Particle Swarm Optimization Assisted Metropolis Hastings Algorithms

A PSO and BBPSO details

[THIS IS JUST PASTED TEXT THAT CAME FROM THE MAIN BODY OF THE DOCUMENT - SECTION NEEDS TO BE WRITTEN IN EARNEST. INCLUDE RING TOPOLOGY STUFF HERE.]

A downside of both versions of BBPSO above is that any particle currently at 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., see Hsieh and 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, the group best particle randomly selects three other distinct particles i_1, i_2, and i_3, and updates:

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

When combined with BBPSOxp, this becomes

$$\theta_{ij}(t+1) = \begin{cases} p_{i_1 j}(t) + 0.5(p_{i_2 j}(t) - p_{i_3 j}(t)) & \text{with probability } 0.5 \\ g_{ij}(t) & \text{otherwise,} \end{cases} \tag{2}$$

but it also introduces another problem — a particle which moves to its group best location on a single coordinate will have a standard deviation of zero for that coordinate during the next iteration. This is easily overcome by manually setting the standard deviation to a small value (e.g., 0.001), which allows us to define the two base BBPSO algorithms that we will consider: BBPSO-MC and BBPSOxp-MC. BBPSO-MC is standard BBPSO where each particle evolves according to (2) except any current group best particle evolves according to (1). BBPSOxp-MC evolves every particle according to

$$\theta_{ij}(t+1) = \begin{cases} N\left(\frac{p_{ij}(t)+g_{ij}(t)}{2}, \sigma_{ij}^2(t)\right) & \text{with probability } 0.5 \\ g_{ij}(t) & \text{otherwise,} \end{cases} \tag{3}$$

where $\sigma_{ij}(t) = |p_{ij}(t) - g_{ij}(t)|$ if $|p_{ij}(t) - g_{ij}(t)| > 0$ and $\sigma_{ij}(t) = 0.001$ otherwise, except current group best particles evolve according to (2).

B Comparing AT-BBPSO, BBPSO, and PSO algorithms

In order to compare AT-BBPSO to other PSO variants, we employ a subset of test functions used in Hsieh and 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). Each function is randomly initialized in a range that does not contain its global maximum.

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

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 and Li (2008) and Clerc and 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 increment or decrement $\log \sigma$ by $c = 0.1$

every iteration. The AT-PSO variants are initialized at $\omega(0) = 1$ and $\log \omega$ is incremented or decremented by $c = 0.1$ every iteration. In addition, each algorithm is implemented using each of three neighborhood structures. The global neighborhood allows each particle to look at each other particle in the swarm in order to determine its group best — in this case the group best is the swarm best and is used by all particles. The ring-1 neighborhood only allows particles to look at their neighbors as defined by a ring structure. Label each particle with an integer, $0, 1, \ldots, n-1$. Then particle i only looks at particles $i-1$ mod n and $i+1$ mod n in order to determine what its group best is. Addition and subtraction is mod n so that particle 0 looks at particles $n-1$ and 1 to determine its group best. This restricts the flow of information across particles allowing them to explore their nearby space more fully before being pushed in the direction of the rest of the swarm. For less well behaved functions this behavior typically improves the algorithm's ability to find the global max by allowing for more exploration of the objective surface and less exploitation of the best known information. Finally the ring-3 neighborhood is similar to ring-1 but allows each particle to look at its 3 nearest neighbors in each direction, so, for example, particle 1 sees particles 2, 3, 4, n_{swarm}, $n_{swarm} - 1$, and $n_{swarm} - 2$. This still restricts the flow of information across the swarm, but less so than the ring-1 neighborhood.

Each algorithm was used to optimize each objective function for 500 iterations over 100 replications. Initializations were changed across replications but held constant across algorithms. The standard PSO, DI-PSO, and AT-PSO algorithms initialized their velocity terms with $v_{ij}(0) \overset{iid}{\sim} U(-1, 1)$. 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 replication. Second, each table includes measures of two convergence criterion — the proportion of the replications that came within 0.01 of the true global

maximum and the proportion that came with 0.0001, denoted by $\widehat{p}_2$ and $\widehat{p}_4$ respectively.

We highlight only some of the features of these tables. First, the BBPSO-MC and AT-BBPSO-MC almost always do worse than their BBPSOxp-MC and AT-BBPSOxp-MC cousins, both in terms of mean absolute difference from the global maximum and in terms of the convergence criterion. The main exception is OF2. Second, for most algorithms the more restrictive neighborhood appears to be result in algorithms which do a better job of finding the global max. This is not universally true, and one interesting class of exceptions are the AT-PSO algorithms. For them, 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. Though for OF4 a more restrictive neighborhood always seems preferable. Typically, the AT-PSO algorithms with $R^* = 0.3$ or $R^* = 0.5$ using the ring-1 neighborhood does the best of the AT-PSO algorithms, though there are exceptions.

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. Call this combination with the ring-1 neighborhood the default DI-PSO algorithm. The default combination 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, lower values of *df* often result in better algorithms, though sometimes the difference is small. The impact of R^* is much smaller, though $R^* = 0.3$ or $R^* = 0.5$ seem to be the safest choices, though other classes of algorithms will often perform better.

In general, the standard PSO algorithm and the BBPSOxp-MC algorithm do pretty well and should serve as baselines, typically using the ring-1 neighborhood. For OF1, PSO meets both convergence criterion 100% of the time in our simulations, while only BBPSOxp-MC has $\widehat{p}_2 = 1$. None of the AT-BBPSOxp-MC algorithms are able to hit the second criterion

at a high rate, though our recommended DI-PSO and AT-PSO algorithms are able to. For OF2, the standard PSO algorithm using the ring-3 neighborhood is the best performing of the baseline algorithms, with $\widehat{p}_2 = 0.86$ and $\widehat{p}_4 = 0.1$. The best performing algorithm overall is AT-PSO with $R^* = 0.3$, yielding $\widehat{p}_2 = \widehat{p}_4 = 1$, and AT-PSO with $R^* = 0.5$ is in a close second with $\widehat{p}_2 = 1$ and $\widehat{p}_4 = 0.72$. None of the AT-BBPSO or DI-PSO algorithms even come close to the baseline. OF3 is much more challenging and almost no algorithm has nonzero convergence rates. The best baseline algorithm is BBPSOxp-MC with the ring-1 neighborhood, with a mean absolute difference from the true max of 18.77 respectively. The ring-1 PSO algorithm has a mean of 25.95, which we will use as another baseline for this objective function. Many of the ring-1 AT-BBPSOxp-MC algorithms are competitive with the baselines, but essentially nothing else is in terms of mean absolute difference. However, the ring-1 AT-PSO algorithm with $R^* = 0.5$ does meet both convergence criterion in 2% of replication, while no other algorithm ever meets the criterion.

