE SERVER]

4. Discussion

[TO BE WRITTEN]

References

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

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

Blum, C & Li, X (2008), 'Swarm intelligence in optimization,' in Blum, C & Merkle, D (eds.), *Swarm Intelligence: Introduction and Applications*, Springer.

Clerc, M (2006), 'Stagnation analysis in particle swarm optimisation or what happens when nothing happens,' Tech. rep.

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

Clerc, M (2011), 'Standard particle swarm optimisation,' Tech. rep.

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

Cressie, N & Wikle, CK (2011), *Statistics for Spatio-Temporal Data*, John Wiley & Sons.

Eberhart, RC & Shi, Y (2000), 'Comparing inertia weights and constriction factors in particle swarm optimization,' in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, IEEE, vol. 1, pp. 84–88.

Ensor, KB, Raun, LH & Persse, D (2013), 'A case-crossover analysis of out-of-hospital cardiac arrest and air pollution,' *Circulation*, **127**(11), pp. 1192–1199.

Gelman, A, Roberts, G & Gilks, W (1996), 'Efficient Metropolis jumping rules,' *Bayesian statistics*, **5**(599-608), p. 42.

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

Hsieh, HI & Lee, TS (2010), 'A modified algorithm of bare bones particle swarm optimization,' *International Journal of Computer Science Issues*, **7**, p. 11.

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

Miller, AJ & Nguyen, NK (1994), 'Algorithm as 295: A fedorov exchange algorithm for d-optimal design,' *Journal of the royal statistical society. series c (applied statistics)*, **43**(4), pp. 669–677.

Miranda, V, Keko, H & Duque, AJ (2008), 'Stochastic star communication topology in evolutionary particle swarms (epso),' *International journal of computational intelligence research*, **4**(2), pp. 105–116.

R Core Team (2016), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Tuppadung, Y & Kurutach, W (2011), 'Comparing nonlinear inertia weights and constriction factors in particle swarm optimization,' *International Journal of Knowledge-based and Intelligent Engineering Systems*, **15**(2), pp. 65–70.

Wikle, CK & Royle, JA (1999), 'Space: time dynamic design of environmental monitoring networks,' *Journal of Agricultural, Biological, and Environmental Statistics*, pp. 489–507.

Wikle, CK & Royle, JA (2005), 'Dynamic design of ecological monitoring networks for non-gaussian spatio-temporal data,' *Environmetrics*, **16**(5), pp. 507–522.

Zhang, W, Liu, Y & Clerc, M (2003), 'An adaptive pso algorithm for reactive power optimization,' in *Advances in Power System Control, Operation and Management, 2003. ASDCOM 2003. Sixth International Conference on (Conf. Publ. No. 497)*, IET, vol. 1, pp. 302–307.

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

Zimmerman, DL & 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), pp. 27–43.

Stat
The ISI's Journal for the Rapid
Dissemination of Statistics Research
(wileyonlinelibrary.com) DOI: 10.100X/sta.0000

Supplemental Web Material: Adaptively-Tuned Particle Swarm Optimization with Application to Spatial Design

Matthew Simpson[a]

Received ; Accepted

A. PSO and BBPSO details

In Section 2 we introduced PSO, BBPSO, and our adaptively tuned variants of both. Here we describe in detail several modifcations to both PSO and BBPSO.

A.1. PSO

The standard PSO algorithm is given by equation (1). We consider several additions and modifications to this algorithm below. Each of these modifications is combined with adaptively tuned inertia as in equation (4) to create the AT-PSO algorithms we employ.

A.1.1. Initialization: In order to initialize the swarm, the number of particles, their initial locations, and their initial velocities have to be chosen. Clerc (2011) suggests making the swarm size a function of the dimension of the search space, but notes that this is known to be suboptimal. We use their alternative suggest to use a default swarm size of 40. We assume the search space is a D-dimensional hypercube given by $\bigtimes_{j=1}^{D}[min_j, max_j]$. Then each particle's initial location is randomly generated uniformly on the cube, i.e. $\theta_{ij}(0) \overset{ind}{\sim} U(min_j, max_j)$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, D$. Each particle's velocity is initalized based on its location via $v_{ij}(0) \overset{ind}{\sim} U(min_j - x_{ij}(0), max_j - x_{ij}(0))$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, D$.

[a]**Department of Statistics, University of Missouri,**
146 Middlebush Hall, Columbia, MO 65211-6100
[*]**Email: themattsimpson@gmail.com**

Prepared using staauth.cls [Version: 2012/05/12 v1.00]

In Section 3.3 each design point is constrained to be in Harris County, TX, wich is not a rectangle. We intialize the swarm by placing intial design points on the smallest rectangle containing Harris County. Design points outside of Harris County will quickly move back into the county due to the confinement strategy.

A.1.2. Confinement: Even though the initialization of the swarm is confined to a hypercube, nothing prevents any given particle from leaving the search space. There are a number of things that can be done in order to solve this problem and each has advantages and disadvantages depending on the situation (Helwig & Wanka, 2007). We focus on two approaches. One is to move any particle that leaves the search space to the nearest point still inside the search space and then adjust its velocity, e.g. Clerc (2011) suggests that whenever $x_{ij}(k) > max_j$, it should be set to $x_{ij}(k) = max_j$ and similarly when $x_{ij}(k) < min_j$ it should be set to $x_{ij}(k) = min_d$, and in both cases the velocity along that dimension should be reversed and halved, i.e. $v_{ij}(k) = -0.5v_{ij}(k)$. This causes particles to bounce off the boundary and move back towards the middle of the search space. This is the default strategy we use.

Another strategy is to simply define the objective function to be $\pm\infty$ outside of the search space ($+$ when minimizing, $-$ when maximizing) so that the swarm tries to stay in bounds naturally. In Section 3.3 we employ this method in combination with the other strategy. When a design point is proposed outside of the smallest rectangle containing Harris County, we move it back to the edge of the rectangle using the method described in the previous paragraph. All points inside the rectangle but outside Harris County are defined to have infinite MSPE variance in order to further restrict the swarm to the desired area.

A.1.3. Redraw Neighborhoods: In Section 2 we briefly described a variant of the stochastic star neighborhood which we use. This neighborhood is stochastic, meaning that each particle's neighbors are randomly drawn when the algorithm is initialized. After an iteration in which the best known value of the objective function is unchanged, each particle's neighbors are randomly redrawn according to the same distribution.

A.1.4. Asynchronous Updates: The way we defined PSO in equation (1) each particle can update simultaneously. This means that the algorithm is parallelizable, which is a major advantage for implementation on modern GPUs. However, asynchronously updating the particles typically results in faster converging algorithms when it is computationally feasible. In an asynchronous update from period k to $k+1$, particle i recognizes that particle $i-1$ has already updated its personal best location to $\boldsymbol{p}_{i-1}(k+1)$ by the time it is i's turn to update. So i computes its group best location taking this into account. This causes particle $i = 1$ to behave differently from particle $i = n$ since particle n always has better information in order to perform its update, so every iteration the particles are randomly reordered. More formally, before every iteration sample $o_1(k+1), o_2(k+1), \ldots, o_n(k+1)$ from $\{1, 2, \ldots, n\}$ without replacement. Then the particles update starting with o_1, o_2, etc., where the group best update becomes $\boldsymbol{g}_{o_i}(k+1) = \arg\min_{\{\boldsymbol{p}_{o_j}(k+1)|j\in\mathcal{N}_i\}} Q(\boldsymbol{p}_j(k_j^*))$ where $k_j^* = k+1$ if $j < i$ and k otherwise. We asynchronously update in all of our algorithms.

A.1.5. Coordinate Free Velocity Updates: The standard velocity update in equation (1) is well known to bias the algorithm towards locations near the coordinate axes and especially the origin (Monson & Seppi, 2005; Spears et al., 2010). In general we may not know if the true optimum is near an axis, so this behavior is undesirable. There are several alternative velocity updates available, e.g. in Monson & Seppi (2005). We use the coordinate free (CF) update suggested by Clerc (2011). First define the center of gravity for particle i to be $\boldsymbol{C}_i(k) = \boldsymbol{\theta}_i(k) + \phi_1\{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\}/3 + \phi_2\{\boldsymbol{g}_i(k) - \boldsymbol{\theta}_i(k)\}/3$. Let $\mathcal{H}_i(k)$ denote the hypersphere centered at $\boldsymbol{C}_i(k)$ with radius $||\boldsymbol{C}_i(k) - \boldsymbol{\theta}_i(k)||$ where $||.||$ denotes Euclidean distance. Then a new point $\boldsymbol{\theta}_i'(k)$ is drawn randomly from $\mathcal{H}_i(k)$ by sampling a direction and a radius, each uniformly. This is *not* the same as drawing uniformly over $\mathcal{H}_i(k)$ and in fact

favors points near the center. Then the CF velocity update is given by $\boldsymbol{v}_i(k+1) = \omega \boldsymbol{v}_i(k) + \boldsymbol{x}_i'(k)$. We use both the standard and CF velocity updates in our algorithms. The standard PSO algorithm with each feature in this subsection including the CF velocity update is what Clerc (2011) calls SPSO 2011.