The best baseline algorithm for OF4 may be ring-1 PSO, with a mean absolute difference of 0.13 and convergence proportions of 0.90 and 0.86. However, the BBPSOxp-MC algorithm is comparable with a mean of 0.01 and $\widehat{p}_2 = 0.86$, but $\widehat{p}_4 = 0$. Which is better depends on how precise you need to estimate the global maximum. All of the AT-BBPSOxp-MC algorithms are similar to BBPSOxp-MC with a near zero mean, $\widehat{p}_2$ approaching 1 and $\widehat{p}_4$ essentially 0. None of the DI-PSO algorithms are competitive and even the best AT-PSO algorithms are worse than the baseline PSO algorithm. OF5 is another difficult one where most algorithms never meet any of the convergence criterion. ring-1 PSO and ring-3 PSO are the best baselines with mean absolute differences of 0.06 and 0.07 respectively. With the ring-1 or ring-3 neighborhood, AT-BBPSOxp-MC algorithms with $df = 1$ are comparable to both baselines, The best performing algorithms in terms of mean absolute difference are the ring-1 AT-PSO algorithms with $R^* = 0.3$ or 0.5. OF6 is the most difficult with many local optima. The baseline algorithms do poorly, though ring-3 PSO and ring-1 PSO have high

convergence proportions despite high mean absolute differences. The AT-BBPSOxp-MC algorithms with $df = 1$ and $R^* = 0.3$ or 0.5 do much better in terms of the mean absolute difference, but almost never meet the convergence criterion. The ring-3 DI-PSO algorithm with $\alpha = 200$ and $\beta = 1$ is the best converging algorithm, with $\widehat{p}_2 = 0.7$ and $\widehat{p}_4 = 0.68$, though ring-3 AT-PSO with $R^* = 0.1$ has $\widehat{p}_2 = 0.7$ and $\widehat{p}_4 = 0.12$

In general ring-1 or ring-3 PSO is a great baseline algorithm that works reasonably well over a wide variety of circumstances. The AT-BBPSOxp-MC algorithms with $df = 1$ and $R^* = 0.3$ or 0.5 also tend to do reasonably well, though their strength seems to be in finding the region around the global max quickly. When the optimization problem is not too difficult, the best AT-BBPSOxp-MC algorithms have trouble converging as fast as the best alternatives, but when the optimization problem is difficult enough that convergence is unlikely in the number of allotted iterations AT-BBPSOxp-MC often gets closer to the global maximum than the alternatives on average.

The DI-PSO and AT-PSO algorithms 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 $\alpha = 200$, i.e. 20% of the total number of PSO iterations, 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 OF8. All three algorithms have an initial inertia of $\omega(0) = 1$. While DI-PSO smoothly decreases its inertia with a slowly decreasing rate, for OF1 AT-PSO very quickly drops its inertia to about 0.5 then bounces around around 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 search (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 $R^* = 0.5$ very rarely converged to the global max. In optimization problems with multiple local optima, both AT-PSO and AT-BBPSO variants can exhibit this behavior prematurely converge to a local optima, so they may not be advantageous for those problems. Reducing the target improvement rate (R^*) can ameliorate this problem though. We used $c = 0.1$ for all AT type algorithms in this section, but decreasing c will slow down how fast the inertia parameter is adapted in AT-PSO algorithms and allow for a longer high inertia exploration phase, mimicing the DI-PSO algorithms to some extent. This may also help in problems with many local optima.

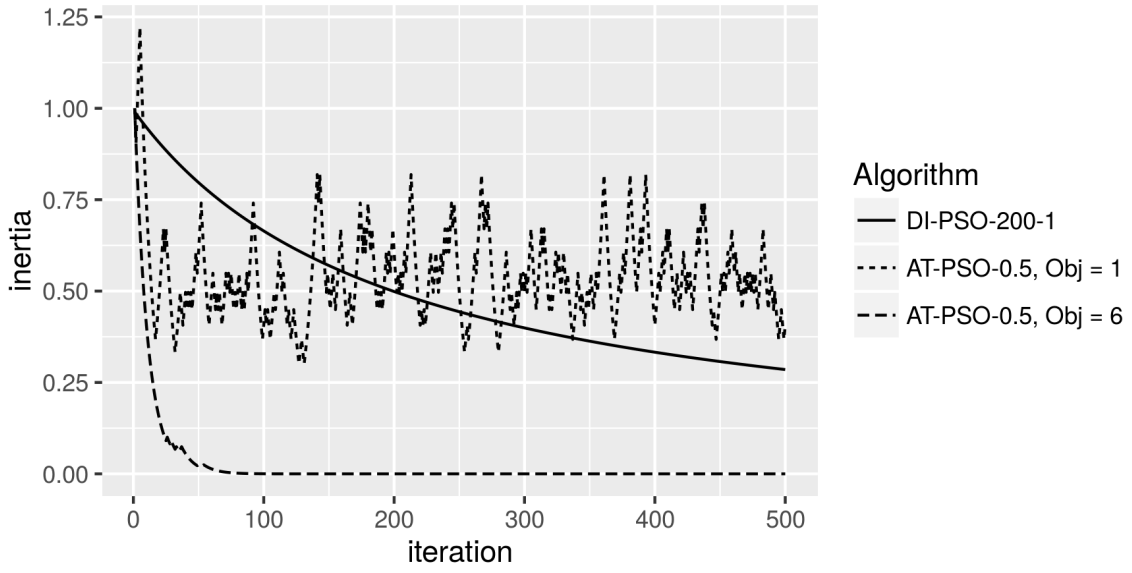

Figure 1: Inertia over time for the DI-PSO algorithm with $\alpha = 200$ and $\beta = 1$, and for one replication of the AT-PSO-0.5 algorithm for each of OFs 1 and 6.

Based on these simulations, our default recommendation is conditional. In problems where convergence is important, such as when finding the posterior mode to use for the normal approximation in a Metropolis proposal, we recommend AT-PSO with $R^* = 0.3$ or 0.5 and using a fairly restricted neighborhood (e.g., ring-1 or ring-3). These algorithms are often the best at meeting convergence criteria, especially in problems with few or no extra local optima. In difficult problems where convergence is less important (e.g., in machine learning) an AT-BBPSO variant with $df = 1$ and $R^* = 0.3$ or 0.5, and again with a restrictive neighborhood seems appropriate. These sorts of algorithms seem to do a better job of getting into a region around the global max, though not as well at searching through that region. This bipartite strategy suggests a combined strategy: use AT-BBPSO or some algorithm at first in order to quickly find a good region of the search space, then use AT-PSO to quickly search through that region.

C Deriving the Hessian

The log posterior in the fully parameterized Poisson case can be written as

$$\begin{aligned}
\log p(\boldsymbol{\beta}, \boldsymbol{L}, \boldsymbol{\delta} | \boldsymbol{z}, \boldsymbol{X}, \boldsymbol{S}) &= \text{constant} + \sum_{i=1}^{n} (y_i z_i - \exp(y_i)) + \sum_{k=1}^{r} \frac{d-k+1}{2} \log \ell_{kk}^2 \\
&\quad - \frac{1}{2} \left\{ \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}' \boldsymbol{\delta} + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v^2} + \text{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}') \right\} \\
&= \text{constant} + \sum_{i=1}^{n} \{ z_i y_i - \exp(y_i) \} + \frac{1}{2} \boldsymbol{R}_r \, \text{vech}(\log \boldsymbol{L}^2) \\
&\quad - \frac{1}{2} \left[\text{vech}(\boldsymbol{L})' \boldsymbol{M}_r \left\{ \boldsymbol{K}_r' (\boldsymbol{\delta}\boldsymbol{\delta}' \otimes \boldsymbol{I}_r) \boldsymbol{K}_r + (\boldsymbol{I}_r \otimes \boldsymbol{E}) \right\} \boldsymbol{M}_r' \, \text{vech}(\boldsymbol{L}) + \frac{(\boldsymbol{\beta} - \boldsymbol{b})'(\boldsymbol{\beta} - \boldsymbol{b})}{v^2} \right]
\end{aligned}$$

where $y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \boldsymbol{s}_i' \boldsymbol{\delta}$, $\otimes$ is the Kronecker product, $\boldsymbol{I}_r$ is the $r \times r$ identity matrix, and $\boldsymbol{R}_r$ is an $r(r+1)/2 \times r(r+1)/2$ matrix of zeroes except the $(r+1)k - k(k+1)/2 + k - r$st diagonal element is equal to $d - k + 1$ for $k = 1, 2, \ldots, r$, corresponding to locations in $\text{vech}(\boldsymbol{L})$ that store the diagonal elements of $\boldsymbol{L}$. In $\text{vech}(\log \boldsymbol{L}^2)$ the log and power are applied element-

wise. The expansions of $\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}$ and $\operatorname{tr}(\boldsymbol{E}\boldsymbol{L}\boldsymbol{L}')$ can be derived from the properties of the trace operator, Kronecker products, and the following identities.

$$\begin{aligned}\boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta} &= \operatorname{vec}(\boldsymbol{L}'\boldsymbol{\delta})'\operatorname{vec}(\boldsymbol{L}'\boldsymbol{\delta})\\ \operatorname{vec}(\boldsymbol{L}'\boldsymbol{\delta}) &= (\boldsymbol{\delta}'\otimes\boldsymbol{I}_r)\operatorname{vec}(\boldsymbol{L}')\\ \operatorname{vec}(\boldsymbol{L}') &= \boldsymbol{K}_r\operatorname{vec}(\boldsymbol{L}) = \boldsymbol{K}_r\boldsymbol{M}_r'\operatorname{vech}(\boldsymbol{L})\end{aligned}$$

and assuming $\boldsymbol{E} = \boldsymbol{C}\boldsymbol{C}'$, then $\operatorname{tr}(\boldsymbol{L}'\boldsymbol{C}\boldsymbol{C}'\boldsymbol{L}) = \operatorname{vec}(\boldsymbol{C}'\boldsymbol{L})'\operatorname{vec}(\boldsymbol{C}'\boldsymbol{L})$ where $\operatorname{vec}(\cdot)$ is the vectorization operation which stacks each column of its argument on top of each other into a single column vector, $\operatorname{vech}(\cdot)$ is the half-vectorization operator which is similar but omits elements above the diagonal, tr is the trace operator, $\boldsymbol{K}_r$ is the $r^2 \times r^2$ commutation matrix such that for any $r \times r$ matrix $\boldsymbol{L}$, $\operatorname{vec}(\boldsymbol{L}') = \boldsymbol{K}_r\operatorname{vec}(\boldsymbol{L})$, $\boldsymbol{M}_r$ is the $r^2 \times r(r+1)/2$ elimination matrix such that for $\operatorname{vech}(\boldsymbol{L}) = \boldsymbol{M}_r\operatorname{vec}(\boldsymbol{L})$ and if additionally $\boldsymbol{L}$ is lower triangular, $\operatorname{vec}(\boldsymbol{L}) = \boldsymbol{M}_r'\operatorname{vech}(\boldsymbol{L})$ (Magnus and Neudecker, 1980; Magnus, 1988). Let $\boldsymbol{E}_{ij}$ be an $r \times r$ matrix of zeroes with a one in only its (i,j)th position, $\boldsymbol{u}_{ij} = \operatorname{vech}(\boldsymbol{E}_{ij})$, and $\boldsymbol{e}_i$ be a r-dimensional column vector of zeroes with a one in the ith row. Then $\boldsymbol{K}_r$ and $\boldsymbol{M}_r$ can be written as

$$\boldsymbol{K}_r = \sum_{i=1}^{r}\sum_{j=1}^{r}\boldsymbol{E}_{ij}\otimes\boldsymbol{E}_{ij}' \qquad \text{and} \qquad \boldsymbol{M}_r = \sum_{i\geq j}^{r}\boldsymbol{u}_{ij}\otimes\boldsymbol{e}_j'\otimes\boldsymbol{e}_i'.$$

Now the first derivatives of the log posterior are given by

$$\begin{aligned}\frac{\partial\log p}{\partial\boldsymbol{\beta}} &= \sum_{i=1}^{n}(z_i - e^{y_i})\boldsymbol{x}_i' - \frac{(\boldsymbol{\beta}-\boldsymbol{b})'}{v^2},\\ \frac{\partial\log p}{\partial\boldsymbol{\delta}} &= \sum_{i=1}^{n}(z_i - e^{y_i})\boldsymbol{s}_i' - \boldsymbol{\delta}'\boldsymbol{L}\boldsymbol{L}',\\ \frac{\partial\log p}{\partial\operatorname{vech}(\boldsymbol{L})} &= -\operatorname{vech}(\boldsymbol{L})'\boldsymbol{M}_r\left[\boldsymbol{K}_r'(\boldsymbol{\delta}\boldsymbol{\delta}'\otimes\boldsymbol{I}_r)\boldsymbol{K}_r + (\boldsymbol{I}_r\otimes\boldsymbol{E})\right]\boldsymbol{M}_r' + \boldsymbol{R}_r\operatorname{vech}(1/\boldsymbol{L}),\end{aligned}$$

where in $\text{vech}(1/\boldsymbol{L})$ the division is applied element-wise. Next the second derivatives are

$$\begin{aligned}
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{x}_i \boldsymbol{x}_i' - \frac{1}{v^2} \boldsymbol{I}_p \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{s}_i \boldsymbol{s}_i' - \boldsymbol{L}\boldsymbol{L}' \\
\frac{\partial^2 \log p}{\partial \,\text{vech}(\boldsymbol{L}) \partial \,\text{vech}(\boldsymbol{L})'} &= -\boldsymbol{M}_r \left[\boldsymbol{K}_r'(\boldsymbol{\delta}\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)\boldsymbol{K}_r + (\boldsymbol{I}_r \otimes \boldsymbol{E})\right] \boldsymbol{M}_r' - \boldsymbol{R}_r \,\text{diag}\left(\text{vech}(\boldsymbol{L})^{-2}\right) \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\delta}'} &= -\sum_{i=1}^{n} e^{y_i} \boldsymbol{s}_i \boldsymbol{x}_i' \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \,\text{vech}(\boldsymbol{L})'} &= \boldsymbol{0}_{p \times r(r+1)/2} \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \,\text{vech}(\boldsymbol{L})'} &= -\frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \,\text{vech}(\boldsymbol{L})'} = -(\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)(\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{L} \otimes \boldsymbol{I}_r)\boldsymbol{M}_r',
\end{aligned}$$

where $\text{diag}(\text{vech}(\boldsymbol{L})^{-2})$ is a diagonal matrix with the elements of $\text{vech}(\boldsymbol{L})$ raised to the power -2 along the diagonal. The last second derivative matrix can be derived using repeated application of the chain rule and from the following facts for any $r \times r$ matrix $\boldsymbol{A}$ (Magnus and Neudecker, 2005):

$$\begin{aligned}
\frac{\partial \,\text{vec}(\boldsymbol{A}\boldsymbol{A}')}{\partial \,\text{vec}(\boldsymbol{A})} &= (\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{A} \otimes \boldsymbol{I}_r) \\
\frac{\partial \boldsymbol{A}\boldsymbol{\delta}}{\partial \,\text{vec}(\boldsymbol{A})} &= \boldsymbol{\delta}' \otimes \boldsymbol{I}_r.
\end{aligned}$$