A.1.6. When Personal Best = Group Best: When a particle's personal best and group best locations coincide, it is often advantageous to allow the particle to explore more than usual. In the standard velocity update we do this by removing the social term so that $\boldsymbol{v}_i(k+1) = \omega \boldsymbol{v}_i(k) + \phi_1 \boldsymbol{r}_{1i}(k) \circ \{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\}$. In the CF velocity update we change the center of gravity to ignore the social term so that $\boldsymbol{C}_i(k) = \boldsymbol{\theta}_i(k) + \phi_2\{\boldsymbol{p}_i(k) - \boldsymbol{\theta}_i(k)\}/2$.

A.2. BBPSO

The standard BBPSO algorithm was introduced by Kennedy (2003) and updates from t to $t+1$ via equation (2). We use each of the features in Section A.1 in our BBPSO algorithms, though some of them need to be modified for the BBPSO setting. We list them below along with another modification of BBPSO which we employ. Each of these modications are combined with adaptively tuning a scale parameter as in equation (3) to create our AT-BBPSO algorithms.

A.2.1. BBPSOxp: A commonly used variant of BBPSO also introduced by Kennedy (2003) is called BBPSOxp. In this variant, each coordinate of each particle has a 50% chance of updating according to (2) and a 50% chance of moving directly to that particle's personal best location on that coordinate. In other words

$$\theta_{ij}(k+1) = \begin{cases} N\left(\frac{p_{ij}(k)+g_{ij}(k)}{2}, h_{ij}^2(k)\right) & \text{with probability 0.5} \\ p_{ij}(k) & \text{otherwise,} \end{cases} \tag{1}$$

where $h_{ij}(k) = |p_{ij}(k) - g_{ij}(k)|$. We use both xp and non-xp versions of our BBPSO algorithms.

A.2.2. CF BBPSO: BBPSO's update also depends on the coordinate system since each coordinate of $\boldsymbol{\theta}$ gets a different standard deviation. We employ BBPSO algorithms using the default standard deviation, but also using a coordinate free standard deviation given by $h_{ij}(k) = ||\boldsymbol{p}_i(k) - \boldsymbol{g}_i(k)||$.

A.2.3. When Personal Best = Group Best in BBPSO: A downside of both BBPSO and BBPSOxp is that any particle whose personal best is currently its group best location does not move due to the definition of the standard deviation term. Several methods have been proposed to overcome this; e.g. Hsieh & Lee (2010) and Zhang et al. (2011). Zhang et al. (2011) propose using mutation and crossover operations for the group best particle. To do this, each group best particle randomly selects three other distinct particles from the entire swarm, i_1, i_2, and i_3, and updates according to

$$\theta_{ij}(k+1) = p_{i_1 j}(k) + 0.5\{p_{i_2 j}(k) - p_{i_3 j}(k)\}. \tag{2}$$

This combines easily with BBPSOxp to update each coordinate of each particle with $h_{ij}(k) = 0$ according to (2) and the rest according to (1).

B. Comparing PSO and BBPSO algorithms

[THIS SECTION IS OUT OF DATE AND WILL BE UPDATED ONCE THE NEW SIMULATIONS COME OFF]

In order to compare AT-BBPSO to other PSO variants, we employ a subset of test functions used in Hsieh & Lee (2010). Each function is listed in Table 1 along with the global maximum and argmax, and the initialization range for the simulations. Further description of many of these functions can be found in Clerc (2010). For each function, we set $D = 20$ so the domain of each function is $\Re^{20}$. For each function, the PSO algorithms are initialized in a range that does not contain the true maximum.

Equation	ArgMax	Maximum	Initialization
$Q_1(\boldsymbol{\theta}) = -\sum_{i=1}^{D} \theta_i^2$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_1(\boldsymbol{\theta}^*) = 0$	$(50, 100)^D$
$Q_2(\boldsymbol{\theta}) = -\sum_{i=1}^{D} \left(\sum_{j=1}^{i} \theta_j\right)^2$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_2(\boldsymbol{\theta}^*) = 0$	$(50, 100)^D$
$Q_3(\boldsymbol{\theta}) = -\sum_{i=1}^{D-1} \left[100\{\theta_{i+1} + 1 - (\theta_i + 1)^2\} + \theta_i^2\right]$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_3(\boldsymbol{\theta}^*) = 0$	$(15, 30)^D$
$Q_4(\boldsymbol{\theta}) = 9D - \sum_{i=1}^{D} \{\theta_i^2 - \cos(2\pi\theta_i) + 10\}$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_4(\boldsymbol{\theta}^*) = 0$	$(2.56, 5.12)^D$
$Q_5(\boldsymbol{\theta}) = -\frac{1}{4000}\|\|\boldsymbol{\theta}\|\|^2 + \prod_{i=1}^{D} \cos\left(\frac{\theta_i}{\sqrt{i}}\right) - 1$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_5(\boldsymbol{\theta}^*) = 0$	$(300, 600)^D$
$Q_6(\boldsymbol{\theta}) = 20\exp\left(-0.2\sqrt{\frac{1}{D}\|\|\boldsymbol{\theta}\|\|}\right)$ $+ \exp\left\{\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi\theta_i)\right\} - 20 - \exp(1)$	$\boldsymbol{\theta}^* = \mathbf{0}$	$Q_6(\boldsymbol{\theta}^*) = 0$	$(16, 32)^D$

Table 1. Test functions for evaluating PSO algorithms. The dimension of $\boldsymbol{\theta}$ is D and $||.||$ is the Euclidean norm: $||\boldsymbol{\theta}|| = \sqrt{\sum_{i=1}^{D} \theta_i^2}$.

We use several PSO algorithms in the simulation study. The standard PSO algorithm uses the parameter values suggested by Blum & Li (2008) and Clerc & Kennedy (2002). The AT-BBPSO variants are implemented a wide variety of parameter values, but all have the scale parameter initialized at $\sigma(0) = 1$, and both set $c = 0.1$. The AT-PSO variants are initialized at $\omega(0) = 1$ and also set $c = 0.1$. In addition, each algorithm is implemented using each of three neighborhood structures — the global, ring-3, and ring-1 neighborhoods. Each algorithm was used to optimize each objective function for 500 iterations over 50 replications using 20 particles. Initializations were changed across replications but held constant across algorithms. The standard PSO, DI-PSO, and AT-PSO algorithms initialized their velocity terms using the same method as a function of the initial locations of the particles, which we will denote by $\boldsymbol{x}_i$ for particle i. Let $x_{max,j}$ be the maximum initial value of coordinate j of $\boldsymbol{x}_i$ for each particle i, and let $x_{min,j}$ be the corresponding minimum. Then let $d_{max} = max_j x_{max,j} - x_{min,j}$. Then we initialize the velocities with $v_{ij}(0) \overset{iid}{\sim} U(-d_{max}/2, d_{max}/2)$. Tables 2-7 contain the simulation results for objective functions 1-6 respectively (OF1, OF2, etc.). We use several measures to quantify how well each algorithm finds the global maximum. First, each table includes the mean and standard deviation of the absolute difference between the true global maximum and the algorithm's estimated global maximum across all 50 replications, denoted by Mean and SD. Second, each table includes a convergence criterion — the proportion of the replications that came within 0.01 of the true global maximum, denoted by $\widehat{p}$. Finally, $\widehat{k}$ denotes the median number of iterations until the algorithm reaches the convergence criterion. When $\widehat{p} < 0.5$ then $\widehat{k} = \infty$ since greater than 50% of the replications did not converge in the maximum number of iterations allowed. Then Mean, $\widehat{p}$, and $\widehat{k}$ can be thought of how close the algorithm gets to the global maximum on average, what proportion of the time it converges, and long it takes to converge respectively.

We highlight only some of the features of these tables. First and foremost, PSO, BBPSO-MC, and BBPSOxp-MC almost always do worse than their AT cousins. Adaptively tuning either the scale or inertia parameter leads to gains in all three of our measures, sometimes large. The comparison is starkest between BBPSO variants and AT-BBPSO variants, partially because BBPSO tends to be pretty bad but also because AT-BBPSO does very well. Second, for most non-AT algorithms the more restrictive neighborhoods appear to yield algorithms which do a better job of finding the global max. For the AT-PSO algorithms, it appears that the target improvement rate (R^*) and the neighborhood

interact. When the rate is high a more restrictive neighborhood is preferable, while when the rate is low a less restrictive neighborhood is preferable. On the other hand, for the AT-BBPSO algorithms, the opposite appears to be true — when the target improvement rate is high, a less restrictive neighborhood is desirable and vice versa. In addition, the best AT-BBPSO algorithms often use the global neighborhood while for the other classes of algorithms their best versions typically use the ring-1 or ring-3 neighborhood.