Then, using the chain rule for lower triangular $\boldsymbol{L}$ we have

$$\begin{aligned}
\frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \,\text{vech}(\boldsymbol{L})'} &= \frac{\partial \boldsymbol{\delta}' \boldsymbol{L}\boldsymbol{L}'}{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}} \frac{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}}{\partial \,\text{vec}(\boldsymbol{L}\boldsymbol{L}')} \frac{\partial \,\text{vec}(\boldsymbol{L}\boldsymbol{L}')}{\partial \,\text{vec}(\boldsymbol{L})} \frac{\partial \,\text{vec}(\boldsymbol{L})}{\partial \,\text{vech}(\boldsymbol{L})} \frac{\partial \,\text{vech}(\boldsymbol{L})}{\partial \,\text{vech}(\boldsymbol{L})'} \\
&= \boldsymbol{I}_r \frac{\partial \boldsymbol{L}\boldsymbol{L}'\boldsymbol{\delta}}{\partial \,\text{vec}(\boldsymbol{L}\boldsymbol{L}')} \frac{\partial \,\text{vec}(\boldsymbol{L}\boldsymbol{L}')}{\partial \,\text{vec}(\boldsymbol{L})} \boldsymbol{M}_r' \boldsymbol{I}_{r(r+1)/2} \\
&= (\boldsymbol{\delta}' \otimes \boldsymbol{I}_r)(\boldsymbol{I}_{r^2} + \boldsymbol{K}_r)(\boldsymbol{L} \otimes \boldsymbol{I}_r)\boldsymbol{M}_r'.
\end{aligned}$$

Finally, the Hessian is

$$\boldsymbol{H} = \begin{bmatrix}
\frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\beta} \partial \,\text{vech}(\boldsymbol{L})'} \\
\frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \boldsymbol{\delta} \partial \,\text{vech}(\boldsymbol{L})'} \\
\frac{\partial^2 \log p}{\partial \,\text{vech}(\boldsymbol{L}) \partial \boldsymbol{\beta}'} & \frac{\partial^2 \log p}{\partial \,\text{vech}(\boldsymbol{L}) \partial \boldsymbol{\delta}'} & \frac{\partial^2 \log p}{\partial \,\text{vech}(\boldsymbol{L}) \partial \,\text{vech}(\boldsymbol{L})'}
\end{bmatrix}.$$

D Alternative MCMC algorithms

[CURRENTLY THIS SECTION ONLY HAS GENERIC VERSIONS OF THE ALGORTIHMS. DETAILS FOR OUR CASES?] This section describes the alternative MCMC algorithms used for comparisons with PSO assisted Metropolis-Hastings algorithms. In particular, we use two types of adaptive random walk Metropolis within Gibbs algorithms. In the generic problem, suppose we wish to sample from the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$ using MCMC methods. Suppose that $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and that the full conditional $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$ can be sampled from easily while the full conditional for $\boldsymbol{\theta}_2$ may be intractable. Then a single move random walk Metropolis within Gibbs (RWwG) algorithm for this problem is as follows.

Algorithm 1 (Single move random walk Metropolis within Gibbs) *Given target posterior $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y})$ where $\boldsymbol{\theta}_2 = (\theta_{21}, \theta_{22}, \ldots, \theta_{2n})$, it is easy to sample from $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$, and given that the support of $p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y})$ is $\Re^n$, iteration $t+1$ is obtained from iteration t via*

1. *Draw $\boldsymbol{\theta}_1^{(t+1)} \sim p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(t)}, \boldsymbol{y})$*

2. *For $k = 1, 2 \ldots, n$*

   Draw $\theta_{2k}^{(prop)} \sim N(\theta_{2k}^{(t)}, \eta_k)$ and form the Metropolis acceptance ratio

$$a_k^{(t)} = \frac{p(\theta_{2k}^{(prop)}|\boldsymbol{\theta}_1^{(t+1)}, \theta_{21}^{(t+1)}, , \ldots, \theta_{2k-1}^{(t+1)}, \theta_{2k+1}^{(t)}, \ldots, \theta_{2n}^{(t)}, \boldsymbol{y})}{p(\theta_{2k}^{(t)}|\boldsymbol{\theta}_1^{(t+1)}, \theta_{21}^{(t+1)}, , \ldots, \theta_{2k-1}^{(t+1)}, \theta_{2k+1}^{(t)}, \ldots, \theta_{2n}^{(t)}, \boldsymbol{y})}.$$

   Then set $\theta_{2k}^{(t+1)} = \theta_{2k}^{(prop)}$ with probability $r_k^{(t)} = \min(a_k, 1)$ and otherwise set $\theta_{2k}^{(t+1)} = \theta_{2k}^{(t)}$

A major problem with this algorithm is that the η_ks must be selected so that the algorithm Metropolis steps accept a reasonable amount of time. According to Gelman et al. (1996) the optimal acceptance rate in a narrow set of problems is 0.44 for a single dimensional random walk, though this is often used as a guideline for more complex problems. We

adaptively tune η_k during the burn-in period of the chain in a manner discussed in Andrieu and Thoms (2008). The computed Metropolis acceptance probability for θ_{2k} is denoted by $r_k^{(t)} = \min(a_k^{(t)}, 1)$; let r^* denote the target acceptance probability. Then we evolve η_k over time via

$$\log\ \eta_k^{(t+1)} = \log\ \eta_k^{(t)} + \gamma^{(t+1)}(r_k^{(t)} - r^*)$$

where $\gamma^{(t)}$ is a fixed sequence decreasing to zero fast enough to enough that the algorithm converges. The intuition here is that if the acceptance rate is too high, we can increase the effective search space by increasing the random walk standard deviation, while if it is too low, we can decrease the effective search space by decreasing the random walk standard deviation. We use $r^* = 0.44$. A tuning method such as this implies that $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots)$ is no longer a Markov chain, and additional conditions are required to ensure convergence to the target posterior distribution. So in practice our RWwG algorithms run Algorithm 1 with the tuning method described above until convergence and until the η_ks settle into a rough equilibrium, then we use Algorithm 1 without tuning but using the last known values of the η_ks. In practice this is equivalent setting $\gamma^{(t)} = 0$ after the burn-in period. During the burn-in we set $\gamma^{(t)} = 1$ and we initialize at $\eta_k^{(0)} = 1$ for all k.

An alternative to RWwG algorithms are block random walk Metropolis within Gibbs algorithms (B-RWwG). Suppose we have an estimate of the covariance structure between the elements of $\boldsymbol{\theta}_2$ in the posterior. Then

Algorithm 2 (Block random walk Metropolis within Gibbs) *Given target posterior* $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\boldsymbol{y})$ *where it is easy to sample from* $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \boldsymbol{y})$*, the support of* $p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{y})$ *is* $\Re^n$*, and given an estimate* $\boldsymbol{\Sigma}$ *of* $\mathrm{cov}(\boldsymbol{\theta}_2|\boldsymbol{y})$*, iteration* $t+1$ *is obtained from iteration* t *via*

1. *Draw* $\boldsymbol{\theta}_1^{(t+1)} \sim p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(t)}, \boldsymbol{y})$

2. *Draw* $\boldsymbol{\theta}_2^{(prop)} \sim N(\boldsymbol{\theta}_2^{(t)}, \eta\boldsymbol{\Sigma})$ *and form the Metropolis acceptance ratio*

$$a = \frac{p(\boldsymbol{\theta}_2^{(prop)}|\boldsymbol{\theta}_1^{(t+1)}, \boldsymbol{y})}{p(\boldsymbol{\theta}_2^{(t)}|\boldsymbol{\theta}_1^{(t+1)}, \boldsymbol{y})}.$$

Then set $\boldsymbol{\theta}_2^{(t+1)} = \boldsymbol{\theta}_2^{(prop)}$ *with probability* $\min(a, 1)$ *and otherwise set* $\boldsymbol{\theta}_2^{(t+1)} = \boldsymbol{\theta}_2^{(t)}$.

We make this algorithm adaptive as well by tuning η in the same fashion as the η_ks above. The main difference is that this algorithm must be initialized with a crude estimate of $\boldsymbol{\Sigma}$ by running, e.g., Algorithm 1.

Obj = 1	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$
PSO	0.00	0.01	0.98	0.92	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00
BBPSO-MC	0.12	0.04	0.00	0.00	0.09	0.03	0.00	0.00	0.04	0.02	0.06	0.00
BBPSOxp-MC	0.02	0.01	0.06	0.00	0.01	0.01	0.60	0.00	0.00	0.00	1.00	0.00
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	0.02	0.01	0.02	0.00	0.01	0.01	0.22	0.00	0.00	0.00	0.96	0.00
$df = 1,\ R^* = 0.3$	0.02	0.01	0.06	0.00	0.01	0.00	0.22	0.00	0.00	0.00	0.96	0.00
$df = 1,\ R^* = 0.5$	0.02	0.01	0.04	0.00	0.01	0.00	0.10	0.00	0.00	0.00	0.96	0.00
$df = 1,\ R^* = 0.7$	0.02	0.01	0.06	0.00	0.01	0.01	0.22	0.00	0.00	0.00	0.98	0.00
$df = 3,\ R^* = 0.1$	0.04	0.01	0.02	0.00	0.03	0.01	0.00	0.00	0.01	0.00	0.70	0.00
$df = 3,\ R^* = 0.3$	0.04	0.02	0.00	0.00	0.03	0.01	0.00	0.00	0.01	0.00	0.56	0.00
$df = 3,\ R^* = 0.5$	0.04	0.01	0.00	0.00	0.03	0.01	0.02	0.00	0.01	0.00	0.60	0.00
$df = 3,\ R^* = 0.7$	0.04	0.01	0.00	0.00	0.03	0.01	0.02	0.00	0.01	0.00	0.64	0.00
$df = 5,\ R^* = 0.1$	0.06	0.02	0.00	0.00	0.04	0.02	0.00	0.00	0.01	0.01	0.32	0.00
$df = 5,\ R^* = 0.3$	0.06	0.02	0.00	0.00	0.04	0.01	0.00	0.00	0.02	0.01	0.26	0.00
$df = 5,\ R^* = 0.5$	0.06	0.02	0.00	0.00	0.04	0.01	0.00	0.00	0.01	0.01	0.34	0.00
$df = 5,\ R^* = 0.7$	0.06	0.02	0.00	0.00	0.04	0.01	0.00	0.00	0.01	0.01	0.36	0.00
$df = \infty,\ R^* = 0.1$	0.13	0.04	0.00	0.00	0.09	0.04	0.00	0.00	0.04	0.02	0.00	0.00
$df = \infty,\ R^* = 0.3$	0.12	0.04	0.00	0.00	0.09	0.02	0.00	0.00	0.04	0.02	0.02	0.00
$df = \infty,\ R^* = 0.5$	0.13	0.04	0.00	0.00	0.10	0.03	0.00	0.00	0.04	0.02	0.02	0.00
$df = \infty,\ R^* = 0.7$	0.12	0.04	0.00	0.00	0.09	0.03	0.00	0.00	0.03	0.02	0.00	0.00
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	0.00	0.00	0.98	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.20
$df = 1,\ R^* = 0.3$	0.00	0.00	0.98	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.32
$df = 1,\ R^* = 0.5$	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.26
$df = 1,\ R^* = 0.7$	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.20
$df = 3,\ R^* = 0.1$	0.01	0.00	0.90	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.10
$df = 3,\ R^* = 0.3$	0.01	0.00	0.94	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.10
$df = 3,\ R^* = 0.5$	0.01	0.00	0.88	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.06
$df = 3,\ R^* = 0.7$	0.01	0.00	0.90	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08
$df = 5,\ R^* = 0.1$	0.01	0.00	0.64	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.08
$df = 5,\ R^* = 0.3$	0.01	0.00	0.50	0.00	0.00	0.00	0.96	0.00	0.00	0.00	1.00	0.02
$df = 5,\ R^* = 0.5$	0.01	0.00	0.58	0.00	0.00	0.00	0.98	0.00	0.00	0.00	1.00	0.02
$df = 5,\ R^* = 0.7$	0.01	0.00	0.54	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.04
$df = \infty,\ R^* = 0.1$	0.02	0.01	0.00	0.00	0.01	0.00	0.58	0.00	0.00	0.00	1.00	0.00
$df = \infty,\ R^* = 0.3$	0.02	0.01	0.06	0.00	0.01	0.00	0.52	0.00	0.00	0.00	1.00	0.02
$df = \infty,\ R^* = 0.5$	0.02	0.01	0.10	0.00	0.01	0.01	0.60	0.00	0.00	0.00	1.00	0.00
$df = \infty,\ R^* = 0.7$	0.02	0.01	0.10	0.00	0.01	0.00	0.58	0.00	0.00	0.00	1.00	0.00
DI-PSO												