For the DI-PSO algorithms often there is a parameter-neighborhood combination that does well, typically from setting $\alpha = 200$ (20% of the 500 iterations) and $\beta = 1$ and using either the ring-1 or ring-3 neighborhood. One of these combinations typically does the best of all the DI-PSO algorithms but they can sometimes still do much worse than the best alternatives (e.g., for OF2). In the AT-BBPSO algorithms, it is not always clear what the best parameter settings are, but good default values appear to be $df = 3$ or 5 and $R^* = 0.3$ or 0.5. When these values are good, they often lead to the best performing PSO algorithms we consider. However, it is not always clear whether to use AT-BBPSO-MC or AT-BBPSOxp-MC. For some objective functions, e.g. OF4, the xp version is consistently better than the non-xp alternative, but the opposite is true for others, e.g. OF2.. AT-PSO is also often very competitive with $R^* = 0.3$ or $R^* = 0.5$, though again sometimes different parameter settings also appear to work well.

The DI-PSO and AT-PSO algorithms are similar conceptually, but often yield very different results. DI-PSO deterministically reduces the inertia parameter over time in the same manner given a fixed set of parameter values (α and β), while AT-PSO dynamically adjusts the inertia parameter to hit a target improvement rate. Figure 1 plots the inertia over time for the DI-PSO algorithm with $\alpha = 200$ and $\beta = 1$, and observed inertia over time for one replication of the AT-PSO algorithm with target rate $R^* = 0.5$ and ring-1 neighborhood for OF1 and one replication for OF6. All three algorithms have an initial inertia of $\omega(0) = 1$. While DI-PSO smoothly decreases its inertia with a slowly decreasing rate, AT-PSO very quickly drops its inertia for OF1 to about 0.55 then bounces around around near that point. It also jumps up above 1 initially, imploring the particles to cast a wider net in search of higher value areas of the search space. This is pretty typical behavior for the inertia parameter of AT-PSO — it tends to bounce around a level which is approximately the average over time of the DI-PSO's inertia, though lower values of R^* will result in higher inertias. In this way, AT-PSO alternates periods of exploration (relatively high inertia) and periods of exploitation (relatively low inertia). The main exception to this pattern is when AT-PSO converges around a local maximum. In this case, inertia plummets to zero as the particles settle down. This is precisely what happens for OF6 in Figure 1, though in this case the maximum is not global — Table 7 indicates that ring-1 AT-PSO with $R^* = 0.5$ never converged to the global max. In optimization problems with multiple local optima, both AT-PSO and AT-BBPSO variants can exhibit this behavior and prematurely converge to a local optima, so they may not be advantageous for those problems.

Based on these simulations, our default recommendation is to use AT-BBPSO-MC or AT-BBPSOxp-MC with $R^* = 0.3$ or 0.5 and $df = 3$ or $df = 5$ along with the global neighborhood. These algorithms will not always be the best of the PSO algorithms, but they will often be very good. AT-PSO with $R^* = 0.3$ or $R^* = 0.5$ with a restrictive neighborhood topology such as ring-3 also tends to be a very good choice, though perhaps less consistent than the AT-BBPSO variants. Default PSO also performs rather well and is a good baseline algorithm to use for comparisons.

References

Blum, C & Li, X (2008), 'Swarm intelligence in optimization,' in Blum, C & Merkle, D (eds.), *Swarm Intelligence: Introduction and Applications*, Springer.

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

Clerc, M (2011), 'Standard particle swarm optimisation,' Tech. rep.

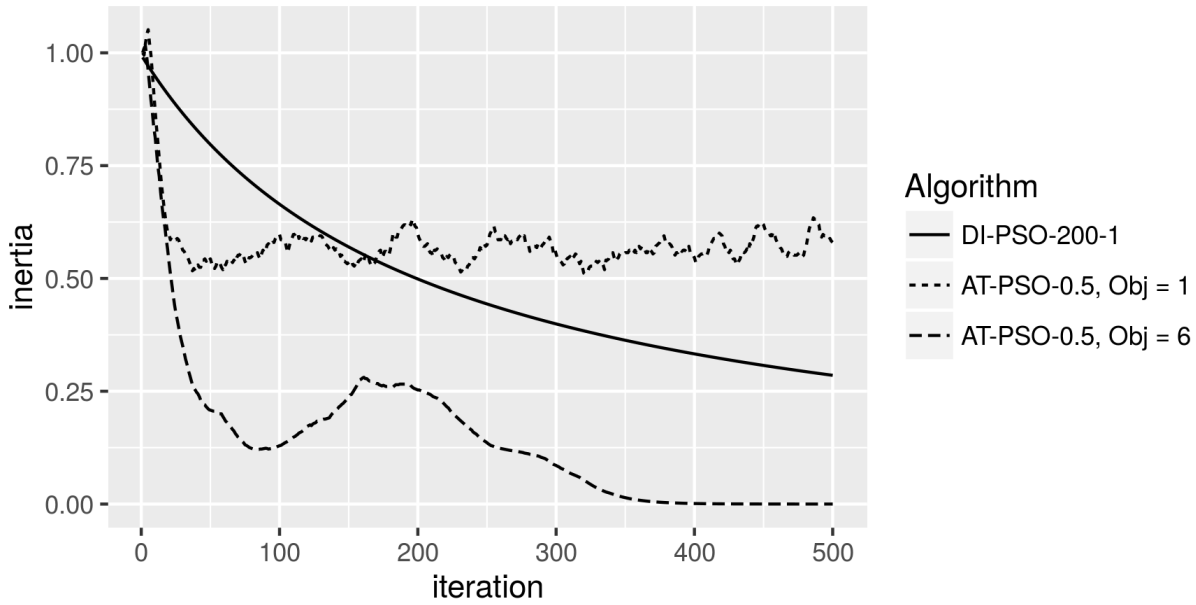

Figure 1. Inertia over time for the DI-PSO algorithm with $\alpha = 200$ and $\beta = 1$, and for one replication of the AT-PSO-0.5 algorithm for each of OFs 1 and 6.

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

Helwig, S & Wanka, R (2007), 'Particle swarm optimization in high-dimensional bounded search spaces,' in *2007 IEEE Swarm Intelligence Symposium*, IEEE, pp. 198–205.

Hsieh, HI & Lee, TS (2010), 'A modified algorithm of bare bones particle swarm optimization,' *International Journal of Computer Science Issues*, **7**, p. 11.

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

Monson, CK & Seppi, KD (2005), 'Exposing origin-seeking bias in pso,' in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, ACM, pp. 241–248.

Spears, WM, Green, DT & Spears, DF (2010), 'Biases in particle swarm optimization,' *International Journal of Swarm Intelligence Research*, **1**(2), pp. 34–57.

Zhang, H, Kennedy, DD, Rangaiah, GP & Bonilla-Petriciolet, A (2011), 'Novel bare-bones particle swarm optimization and its performance for modeling vapor–liquid equilibrium data,' *Fluid Phase Equilibria*, **301**(1), pp. 33–45.

OF1	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	4015.81	3124.33	0.00	∞	0.00	0.01	0.98	362.00	0.01	0.01	0.78	488.50
BBPSO-MC	84517.26	8944.33	0.00	∞	83674.80	9956.73	0.00	∞	84784.51	9451.47	0.00	∞
BBPSOxp-MC	84104.44	8192.44	0.00	∞	82952.29	9655.82	0.00	∞	85886.72	7182.36	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	1.59	0.47	0.00	∞	2.08	0.73	0.00	∞	2.10	0.89	0.00	∞
$df = 1,\ R^* = 0.3$	0.00	0.00	1.00	445.00	0.01	0.00	0.96	467.00	0.01	0.01	0.54	499.50
$df = 1,\ R^* = 0.5$	0.00	0.00	1.00	289.00	0.00	0.00	1.00	309.50	0.00	0.00	1.00	348.00
$df = 1,\ R^* = 0.7$	0.00	0.00	1.00	223.00	0.00	0.00	1.00	256.00	0.00	0.00	1.00	329.00
$df = 3,\ R^* = 0.1$	6.92	2.41	0.00	∞	6.92	1.79	0.00	∞	5.10	2.02	0.00	∞
$df = 3,\ R^* = 0.3$	0.01	0.00	0.22	∞	0.02	0.01	0.06	∞	0.03	0.01	0.00	∞
$df = 3,\ R^* = 0.5$	0.00	0.00	1.00	331.50	0.00	0.00	1.00	347.50	0.00	0.00	1.00	387.50
$df = 3,\ R^* = 0.7$	0.00	0.00	1.00	259.50	0.00	0.00	1.00	288.50	0.00	0.00	1.00	400.50
$df = 5,\ R^* = 0.1$	11.50	2.93	0.00	∞	11.72	2.53	0.00	∞	8.14	2.55	0.00	∞
$df = 5,\ R^* = 0.3$	0.03	0.01	0.00	∞	0.04	0.01	0.00	∞	0.06	0.02	0.00	∞
$df = 5,\ R^* = 0.5$	0.00	0.00	1.00	356.50	0.00	0.00	1.00	368.50	0.00	0.00	1.00	415.50
$df = 5,\ R^* = 0.7$	0.00	0.00	1.00	288.50	0.00	0.00	1.00	326.50	222.69	288.53	0.02	∞
$df = \infty,\ R^* = 0.1$	42.86	10.31	0.00	∞	38.44	8.12	0.00	∞	20.28	8.59	0.00	∞
$df = \infty,\ R^* = 0.3$	0.14	0.05	0.00	∞	0.15	0.03	0.00	∞	0.25	0.08	0.00	∞
$df = \infty,\ R^* = 0.5$	0.00	0.00	1.00	415.00	0.00	0.00	1.00	433.00	0.01	0.00	0.94	489.00
$df = \infty,\ R^* = 0.7$	0.00	0.00	1.00	352.00	0.00	0.00	0.98	441.50	16949.57	3141.36	0.00	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	5.71	1.58	0.00	∞	6.00	1.76	0.00	∞	3.91	1.75	0.00	∞
$df = 1,\ R^* = 0.3$	0.02	0.01	0.00	∞	0.04	0.01	0.00	∞	0.05	0.02	0.00	∞
$df = 1,\ R^* = 0.5$	0.00	0.00	1.00	358.00	0.00	0.00	1.00	381.00	0.00	0.00	1.00	419.50
$df = 1,\ R^* = 0.7$	0.00	0.00	1.00	291.00	0.00	0.00	1.00	326.00	0.00	0.01	0.98	432.50
$df = 3,\ R^* = 0.1$	17.42	4.87	0.00	∞	16.91	4.41	0.00	∞	8.64	3.12	0.00	∞
$df = 3,\ R^* = 0.3$	0.10	0.03	0.00	∞	0.12	0.04	0.00	∞	0.15	0.05	0.00	∞
$df = 3,\ R^* = 0.5$	0.00	0.00	1.00	412.00	0.00	0.00	1.00	431.50	0.01	0.00	0.92	470.50
$df = 3,\ R^* = 0.7$	0.00	0.00	1.00	350.50	0.00	0.01	0.98	438.00	4193.16	2013.59	0.00	∞
$df = 5,\ R^* = 0.1$	28.39	6.37	0.00	∞	28.42	7.58	0.00	∞	13.22	5.27	0.00	∞
$df = 5,\ R^* = 0.3$	0.19	0.05	0.00	∞	0.22	0.07	0.00	∞	0.26	0.10	0.00	∞
$df = 5,\ R^* = 0.5$	0.00	0.00	1.00	440.00	0.00	0.00	1.00	466.00	0.03	0.02	0.04	∞
$df = 5,\ R^* = 0.7$	0.00	0.00	1.00	405.50	505.43	755.11	0.02	∞	17674.52	4505.53	0.00	∞
$df = \infty,\ R^* = 0.1$	65.18	14.92	0.00	∞	60.90	13.82	0.00	∞	26.77	10.11	0.00	∞
$df = \infty,\ R^* = 0.3$	0.50	0.11	0.00	∞	0.66	0.14	0.00	∞	0.82	0.25	0.00	∞
$df = \infty,\ R^* = 0.5$	0.01	0.00	0.82	492.00	0.03	0.01	0.02	∞	0.85	0.75	0.00	∞
$df = \infty,\ R^* = 0.7$	0.19	0.33	0.04	∞	17369.34	3631.71	0.00	∞	35172.80	4377.92	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	6703.90	3993.99	0.00	∞	543.02	683.48	0.00	∞	51.63	95.71	0.00	∞
$\alpha = 50,\ \beta = 2$	7823.89	4787.27	0.00	∞	1821.42	1508.50	0.00	∞	499.64	441.33	0.00	∞
$\alpha = 50,\ \beta = 4$	15373.54	6906.92	0.00	∞	5572.40	3137.25	0.00	∞	3705.93	2324.41	0.00	∞
$\alpha = 100,\ \beta = 1$	3928.51	3855.22	0.00	∞	236.79	733.83	0.00	∞	0.99	4.03	0.20	∞
$\alpha = 100,\ \beta = 2$	7618.89	4250.92	0.00	∞	1206.09	1393.42	0.00	∞	84.99	137.32	0.00	∞
$\alpha = 100,\ \beta = 4$	13044.34	6195.43	0.00	∞	4218.13	2881.34	0.00	∞	1745.64	1532.00	0.00	∞
$\alpha = 200,\ \beta = 1$	2321.68	2088.02	0.00	∞	178.09	618.06	0.02	∞	0.00	0.00	1.00	383.50
$\alpha = 200,\ \beta = 2$	6427.62	3741.12	0.00	∞	305.62	523.71	0.00	∞	5.45	12.55	0.00	∞
$\alpha = 200,\ \beta = 4$	12004.75	5708.12	0.00	∞	2236.94	1729.58	0.00	∞	475.93	503.87	0.00	∞
AT-PSO												
$R^* = 0.1$	111.02	237.73	0.00	∞	211.79	889.18	0.00	∞	12223.23	9590.30	0.00	∞
$R^* = 0.3$	54.27	252.08	0.04	∞	0.00	0.00	1.00	347.00	0.00	0.01	0.88	457.00
$R^* = 0.5$	329.77	586.04	0.00	∞	0.00	0.00	1.00	274.50	0.00	0.00	1.00	303.50
$R^* = 0.7$	2131.94	2209.74	0.00	∞	0.04	0.21	0.92	381.50	0.00	0.00	1.00	341.50

Table 2. Simulation results for OF1. See text for description.