Obj = 2	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$
PSO	15.69	80.28	0.66	0.42	0.01	0.01	0.86	0.10	1.92	4.16	0.00	0.00
BBPSO-MC	0.43	0.13	0.00	0.00	0.47	0.23	0.00	0.00	0.72	0.53	0.00	0.00
BBPSOxp-MC	2.10	1.59	0.00	0.00	3.64	4.33	0.00	0.00	3.20	7.55	0.00	0.00
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	0.17	0.10	0.00	0.00	0.18	0.09	0.00	0.00	0.10	0.08	0.00	0.00
$df = 1,\ R^* = 0.3$	0.19	0.08	0.00	0.00	0.17	0.09	0.00	0.00	0.10	0.06	0.00	0.00
$df = 1,\ R^* = 0.5$	0.17	0.09	0.00	0.00	0.19	0.11	0.00	0.00	0.16	0.14	0.00	0.00
$df = 1,\ R^* = 0.7$	0.18	0.09	0.00	0.00	0.16	0.09	0.00	0.00	0.11	0.09	0.00	0.00
$df = 3,\ R^* = 0.1$	0.20	0.08	0.00	0.00	0.18	0.10	0.00	0.00	0.17	0.17	0.00	0.00
$df = 3,\ R^* = 0.3$	0.20	0.08	0.00	0.00	0.18	0.08	0.00	0.00	0.12	0.08	0.00	0.00
$df = 3,\ R^* = 0.5$	0.20	0.09	0.00	0.00	0.19	0.09	0.00	0.00	0.12	0.07	0.00	0.00
$df = 3,\ R^* = 0.7$	0.21	0.08	0.00	0.00	0.22	0.08	0.00	0.00	0.12	0.10	0.00	0.00
$df = 5,\ R^* = 0.1$	0.26	0.10	0.00	0.00	0.22	0.10	0.00	0.00	0.16	0.09	0.00	0.00
$df = 5,\ R^* = 0.3$	0.27	0.13	0.00	0.00	0.24	0.09	0.00	0.00	0.17	0.11	0.00	0.00
$df = 5,\ R^* = 0.5$	0.25	0.08	0.00	0.00	0.24	0.10	0.00	0.00	0.16	0.10	0.00	0.00
$df = 5,\ R^* = 0.7$	0.27	0.09	0.00	0.00	0.20	0.09	0.00	0.00	0.19	0.10	0.00	0.00
$df = \infty,\ R^* = 0.1$	0.45	0.18	0.00	0.00	0.47	0.17	0.00	0.00	0.58	0.65	0.00	0.00
$df = \infty,\ R^* = 0.3$	0.52	0.22	0.00	0.00	0.46	0.20	0.00	0.00	0.57	0.38	0.00	0.00
$df = \infty,\ R^* = 0.5$	0.42	0.17	0.00	0.00	0.42	0.15	0.00	0.00	0.66	0.63	0.00	0.00
$df = \infty,\ R^* = 0.7$	0.45	0.20	0.00	0.00	0.46	0.18	0.00	0.00	0.60	0.46	0.00	0.00
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	1.32	0.93	0.00	0.00	0.97	0.95	0.00	0.00	0.29	0.36	0.00	0.00
$df = 1,\ R^* = 0.3$	1.33	1.25	0.00	0.00	0.98	0.86	0.00	0.00	0.23	0.26	0.00	0.00
$df = 1,\ R^* = 0.5$	1.55	1.65	0.00	0.00	1.09	1.76	0.00	0.00	0.27	0.44	0.00	0.00
$df = 1,\ R^* = 0.7$	1.07	0.83	0.00	0.00	1.10	1.05	0.00	0.00	0.19	0.11	0.00	0.00
$df = 3,\ R^* = 0.1$	0.72	0.56	0.00	0.00	0.69	0.50	0.00	0.00	0.29	0.24	0.00	0.00
$df = 3,\ R^* = 0.3$	0.70	0.45	0.00	0.00	0.95	1.25	0.00	0.00	0.25	0.31	0.00	0.00
$df = 3,\ R^* = 0.5$	0.63	0.35	0.00	0.00	0.77	0.66	0.00	0.00	0.36	0.48	0.00	0.00
$df = 3,\ R^* = 0.7$	0.64	0.49	0.00	0.00	0.73	0.47	0.00	0.00	0.32	0.42	0.00	0.00
$df = 5,\ R^* = 0.1$	0.69	0.42	0.00	0.00	1.83	5.74	0.00	0.00	0.77	1.64	0.00	0.00
$df = 5,\ R^* = 0.3$	0.80	0.49	0.00	0.00	0.93	0.77	0.00	0.00	0.50	0.45	0.00	0.00
$df = 5,\ R^* = 0.5$	0.99	0.79	0.00	0.00	0.83	0.69	0.00	0.00	0.73	1.06	0.00	0.00
$df = 5,\ R^* = 0.7$	0.76	0.47	0.00	0.00	0.80	0.72	0.00	0.00	0.38	0.40	0.00	0.00
$df = \infty,\ R^* = 0.1$	2.05	1.82	0.00	0.00	4.80	6.08	0.00	0.00	4.89	8.52	0.00	0.00
$df = \infty,\ R^* = 0.3$	1.53	0.80	0.00	0.00	3.90	4.95	0.00	0.00	5.17	17.76	0.00	0.00
$df = \infty,\ R^* = 0.5$	2.48	3.39	0.00	0.00	5.39	5.48	0.00	0.00	6.77	14.34	0.00	0.00
$df = \infty,\ R^* = 0.7$	2.00	1.95	0.00	0.00	4.98	7.45	0.00	0.00	5.33	10.20	0.00	0.00
DI-PSO												

Obj = 3	Global nbhd				Ring-3 nbhd				Ring-1 nbhd		
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$
PSO	144.46	295.40	0.00	0.00	35.66	54.01	0.04	0.00	25.95	68.25	0.00
BBPSO-MC	131.31	90.51	0.00	0.00	110.19	63.02	0.00	0.00	84.45	71.65	0.00
BBPSOxp-MC	37.56	14.15	0.00	0.00	25.53	8.62	0.00	0.00	18.77	6.24	0.00
AT-BBPSO-MC											
$df = 1,\ R^* = 0.1$	111.48	250.62	0.00	0.00	53.72	43.43	0.00	0.00	110.78	165.53	0.00
$df = 1,\ R^* = 0.3$	149.32	242.67	0.00	0.00	90.41	154.44	0.00	0.00	60.90	60.45	0.00
$df = 1,\ R^* = 0.5$	106.08	127.98	0.00	0.00	78.98	111.62	0.00	0.00	49.87	53.05	0.00
$df = 1,\ R^* = 0.7$	120.35	187.50	0.00	0.00	79.60	107.68	0.00	0.00	83.29	126.26	0.00
$df = 3,\ R^* = 0.1$	114.66	118.03	0.00	0.00	88.22	58.20	0.00	0.00	69.83	60.08	0.00
$df = 3,\ R^* = 0.3$	144.03	225.00	0.00	0.00	78.97	65.90	0.00	0.00	73.54	100.15	0.00
$df = 3,\ R^* = 0.5$	121.44	127.02	0.00	0.00	79.52	71.45	0.00	0.00	57.98	48.82	0.00
$df = 3,\ R^* = 0.7$	108.30	81.33	0.00	0.00	71.77	64.65	0.00	0.00	54.11	46.59	0.00
$df = 5,\ R^* = 0.1$	125.89	116.43	0.00	0.00	78.12	32.51	0.00	0.00	71.77	44.25	0.00
$df = 5,\ R^* = 0.3$	108.21	69.88	0.00	0.00	94.78	44.39	0.00	0.00	67.79	65.60	0.00
$df = 5,\ R^* = 0.5$	111.18	90.71	0.00	0.00	78.14	35.93	0.00	0.00	68.41	67.38	0.00
$df = 5,\ R^* = 0.7$	107.09	63.45	0.00	0.00	93.72	69.07	0.00	0.00	71.64	92.09	0.00
$df = \infty,\ R^* = 0.1$	126.16	71.55	0.00	0.00	99.98	25.30	0.00	0.00	68.42	43.53	0.00
$df = \infty,\ R^* = 0.3$	127.38	66.15	0.00	0.00	103.96	31.69	0.00	0.00	68.21	54.76	0.00
$df = \infty,\ R^* = 0.5$	109.58	30.42	0.00	0.00	102.75	39.42	0.00	0.00	76.03	59.73	0.00
$df = \infty,\ R^* = 0.7$	112.31	31.53	0.00	0.00	111.33	47.31	0.00	0.00	72.48	29.60	0.00
AT-BBPSOxp-MC											
$df = 1,\ R^* = 0.1$	54.91	71.36	0.00	0.00	46.43	53.25	0.00	0.00	27.74	33.78	0.00
$df = 1,\ R^* = 0.3$	42.75	27.54	0.00	0.00	34.70	44.52	0.00	0.00	26.56	35.25	0.00
$df = 1,\ R^* = 0.5$	47.05	33.70	0.00	0.00	52.38	84.94	0.00	0.00	35.21	47.68	0.00
$df = 1,\ R^* = 0.7$	44.34	44.22	0.00	0.00	55.48	108.75	0.00	0.00	39.44	46.50	0.00
$df = 3,\ R^* = 0.1$	34.88	12.52	0.00	0.00	35.55	22.20	0.00	0.00	19.04	13.27	0.00
$df = 3,\ R^* = 0.3$	40.08	14.57	0.00	0.00	33.62	25.98	0.00	0.00	23.77	19.44	0.00
$df = 3,\ R^* = 0.5$	44.76	48.60	0.00	0.00	27.92	12.79	0.00	0.00	27.27	35.43	0.00
$df = 3,\ R^* = 0.7$	46.32	77.32	0.00	0.00	30.46	14.12	0.00	0.00	18.80	8.33	0.00
$df = 5,\ R^* = 0.1$	40.72	15.41	0.00	0.00	35.09	28.26	0.00	0.00	20.00	19.18	0.00
$df = 5,\ R^* = 0.3$	40.81	22.60	0.00	0.00	32.33	19.05	0.00	0.00	20.36	12.27	0.00
$df = 5,\ R^* = 0.5$	43.59	51.50	0.00	0.00	43.91	76.14	0.00	0.00	29.54	51.03	0.00
$df = 5,\ R^* = 0.7$	38.25	13.76	0.00	0.00	29.31	12.40	0.00	0.00	21.43	26.09	0.00
$df = \infty,\ R^* = 0.1$	40.62	19.73	0.00	0.00	31.86	18.35	0.00	0.00	25.61	42.18	0.00
$df = \infty,\ R^* = 0.3$	37.54	31.58	0.00	0.00	27.73	9.34	0.00	0.00	17.79	6.45	0.00
$df = \infty,\ R^* = 0.5$	39.05	14.07	0.00	0.00	27.79	8.39	0.00	0.00	20.38	6.64	0.00
$df = \infty,\ R^* = 0.7$	35.58	12.57	0.00	0.00	28.22	8.33	0.00	0.00	18.96	7.86	0.00
DI-PSO											