OF2	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	10915.05	17201.47	0.00	∞	1246.38	1002.75	0.00	∞	5738.34	3835.37	0.00	∞
BBPSO-MC	9081909.83	1958288.73	0.00	∞	9194595.70	2157840.19	0.00	∞	8830368.63	2234188.42	0.00	∞
BBPSOxp-MC	9138885.89	2265920.11	0.00	∞	8889637.56	2035964.50	0.00	∞	9038770.99	1858659.66	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	950.30	809.12	0.00	∞	690.95	535.97	0.00	∞	743.12	728.28	0.00	∞
$df = 1,\ R^* = 0.3$	300.03	258.36	0.00	∞	685.68	458.39	0.00	∞	1091.45	1375.54	0.00	∞
$df = 1,\ R^* = 0.5$	440.38	436.08	0.00	∞	1419.88	1239.43	0.00	∞	3652.45	4072.28	0.00	∞
$df = 1,\ R^* = 0.7$	1615.81	1539.22	0.00	∞	4330.31	4863.92	0.00	∞	8858.46	4578.27	0.00	∞
$df = 3,\ R^* = 0.1$	196.07	88.33	0.00	∞	226.17	163.16	0.00	∞	252.63	187.08	0.00	∞
$df = 3,\ R^* = 0.3$	23.27	17.11	0.00	∞	54.95	48.43	0.00	∞	189.83	399.03	0.00	∞
$df = 3,\ R^* = 0.5$	30.94	29.21	0.00	∞	185.87	161.77	0.00	∞	1073.22	1233.74	0.00	∞
$df = 3,\ R^* = 0.7$	222.12	524.53	0.00	∞	939.33	789.54	0.00	∞	5897.58	4513.86	0.00	∞
$df = 5,\ R^* = 0.1$	198.27	75.35	0.00	∞	175.31	80.21	0.00	∞	205.86	117.06	0.00	∞
$df = 5,\ R^* = 0.3$	12.25	9.35	0.00	∞	24.76	18.02	0.00	∞	60.79	63.04	0.00	∞
$df = 5,\ R^* = 0.5$	11.51	7.41	0.00	∞	63.91	51.53	0.00	∞	458.47	317.21	0.00	∞
$df = 5,\ R^* = 0.7$	61.67	36.27	0.00	∞	378.27	218.82	0.00	∞	7696.71	5485.99	0.00	∞
$df = \infty,\ R^* = 0.1$	280.79	97.77	0.00	∞	267.44	87.93	0.00	∞	287.37	128.28	0.00	∞
$df = \infty,\ R^* = 0.3$	6.75	3.83	0.00	∞	13.91	9.02	0.00	∞	39.37	32.01	0.00	∞
$df = \infty,\ R^* = 0.5$	4.96	3.73	0.00	∞	30.82	22.83	0.00	∞	269.58	161.10	0.00	∞
$df = \infty,\ R^* = 0.7$	51.31	38.92	0.00	∞	550.48	301.85	0.00	∞	1100097.72	493290.33	0.00	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	1620.04	724.59	0.00	∞	1434.81	637.94	0.00	∞	1004.77	608.51	0.00	∞
$df = 1,\ R^* = 0.3$	1117.65	614.18	0.00	∞	1541.55	913.10	0.00	∞	1372.47	1108.92	0.00	∞
$df = 1,\ R^* = 0.5$	2424.76	1458.65	0.00	∞	3639.53	2048.74	0.00	∞	5212.21	3221.11	0.00	∞
$df = 1,\ R^* = 0.7$	5685.79	3365.32	0.00	∞	6746.82	4254.76	0.00	∞	13170.58	7291.81	0.00	∞
$df = 3,\ R^* = 0.1$	609.04	289.92	0.00	∞	568.65	326.15	0.00	∞	562.17	301.77	0.00	∞
$df = 3,\ R^* = 0.3$	231.40	167.47	0.00	∞	320.46	308.01	0.00	∞	366.68	274.58	0.00	∞
$df = 3,\ R^* = 0.5$	360.34	224.38	0.00	∞	1026.42	709.48	0.00	∞	5089.66	5857.62	0.00	∞
$df = 3,\ R^* = 0.7$	1426.88	1209.92	0.00	∞	5568.83	7882.29	0.00	∞	38081.89	31747.85	0.00	∞
$df = 5,\ R^* = 0.1$	476.81	215.75	0.00	∞	497.74	194.10	0.00	∞	540.99	263.49	0.00	∞
$df = 5,\ R^* = 0.3$	109.01	45.98	0.00	∞	170.67	120.10	0.00	∞	272.26	192.17	0.00	∞
$df = 5,\ R^* = 0.5$	243.73	154.52	0.00	∞	448.77	241.51	0.00	∞	2147.17	1308.53	0.00	∞
$df = 5,\ R^* = 0.7$	794.68	388.65	0.00	∞	17971.50	16875.10	0.00	∞	507347.01	409181.08	0.00	∞
$df = \infty,\ R^* = 0.1$	609.19	197.59	0.00	∞	585.88	163.05	0.00	∞	536.88	263.19	0.00	∞
$df = \infty,\ R^* = 0.3$	70.82	44.69	0.00	∞	111.39	79.50	0.00	∞	220.30	168.06	0.00	∞
$df = \infty,\ R^* = 0.5$	114.86	88.86	0.00	∞	329.69	169.04	0.00	∞	2889.16	2088.86	0.00	∞
$df = \infty,\ R^* = 0.7$	2469.49	2799.80	0.00	∞	1483405.94	407996.51	0.00	∞	2535125.64	778910.46	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	40714.56	24208.86	0.00	∞	16548.67	9312.34	0.00	∞	7839.43	5031.49	0.00	∞
$\alpha = 50,\ \beta = 2$	47763.21	31361.92	0.00	∞	24627.59	11610.10	0.00	∞	11660.25	6362.56	0.00	∞
$\alpha = 50,\ \beta = 4$	62225.74	28459.35	0.00	∞	31511.13	15145.07	0.00	∞	16416.52	7831.71	0.00	∞
$\alpha = 100,\ \beta = 1$	26613.11	14611.63	0.00	∞	10168.37	6029.10	0.00	∞	5378.87	3302.21	0.00	∞
$\alpha = 100,\ \beta = 2$	46177.71	37974.29	0.00	∞	17795.63	8897.56	0.00	∞	9818.51	4708.72	0.00	∞
$\alpha = 100,\ \beta = 4$	64833.73	36534.52	0.00	∞	28627.71	14257.04	0.00	∞	14536.17	7921.94	0.00	∞
$\alpha = 200,\ \beta = 1$	21611.55	12914.90	0.00	∞	5699.97	3916.44	0.00	∞	3499.49	1865.32	0.00	∞
$\alpha = 200,\ \beta = 2$	36405.44	27997.69	0.00	∞	12492.90	6829.93	0.00	∞	6944.87	4559.84	0.00	∞
$\alpha = 200,\ \beta = 4$	57723.16	30860.32	0.00	∞	21807.81	11567.18	0.00	∞	12201.41	5365.69	0.00	∞
AT-PSO												
$R^* = 0.1$	8823.66	9727.43	0.00	∞	8852.96	6912.25	0.00	∞	58369.56	38288.17	0.00	∞
$R^* = 0.3$	6658.84	6659.87	0.00	∞	769.74	850.01	0.00	∞	5183.03	4046.47	0.00	∞
$R^* = 0.5$	20994.61	16398.66	0.00	∞	1713.89	2589.20	0.00	∞	1877.08	1465.51	0.00	∞
$R^* = 0.7$	35128.87	17886.90	0.00	∞	7126.76	5756.50	0.00	∞	5771.37	3581.95	0.00	∞

Table 3. Simulation results for OF2. See text for description.