Obj = 4	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$
PSO	4.32	2.75	0.02	0.02	0.65	0.89	0.56	0.56	0.13	0.47	0.90	0.86
BBPSO-MC	2.75	0.93	0.00	0.00	2.67	0.81	0.00	0.00	1.47	1.00	0.00	0.00
BBPSOxp-MC	0.13	0.05	0.00	0.00	0.04	0.02	0.04	0.00	0.01	0.00	0.86	0.00
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	1.02	0.80	0.00	0.00	0.48	0.38	0.00	0.00	0.21	0.23	0.00	0.00
$df = 1,\ R^* = 0.3$	1.10	0.88	0.00	0.00	0.58	0.39	0.00	0.00	0.23	0.28	0.00	0.00
$df = 1,\ R^* = 0.5$	0.90	0.66	0.00	0.00	0.51	0.33	0.00	0.00	0.37	0.42	0.00	0.00
$df = 1,\ R^* = 0.7$	0.98	0.68	0.00	0.00	0.50	0.31	0.00	0.00	0.29	0.43	0.00	0.00
$df = 3,\ R^* = 0.1$	1.13	0.56	0.00	0.00	0.91	0.42	0.00	0.00	0.47	0.48	0.00	0.00
$df = 3,\ R^* = 0.3$	1.25	0.54	0.00	0.00	0.89	0.52	0.00	0.00	0.46	0.48	0.00	0.00
$df = 3,\ R^* = 0.5$	1.23	0.48	0.00	0.00	0.97	0.48	0.00	0.00	0.40	0.47	0.00	0.00
$df = 3,\ R^* = 0.7$	1.25	0.63	0.00	0.00	1.05	0.64	0.00	0.00	0.40	0.45	0.00	0.00
$df = 5,\ R^* = 0.1$	1.54	0.64	0.00	0.00	1.25	0.53	0.00	0.00	0.81	0.62	0.00	0.00
$df = 5,\ R^* = 0.3$	1.64	0.48	0.00	0.00	1.38	0.63	0.00	0.00	0.53	0.49	0.00	0.00
$df = 5,\ R^* = 0.5$	1.72	0.60	0.00	0.00	1.59	0.77	0.00	0.00	0.55	0.47	0.00	0.00
$df = 5,\ R^* = 0.7$	1.77	0.66	0.00	0.00	1.30	0.67	0.00	0.00	0.53	0.43	0.00	0.00
$df = \infty,\ R^* = 0.1$	2.63	0.78	0.00	0.00	2.42	0.86	0.00	0.00	1.38	0.86	0.00	0.00
$df = \infty,\ R^* = 0.3$	2.52	0.63	0.00	0.00	2.51	0.58	0.00	0.00	1.22	0.76	0.00	0.00
$df = \infty,\ R^* = 0.5$	2.50	0.84	0.00	0.00	2.40	0.77	0.00	0.00	1.49	0.90	0.00	0.00
$df = \infty,\ R^* = 0.7$	2.63	0.66	0.00	0.00	2.62	0.72	0.00	0.00	1.34	0.95	0.00	0.00
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	0.09	0.04	0.00	0.00	0.02	0.01	0.06	0.00	0.00	0.00	1.00	0.00
$df = 1,\ R^* = 0.3$	0.08	0.04	0.00	0.00	0.03	0.01	0.02	0.00	0.00	0.00	0.94	0.00
$df = 1,\ R^* = 0.5$	0.07	0.03	0.00	0.00	0.03	0.01	0.10	0.00	0.00	0.00	0.94	0.00
$df = 1,\ R^* = 0.7$	0.07	0.03	0.00	0.00	0.03	0.01	0.04	0.00	0.00	0.00	1.00	0.00
$df = 3,\ R^* = 0.1$	0.10	0.04	0.00	0.00	0.03	0.01	0.02	0.00	0.00	0.00	0.98	0.00
$df = 3,\ R^* = 0.3$	0.10	0.04	0.00	0.00	0.03	0.01	0.02	0.00	0.00	0.00	0.98	0.00
$df = 3,\ R^* = 0.5$	0.10	0.04	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	1.00	0.00
$df = 3,\ R^* = 0.7$	0.09	0.04	0.00	0.00	0.03	0.02	0.06	0.00	0.00	0.00	0.96	0.00
$df = 5,\ R^* = 0.1$	0.11	0.04	0.00	0.00	0.03	0.02	0.02	0.00	0.00	0.00	0.90	0.00
$df = 5,\ R^* = 0.3$	0.10	0.04	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.96	0.00
$df = 5,\ R^* = 0.5$	0.11	0.05	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.98	0.00
$df = 5,\ R^* = 0.7$	0.10	0.04	0.00	0.00	0.03	0.02	0.02	0.00	0.00	0.00	0.94	0.00
$df = \infty,\ R^* = 0.1$	0.12	0.05	0.00	0.00	0.04	0.02	0.00	0.00	0.00	0.00	0.98	0.02
$df = \infty,\ R^* = 0.3$	0.12	0.05	0.00	0.00	0.04	0.02	0.00	0.00	0.01	0.00	0.84	0.00
$df = \infty,\ R^* = 0.5$	0.13	0.05	0.00	0.00	0.04	0.02	0.00	0.00	0.01	0.00	0.86	0.00
$df = \infty,\ R^* = 0.7$	0.12	0.04	0.00	0.00	0.04	0.02	0.00	0.00	0.00	0.00	0.94	0.00
DI-PSO												