OF3	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	1242764.67	2075592.00	0.00	∞	121.33	104.18	0.00	∞	241.45	255.69	0.00	∞
BBPSO-MC	232830400.95	37473167.09	0.00	∞	247662755.14	43820300.43	0.00	∞	247321508.63	39096259.33	0.00	∞
BBPSOxp-MC	235491326.54	42124202.61	0.00	∞	242679907.63	42872974.55	0.00	∞	253599464.23	43615037.04	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	382.54	325.35	0.00	∞	486.70	533.46	0.00	∞	669.09	932.66	0.00	∞
$df = 1,\ R^* = 0.3$	233.17	429.80	0.00	∞	182.89	272.06	0.00	∞	352.64	465.63	0.00	∞
$df = 1,\ R^* = 0.5$	315.43	497.55	0.00	∞	159.55	216.03	0.00	∞	254.78	669.72	0.00	∞
$df = 1,\ R^* = 0.7$	235.08	455.38	0.00	∞	96.32	106.25	0.00	∞	189.20	288.45	0.00	∞
$df = 3,\ R^* = 0.1$	545.05	481.33	0.00	∞	629.79	657.37	0.00	∞	567.75	585.07	0.00	∞
$df = 3,\ R^* = 0.3$	176.81	186.22	0.00	∞	172.84	204.18	0.00	∞	256.32	426.46	0.00	∞
$df = 3,\ R^* = 0.5$	159.09	170.77	0.00	∞	128.99	127.25	0.00	∞	137.56	202.68	0.00	∞
$df = 3,\ R^* = 0.7$	155.74	399.33	0.00	∞	132.16	339.64	0.00	∞	82.12	170.21	0.00	∞
$df = 5,\ R^* = 0.1$	833.71	627.25	0.00	∞	928.42	982.63	0.00	∞	450.93	444.86	0.00	∞
$df = 5,\ R^* = 0.3$	283.78	431.37	0.00	∞	135.95	98.01	0.00	∞	229.39	314.25	0.00	∞
$df = 5,\ R^* = 0.5$	198.53	185.97	0.00	∞	173.66	278.73	0.00	∞	118.79	114.49	0.00	∞
$df = 5,\ R^* = 0.7$	142.06	190.31	0.00	∞	64.57	98.26	0.00	∞	108.71	538.33	0.00	∞
$df = \infty,\ R^* = 0.1$	1343.31	757.87	0.00	∞	1056.96	522.89	0.00	∞	958.18	773.33	0.00	∞
$df = \infty,\ R^* = 0.3$	230.32	320.08	0.00	∞	138.02	120.68	0.00	∞	212.78	331.90	0.00	∞
$df = \infty,\ R^* = 0.5$	225.28	219.75	0.00	∞	156.78	132.24	0.00	∞	120.23	106.23	0.00	∞
$df = \infty,\ R^* = 0.7$	131.34	91.53	0.00	∞	56.10	76.80	0.00	∞	179.69	683.04	0.00	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	616.80	278.83	0.00	∞	633.00	276.25	0.00	∞	490.75	350.79	0.00	∞
$df = 1,\ R^* = 0.3$	152.80	179.75	0.00	∞	163.64	131.42	0.00	∞	140.13	158.06	0.00	∞
$df = 1,\ R^* = 0.5$	139.49	269.65	0.00	∞	99.39	225.41	0.00	∞	202.26	429.55	0.00	∞
$df = 1,\ R^* = 0.7$	149.01	459.29	0.00	∞	73.54	126.07	0.00	∞	188.28	370.67	0.00	∞
$df = 3,\ R^* = 0.1$	961.94	370.97	0.00	∞	931.38	294.49	0.00	∞	709.41	452.45	0.00	∞
$df = 3,\ R^* = 0.3$	168.44	279.24	0.00	∞	144.15	200.78	0.00	∞	120.33	48.68	0.00	∞
$df = 3,\ R^* = 0.5$	123.39	102.98	0.00	∞	97.56	50.88	0.00	∞	77.47	81.83	0.00	∞
$df = 3,\ R^* = 0.7$	49.75	70.72	0.00	∞	33.96	52.12	0.00	∞	3586.02	9973.24	0.00	∞
$df = 5,\ R^* = 0.1$	1229.87	526.97	0.00	∞	1171.93	406.81	0.00	∞	727.95	436.07	0.00	∞
$df = 5,\ R^* = 0.3$	267.86	435.67	0.00	∞	124.80	75.86	0.00	∞	157.72	217.77	0.00	∞
$df = 5,\ R^* = 0.5$	137.08	104.13	0.00	∞	91.44	86.38	0.00	∞	75.74	95.04	0.00	∞
$df = 5,\ R^* = 0.7$	66.87	126.23	0.00	∞	19.28	17.08	0.00	∞	82498.29	102684.12	0.00	∞
$df = \infty,\ R^* = 0.1$	2002.70	822.46	0.00	∞	1718.67	698.34	0.00	∞	1158.47	903.16	0.00	∞
$df = \infty,\ R^* = 0.3$	137.09	84.51	0.00	∞	116.46	66.60	0.00	∞	114.47	57.68	0.00	∞
$df = \infty,\ R^* = 0.5$	121.71	103.55	0.00	∞	104.78	67.31	0.00	∞	41.15	33.00	0.00	∞
$df = \infty,\ R^* = 0.7$	24.31	24.58	0.00	∞	24.31	20.57	0.00	∞	2088154.00	1258187.91	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	4180196.82	4677639.48	0.00	∞	354071.40	579847.34	0.00	∞	70128.62	165037.08	0.00	∞
$\alpha = 50,\ \beta = 2$	6964209.15	8322398.93	0.00	∞	2341768.78	4277548.98	0.00	∞	1232022.80	1464906.82	0.00	∞
$\alpha = 50,\ \beta = 4$	20143670.82	15026478.57	0.00	∞	11740754.91	8374232.25	0.00	∞	6766696.51	4472751.26	0.00	∞
$\alpha = 100,\ \beta = 1$	1610827.18	3342137.70	0.00	∞	55104.82	171942.01	0.00	∞	1691.16	3160.57	0.00	∞
$\alpha = 100,\ \beta = 2$	6945649.64	6844592.19	0.00	∞	1001725.17	1774982.73	0.00	∞	150478.59	213009.16	0.00	∞
$\alpha = 100,\ \beta = 4$	21536033.79	15337048.80	0.00	∞	6691579.83	5932202.88	0.00	∞	4328277.03	4060546.38	0.00	∞
$\alpha = 200,\ \beta = 1$	1543367.07	2077471.63	0.00	∞	4350.55	16257.10	0.00	∞	266.44	453.75	0.00	∞
$\alpha = 200,\ \beta = 2$	6626917.67	9093854.31	0.00	∞	299834.83	1040979.02	0.00	∞	5848.88	14329.18	0.00	∞
$\alpha = 200,\ \beta = 4$	17830796.63	14245007.26	0.00	∞	3367710.63	3194612.95	0.00	∞	776424.69	865443.75	0.00	∞
AT-PSO												
$R^* = 0.1$	5779.95	14928.86	0.00	∞	297272.41	1261108.56	0.00	∞	33785179.04	23020834.78	0.00	∞
$R^* = 0.3$	869.69	2100.98	0.00	∞	131.15	175.33	0.00	∞	184.15	186.17	0.00	∞
$R^* = 0.5$	71363.37	218211.42	0.00	∞	190.13	271.20	0.00	∞	134.32	200.92	0.00	∞
$R^* = 0.7$	1572669.78	3884482.27	0.00	∞	391.10	857.02	0.00	∞	535.09	1277.83	0.00	∞

Table 4. Simulation results for OF3. See text for description.

OF4	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	29.94	15.21	0.00	∞	5.37	3.47	0.02	∞	3.24	1.44	0.00	∞
BBPSO-MC	237.31	18.67	0.00	∞	232.86	24.40	0.00	∞	232.69	21.86	0.00	∞
BBPSOxp-MC	229.80	22.24	0.00	∞	236.60	25.20	0.00	∞	230.84	21.05	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	4.96	1.58	0.00	∞	3.93	1.29	0.00	∞	4.07	1.88	0.00	∞
$df = 1,\ R^* = 0.3$	5.14	1.89	0.00	∞	2.76	1.45	0.06	∞	2.48	1.57	0.06	∞
$df = 1,\ R^* = 0.5$	6.32	2.16	0.00	∞	4.68	1.94	0.00	∞	4.24	2.36	0.02	∞
$df = 1,\ R^* = 0.7$	8.56	2.84	0.00	∞	5.78	1.67	0.00	∞	6.03	2.01	0.00	∞
$df = 3,\ R^* = 0.1$	5.44	1.30	0.00	∞	5.40	1.17	0.00	∞	5.39	1.54	0.00	∞
$df = 3,\ R^* = 0.3$	3.56	1.76	0.00	∞	1.59	1.24	0.20	∞	1.43	1.00	0.18	∞
$df = 3,\ R^* = 0.5$	4.07	1.85	0.00	∞	2.70	1.57	0.08	∞	2.57	1.44	0.02	∞
$df = 3,\ R^* = 0.7$	4.70	1.73	0.00	∞	3.83	1.72	0.00	∞	3.43	1.75	0.06	∞
$df = 5,\ R^* = 0.1$	6.63	1.34	0.00	∞	6.45	1.92	0.00	∞	6.72	1.86	0.00	∞
$df = 5,\ R^* = 0.3$	2.80	1.51	0.02	∞	1.24	1.12	0.24	∞	1.13	1.21	0.30	∞
$df = 5,\ R^* = 0.5$	4.11	2.16	0.02	∞	2.26	1.60	0.12	∞	1.90	1.43	0.16	∞
$df = 5,\ R^* = 0.7$	4.40	2.00	0.00	∞	2.91	1.48	0.00	∞	3.56	1.84	0.00	∞
$df = \infty,\ R^* = 0.1$	9.12	1.88	0.00	∞	9.17	1.58	0.00	∞	9.42	1.70	0.00	∞
$df = \infty,\ R^* = 0.3$	2.15	1.31	0.00	∞	0.72	0.92	0.00	∞	0.61	0.74	0.04	∞
$df = \infty,\ R^* = 0.5$	2.74	1.55	0.04	∞	1.62	1.38	0.18	∞	1.01	0.91	0.26	∞
$df = \infty,\ R^* = 0.7$	3.22	1.78	0.06	∞	2.11	1.56	0.18	∞	2.13	1.30	0.10	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	4.25	1.24	0.00	∞	4.70	1.25	0.00	∞	6.34	1.56	0.00	∞
$df = 1,\ R^* = 0.3$	0.74	0.87	0.04	∞	0.23	0.44	0.02	∞	0.51	0.74	0.00	∞
$df = 1,\ R^* = 0.5$	1.75	1.24	0.12	∞	0.95	1.00	0.38	∞	1.41	1.16	0.18	∞
$df = 1,\ R^* = 0.7$	3.07	1.65	0.02	∞	2.14	1.20	0.06	∞	3.10	1.62	0.02	∞
$df = 3,\ R^* = 0.1$	5.34	1.01	0.00	∞	6.29	1.31	0.00	∞	7.39	1.59	0.00	∞
$df = 3,\ R^* = 0.3$	0.25	0.45	0.00	∞	0.08	0.23	0.00	∞	0.19	0.41	0.02	∞
$df = 3,\ R^* = 0.5$	0.65	0.78	0.52	345.50	0.30	0.52	0.72	342.00	0.56	0.66	0.46	∞
$df = 3,\ R^* = 0.7$	1.43	1.21	0.22	∞	1.24	1.02	0.24	∞	1.85	1.12	0.08	∞
$df = 5,\ R^* = 0.1$	6.02	1.08	0.00	∞	6.50	1.08	0.00	∞	7.94	1.31	0.00	∞
$df = 5,\ R^* = 0.3$	0.10	0.26	0.00	∞	0.08	0.23	0.00	∞	0.10	0.33	0.02	∞
$df = 5,\ R^* = 0.5$	0.40	0.61	0.66	338.00	0.25	0.42	0.74	338.50	0.46	0.70	0.64	367.00
$df = 5,\ R^* = 0.7$	1.13	1.01	0.28	∞	0.75	0.85	0.44	∞	1.74	1.10	0.04	∞
$df = \infty,\ R^* = 0.1$	7.81	1.19	0.00	∞	8.10	1.36	0.00	∞	9.02	1.36	0.00	∞
$df = \infty,\ R^* = 0.3$	0.09	0.23	0.00	∞	0.03	0.01	0.00	∞	0.05	0.13	0.00	∞
$df = \infty,\ R^* = 0.5$	0.25	0.57	0.80	346.00	0.13	0.33	0.86	345.00	0.17	0.42	0.84	352.50
$df = \infty,\ R^* = 0.7$	0.90	0.89	0.36	∞	0.77	0.92	0.46	∞	2.66	1.39	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	32.08	11.51	0.00	∞	13.04	5.65	0.00	∞	6.33	3.37	0.00	∞
$\alpha = 50,\ \beta = 2$	38.18	17.28	0.00	∞	15.90	5.67	0.00	∞	10.13	4.19	0.00	∞
$\alpha = 50,\ \beta = 4$	56.49	22.89	0.00	∞	28.28	11.13	0.00	∞	15.83	7.08	0.00	∞
$\alpha = 100, \beta = 1$	28.51	12.71	0.00	∞	11.32	5.72	0.00	∞	4.66	2.48	0.00	∞
$\alpha = 100, \beta = 2$	32.85	13.96	0.00	∞	14.35	7.57	0.00	∞	6.86	2.78	0.00	∞
$\alpha = 100, \beta = 4$	49.67	16.91	0.00	∞	24.47	9.82	0.00	∞	13.74	6.18	0.00	∞
$\alpha = 200, \beta = 1$	22.39	12.14	0.00	∞	7.53	4.31	0.00	∞	3.92	2.11	0.02	∞
$\alpha = 200, \beta = 2$	36.56	14.29	0.00	∞	11.21	5.42	0.00	∞	5.28	2.77	0.00	∞
$\alpha = 200, \beta = 4$	55.67	19.91	0.00	∞	19.67	8.61	0.00	∞	11.18	5.41	0.00	∞
AT-PSO												
$R^* = 0.1$	7.73	5.50	0.00	∞	5.72	1.90	0.00	∞	23.02	10.87	0.00	∞
$R^* = 0.3$	14.10	6.74	0.00	∞	6.67	3.74	0.00	∞	3.77	2.04	0.02	∞
$R^* = 0.5$	23.89	8.43	0.00	∞	11.39	5.91	0.00	∞	6.06	2.92	0.00	∞
$R^* = 0.7$	34.91	12.20	0.00	∞	14.91	5.84	0.00	∞	9.80	3.78	0.00	∞

Table 5. Simulation results for OF4. See text for description.

OF5	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	27.55	20.12	0.00	∞	0.03	0.03	0.28	∞	0.08	0.07	0.04	∞
BBPSO-MC	762.36	82.36	0.00	∞	785.33	69.84	0.00	∞	751.88	80.37	0.00	∞
BBPSOxp-MC	766.30	80.16	0.00	∞	785.43	64.92	0.00	∞	771.17	76.11	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	0.83	0.10	0.00	∞	0.88	0.09	0.00	∞	0.92	0.06	0.00	∞
$df = 1,\ R^* = 0.3$	0.02	0.01	0.24	∞	0.02	0.01	0.34	∞	0.05	0.03	0.02	∞
$df = 1,\ R^* = 0.5$	0.01	0.01	0.52	455.50	0.01	0.01	0.56	455.00	0.01	0.01	0.48	∞
$df = 1,\ R^* = 0.7$	0.02	0.01	0.36	∞	0.02	0.02	0.46	∞	0.04	0.04	0.16	∞
$df = 3,\ R^* = 0.1$	1.06	0.02	0.00	∞	1.05	0.02	0.00	∞	1.04	0.02	0.00	∞
$df = 3,\ R^* = 0.3$	0.08	0.03	0.00	∞	0.12	0.04	0.00	∞	0.32	0.15	0.00	∞
$df = 3,\ R^* = 0.5$	0.01	0.01	0.54	459.00	0.01	0.01	0.62	481.00	0.09	0.07	0.02	∞
$df = 3,\ R^* = 0.7$	0.01	0.01	0.46	∞	0.12	0.15	0.04	∞	205.76	57.74	0.00	∞
$df = 5,\ R^* = 0.1$	1.11	0.03	0.00	∞	1.11	0.03	0.00	∞	1.09	0.03	0.00	∞
$df = 5,\ R^* = 0.3$	0.18	0.05	0.00	∞	0.29	0.09	0.00	∞	0.66	0.15	0.00	∞
$df = 5,\ R^* = 0.5$	0.02	0.01	0.32	∞	0.02	0.01	0.26	∞	0.69	0.21	0.00	∞
$df = 5,\ R^* = 0.7$	0.01	0.01	0.32	∞	39.11	26.34	0.00	∞	480.85	48.89	0.00	∞
$df = \infty,\ R^* = 0.1$	1.47	0.11	0.00	∞	1.45	0.09	0.00	∞	1.25	0.08	0.00	∞
$df = \infty,\ R^* = 0.3$	0.69	0.11	0.00	∞	0.85	0.07	0.00	∞	0.97	0.05	0.00	∞
$df = \infty,\ R^* = 0.5$	0.04	0.02	0.00	∞	0.26	0.09	0.00	∞	9.61	10.88	0.00	∞
$df = \infty,\ R^* = 0.7$	0.82	0.19	0.00	∞	395.75	42.10	0.00	∞	638.33	59.56	0.00	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	1.01	0.02	0.00	∞	1.01	0.02	0.00	∞	1.00	0.03	0.00	∞
$df = 1,\ R^* = 0.3$	0.07	0.04	0.00	∞	0.13	0.06	0.00	∞	0.25	0.13	0.00	∞
$df = 1,\ R^* = 0.5$	0.01	0.01	0.54	465.50	0.01	0.01	0.44	∞	0.06	0.08	0.06	∞
$df = 1,\ R^* = 0.7$	0.02	0.02	0.46	∞	0.02	0.03	0.46	∞	1.20	0.54	0.00	∞
$df = 3,\ R^* = 0.1$	1.17	0.04	0.00	∞	1.16	0.04	0.00	∞	1.09	0.03	0.00	∞
$df = 3,\ R^* = 0.3$	0.56	0.12	0.00	∞	0.73	0.08	0.00	∞	0.87	0.09	0.00	∞
$df = 3,\ R^* = 0.5$	0.04	0.02	0.00	∞	0.33	0.14	0.00	∞	5.84	7.01	0.00	∞
$df = 3,\ R^* = 0.7$	7.20	9.68	0.00	∞	295.56	44.31	0.00	∞	439.15	53.40	0.00	∞
$df = 5,\ R^* = 0.1$	1.26	0.06	0.00	∞	1.30	0.06	0.00	∞	1.16	0.06	0.00	∞
$df = 5,\ R^* = 0.3$	0.80	0.08	0.00	∞	0.92	0.05	0.00	∞	1.00	0.02	0.00	∞
$df = 5,\ R^* = 0.5$	0.22	0.10	0.00	∞	0.93	0.09	0.00	∞	72.53	27.81	0.00	∞
$df = 5,\ R^* = 0.7$	172.86	53.14	0.00	∞	491.44	39.13	0.00	∞	527.93	56.57	0.00	∞
$df = \infty,\ R^* = 0.1$	1.71	0.13	0.00	∞	1.70	0.17	0.00	∞	1.34	0.14	0.00	∞
$df = \infty,\ R^* = 0.3$	0.99	0.04	0.00	∞	1.03	0.01	0.00	∞	1.06	0.03	0.00	∞
$df = \infty,\ R^* = 0.5$	0.82	0.12	0.00	∞	2.65	1.75	0.00	∞	255.87	50.83	0.00	∞
$df = \infty,\ R^* = 0.7$	451.87	38.28	0.00	∞	622.22	35.60	0.00	∞	586.09	47.80	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	55.37	25.87	0.00	∞	7.99	9.84	0.00	∞	1.07	0.35	0.00	∞
$\alpha = 50,\ \beta = 2$	73.04	43.14	0.00	∞	21.12	16.40	0.00	∞	6.20	4.72	0.00	∞
$\alpha = 50,\ \beta = 4$	142.00	66.28	0.00	∞	61.48	36.04	0.00	∞	27.71	18.29	0.00	∞
$\alpha = 100, \beta = 1$	32.00	20.76	0.00	∞	2.52	2.93	0.00	∞	0.22	0.19	0.00	∞
$\alpha = 100, \beta = 2$	78.45	51.91	0.00	∞	12.10	18.31	0.00	∞	1.76	1.50	0.00	∞
$\alpha = 100, \beta = 4$	139.13	62.71	0.00	∞	42.32	21.23	0.00	∞	15.40	12.26	0.00	∞
$\alpha = 200, \beta = 1$	28.74	20.48	0.00	∞	0.56	0.70	0.00	∞	0.06	0.07	0.18	∞
$\alpha = 200, \beta = 2$	76.70	56.65	0.00	∞	4.61	6.06	0.00	∞	0.52	0.34	0.00	∞
$\alpha = 200, \beta = 4$	130.62	67.27	0.00	∞	21.48	17.97	0.00	∞	5.24	5.82	0.00	∞
AT-PSO												
$R^* = 0.1$	3.28	6.03	0.00	∞	2.22	2.69	0.00	∞	111.55	65.99	0.00	∞
$R^* = 0.3$	0.74	1.00	0.02	∞	0.02	0.02	0.34	∞	0.06	0.06	0.16	∞
$R^* = 0.5$	4.06	8.44	0.00	∞	0.07	0.13	0.22	∞	0.01	0.02	0.56	414.50
$R^* = 0.7$	28.82	33.05	0.00	∞	0.43	1.25	0.04	∞	0.07	0.16	0.22	∞

Table 6. Simulation results for OF5. See text for description.

OF6	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$	Mean	SD	$\widehat{p}$	$\widehat{k}$
PSO	19.66	0.56	0.00	∞	14.77	8.35	0.02	∞	19.48	2.63	0.00	∞
BBPSO-MC	20.72	0.23	0.00	∞	20.74	0.22	0.00	∞	20.74	0.21	0.00	∞
BBPSOxp-MC	20.75	0.18	0.00	∞	20.73	0.18	0.00	∞	20.73	0.18	0.00	∞
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	18.35	5.39	0.00	∞	9.73	9.19	0.00	∞	6.24	7.88	0.00	∞
$df = 1,\ R^* = 0.3$	18.65	4.66	0.00	∞	14.66	8.70	0.00	∞	11.40	9.71	0.00	∞
$df = 1,\ R^* = 0.5$	19.93	0.37	0.00	∞	18.90	4.71	0.04	∞	16.91	6.85	0.06	∞
$df = 1,\ R^* = 0.7$	19.69	2.00	0.00	∞	19.52	2.28	0.00	∞	18.05	5.13	0.00	∞
$df = 3,\ R^* = 0.1$	19.98	0.09	0.00	∞	19.48	3.55	0.00	∞	14.60	8.47	0.00	∞
$df = 3,\ R^* = 0.3$	19.82	0.05	0.00	∞	19.48	2.82	0.00	∞	19.94	0.54	0.00	∞
$df = 3,\ R^* = 0.5$	19.83	0.06	0.00	∞	19.95	0.30	0.00	∞	20.24	0.38	0.00	∞
$df = 3,\ R^* = 0.7$	19.85	0.10	0.00	∞	20.05	0.37	0.00	∞	20.34	0.39	0.00	∞
$df = 5,\ R^* = 0.1$	20.00	0.07	0.00	∞	20.12	0.25	0.00	∞	17.83	6.21	0.00	∞
$df = 5,\ R^* = 0.3$	19.83	0.04	0.00	∞	19.87	0.21	0.00	∞	20.12	0.45	0.00	∞
$df = 5,\ R^* = 0.5$	19.83	0.04	0.00	∞	19.90	0.26	0.00	∞	20.32	0.45	0.00	∞
$df = 5,\ R^* = 0.7$	19.85	0.12	0.00	∞	20.06	0.35	0.00	∞	20.42	0.39	0.00	∞
$df = \infty,\ R^* = 0.1$	20.09	0.08	0.00	∞	20.16	0.14	0.00	∞	20.32	0.27	0.00	∞
$df = \infty,\ R^* = 0.3$	19.83	0.03	0.00	∞	19.85	0.17	0.00	∞	19.98	0.33	0.00	∞
$df = \infty,\ R^* = 0.5$	19.83	0.03	0.00	∞	19.89	0.20	0.00	∞	20.14	0.38	0.00	∞
$df = \infty,\ R^* = 0.7$	19.83	0.06	0.00	∞	20.06	0.36	0.00	∞	20.36	0.37	0.00	∞
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	15.26	7.90	0.00	∞	11.33	8.97	0.00	∞	9.05	8.54	0.00	∞
$df = 1,\ R^* = 0.3$	18.42	5.50	0.00	∞	17.06	6.87	0.00	∞	14.76	8.83	0.00	∞
$df = 1,\ R^* = 0.5$	20.19	0.20	0.00	∞	19.48	3.44	0.02	∞	17.05	6.20	0.00	∞
$df = 1,\ R^* = 0.7$	20.22	0.26	0.00	∞	20.08	0.67	0.00	∞	19.74	2.08	0.00	∞
$df = 3,\ R^* = 0.1$	20.19	0.16	0.00	∞	20.26	0.13	0.00	∞	19.58	3.36	0.00	∞
$df = 3,\ R^* = 0.3$	20.08	0.17	0.00	∞	20.23	0.18	0.00	∞	20.27	0.18	0.00	∞
$df = 3,\ R^* = 0.5$	20.19	0.13	0.00	∞	20.27	0.13	0.00	∞	20.33	0.25	0.00	∞
$df = 3,\ R^* = 0.7$	20.21	0.13	0.00	∞	20.32	0.11	0.00	∞	20.39	0.11	0.00	∞
$df = 5,\ R^* = 0.1$	20.25	0.08	0.00	∞	20.27	0.11	0.00	∞	20.32	0.12	0.00	∞
$df = 5,\ R^* = 0.3$	20.07	0.16	0.00	∞	20.20	0.17	0.00	∞	20.21	0.18	0.00	∞
$df = 5,\ R^* = 0.5$	20.14	0.15	0.00	∞	20.23	0.12	0.00	∞	20.33	0.14	0.00	∞
$df = 5,\ R^* = 0.7$	20.20	0.11	0.00	∞	20.31	0.13	0.00	∞	20.37	0.11	0.00	∞
$df = \infty,\ R^* = 0.1$	20.18	0.08	0.00	∞	20.27	0.08	0.00	∞	20.29	0.12	0.00	∞
$df = \infty,\ R^* = 0.3$	20.01	0.17	0.00	∞	20.14	0.18	0.00	∞	20.20	0.19	0.00	∞
$df = \infty,\ R^* = 0.5$	20.06	0.15	0.00	∞	20.22	0.14	0.00	∞	20.27	0.15	0.00	∞
$df = \infty,\ R^* = 0.7$	20.14	0.15	0.00	∞	20.30	0.13	0.00	∞	20.36	0.14	0.00	∞
DI-PSO												
$\alpha = 50,\ \beta = 1$	19.42	1.48	0.00	∞	18.52	3.08	0.00	∞	19.02	3.09	0.00	∞
$\alpha = 50,\ \beta = 2$	19.50	1.04	0.00	∞	17.94	3.01	0.00	∞	19.33	1.89	0.00	∞
$\alpha = 50,\ \beta = 4$	19.83	0.64	0.00	∞	19.66	1.00	0.00	∞	19.76	1.07	0.00	∞
$\alpha = 100, \beta = 1$	18.71	2.25	0.00	∞	16.04	5.48	0.00	∞	17.63	5.10	0.00	∞
$\alpha = 100, \beta = 2$	19.75	1.02	0.00	∞	19.27	1.95	0.00	∞	18.96	3.30	0.00	∞
$\alpha = 100, \beta = 4$	19.99	1.04	0.00	∞	19.80	1.11	0.00	∞	20.13	0.75	0.00	∞
$\alpha = 200, \beta = 1$	19.13	2.10	0.00	∞	14.83	6.47	0.00	∞	16.77	5.92	0.00	∞
$\alpha = 200, \beta = 2$	20.00	1.29	0.00	∞	19.63	2.37	0.00	∞	20.27	0.32	0.00	∞
$\alpha = 200, \beta = 4$	20.38	0.65	0.00	∞	20.19	0.65	0.00	∞	20.47	0.31	0.00	∞
AT-PSO												
$R^* = 0.1$	18.67	4.43	0.00	∞	18.18	5.06	0.00	∞	20.18	0.66	0.00	∞
$R^* = 0.3$	19.44	1.34	0.00	∞	16.84	6.30	0.04	∞	15.62	7.72	0.00	∞
$R^* = 0.5$	19.69	0.92	0.00	∞	18.88	3.16	0.00	∞	19.66	2.29	0.00	∞
$R^* = 0.7$	19.97	0.53	0.00	∞	19.76	0.69	0.00	∞	20.12	0.32	0.00	∞

Table 7. Simulation results for OF6. See text for description.