Obj = 5	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$
PSO	0.18	0.17	0.00	0.00	0.07	0.04	0.06	0.02	0.06	0.04	0.04	0.02
BBPSO-MC	107.55	12.45	0.00	0.00	123.72	13.02	0.00	0.00	139.39	20.19	0.00	0.00
BBPSOxp-MC	161.19	16.06	0.00	0.00	147.39	15.95	0.00	0.00	88.54	22.99	0.00	0.00
AT-BBPSO-MC												
$df = 1,\ \ R^* = 0.1$	0.24	0.20	0.00	0.00	0.11	0.06	0.00	0.00	0.10	0.06	0.00	0.00
$df = 1,\ \ R^* = 0.3$	0.22	0.15	0.00	0.00	0.14	0.08	0.00	0.00	0.10	0.08	0.02	0.00
$df = 1,\ \ R^* = 0.5$	0.23	0.14	0.00	0.00	0.12	0.09	0.00	0.00	0.10	0.05	0.00	0.00
$df = 1,\ \ R^* = 0.7$	0.24	0.16	0.00	0.00	0.13	0.09	0.00	0.00	0.09	0.05	0.00	0.00
$df = 3,\ \ R^* = 0.1$	0.75	0.61	0.00	0.00	0.29	0.18	0.00	0.00	0.50	0.34	0.00	0.00
$df = 3,\ \ R^* = 0.3$	0.78	0.59	0.00	0.00	0.26	0.15	0.00	0.00	0.61	0.37	0.00	0.00
$df = 3,\ \ R^* = 0.5$	0.59	0.41	0.00	0.00	0.27	0.17	0.00	0.00	0.51	0.31	0.00	0.00
$df = 3,\ \ R^* = 0.7$	0.64	0.45	0.00	0.00	0.28	0.15	0.00	0.00	0.57	0.40	0.00	0.00
$df = 5,\ \ R^* = 0.1$	7.13	3.33	0.00	0.00	16.46	5.52	0.00	0.00	33.67	9.42	0.00	0.00
$df = 5,\ \ R^* = 0.3$	8.03	3.84	0.00	0.00	16.27	4.90	0.00	0.00	31.74	9.31	0.00	0.00
$df = 5,\ \ R^* = 0.5$	7.18	4.05	0.00	0.00	16.78	5.88	0.00	0.00	36.30	12.42	0.00	0.00
$df = 5,\ \ R^* = 0.7$	7.62	3.58	0.00	0.00	15.76	5.55	0.00	0.00	32.13	9.37	0.00	0.00
$df = \infty,\ R^* = 0.1$	108.04	13.07	0.00	0.00	122.56	10.67	0.00	0.00	138.20	17.14	0.00	0.00
$df = \infty,\ R^* = 0.3$	109.07	12.30	0.00	0.00	123.46	10.82	0.00	0.00	137.72	19.34	0.00	0.00
$df = \infty,\ R^* = 0.5$	108.41	13.09	0.00	0.00	124.54	11.65	0.00	0.00	137.44	19.45	0.00	0.00
$df = \infty,\ R^* = 0.7$	108.26	12.55	0.00	0.00	125.60	10.86	0.00	0.00	138.92	19.62	0.00	0.00
AT-BBPSOxp-MC												
$df = 1,\ \ R^* = 0.1$	0.09	0.04	0.00	0.00	0.08	0.04	0.00	0.00	0.05	0.04	0.02	0.00
$df = 1,\ \ R^* = 0.3$	0.10	0.05	0.00	0.00	0.08	0.05	0.06	0.00	0.06	0.03	0.02	0.00
$df = 1,\ \ R^* = 0.5$	0.10	0.06	0.00	0.00	0.08	0.05	0.00	0.00	0.07	0.03	0.02	0.00
$df = 1,\ \ R^* = 0.7$	0.10	0.06	0.00	0.00	0.08	0.05	0.00	0.00	0.06	0.04	0.06	0.00
$df = 3,\ \ R^* = 0.1$	3.37	2.43	0.00	0.00	4.40	2.86	0.00	0.00	0.90	0.87	0.00	0.00
$df = 3,\ \ R^* = 0.3$	3.18	2.42	0.00	0.00	3.26	1.80	0.00	0.00	1.10	1.16	0.00	0.00
$df = 3,\ \ R^* = 0.5$	3.27	2.30	0.00	0.00	3.59	3.42	0.00	0.00	0.97	1.20	0.00	0.00
$df = 3,\ \ R^* = 0.7$	3.71	2.43	0.00	0.00	4.97	4.03	0.00	0.00	0.83	0.70	0.00	0.00
$df = 5,\ \ R^* = 0.1$	64.33	10.31	0.00	0.00	62.16	10.89	0.00	0.00	25.49	10.74	0.00	0.00
$df = 5,\ \ R^* = 0.3$	63.62	10.26	0.00	0.00	61.79	12.09	0.00	0.00	26.34	10.67	0.00	0.00
$df = 5,\ \ R^* = 0.5$	62.01	10.94	0.00	0.00	62.58	9.87	0.00	0.00	26.15	13.04	0.00	0.00
$df = 5,\ \ R^* = 0.7$	64.75	10.55	0.00	0.00	62.63	11.46	0.00	0.00	25.45	10.98	0.00	0.00
$df = \infty,\ R^* = 0.1$	157.62	14.20	0.00	0.00	148.31	13.62	0.00	0.00	87.94	24.29	0.00	0.00
$df = \infty,\ R^* = 0.3$	160.30	15.23	0.00	0.00	146.65	16.52	0.00	0.00	87.96	23.07	0.00	0.00
$df = \infty,\ R^* = 0.5$	158.78	16.14	0.00	0.00	150.94	14.95	0.00	0.00	91.44	21.82	0.00	0.00
$df = \infty,\ R^* = 0.7$	162.49	16.45	0.00	0.00	145.00	16.78	0.00	0.00	87.56	20.07	0.00	0.00
DI-PSO												

Obj = 6	Global nbhd				Ring-3 nbhd				Ring-1 nbhd			
Algorithm	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$	Mean	SD	$\widehat{p}_2$	$\widehat{p}_4$
PSO	18.95	3.04	0.02	0.02	9.06	9.92	0.54	0.50	13.43	9.17	0.24	0.08
BBPSO-MC	20.25	0.10	0.00	0.00	20.28	0.12	0.00	0.00	17.53	6.84	0.00	0.00
BBPSOxp-MC	19.87	0.07	0.00	0.00	19.75	0.10	0.00	0.00	19.44	0.30	0.00	0.00
AT-BBPSO-MC												
$df = 1,\ R^* = 0.1$	5.91	9.04	0.00	0.00	0.69	2.72	0.00	0.00	0.13	0.06	0.00	0.00
$df = 1,\ R^* = 0.3$	5.56	8.81	0.00	0.00	1.10	3.96	0.00	0.00	0.92	3.99	0.00	0.00
$df = 1,\ R^* = 0.5$	5.94	8.99	0.00	0.00	0.27	0.08	0.00	0.00	0.22	0.42	0.00	0.00
$df = 1,\ R^* = 0.7$	4.05	7.69	0.00	0.00	0.64	2.85	0.00	0.00	0.59	2.89	0.00	0.00
$df = 3,\ R^* = 0.1$	18.85	4.73	0.00	0.00	8.42	9.54	0.00	0.00	7.18	9.53	0.00	0.00
$df = 3,\ R^* = 0.3$	19.44	3.81	0.00	0.00	8.01	9.68	0.00	0.00	4.67	8.33	0.00	0.00
$df = 3,\ R^* = 0.5$	19.39	3.85	0.00	0.00	10.02	9.83	0.00	0.00	5.14	8.57	0.00	0.00
$df = 3,\ R^* = 0.7$	17.86	6.44	0.00	0.00	10.36	9.94	0.00	0.00	5.23	8.48	0.00	0.00
$df = 5,\ R^* = 0.1$	20.21	0.12	0.00	0.00	17.11	7.23	0.00	0.00	10.20	9.72	0.00	0.00
$df = 5,\ R^* = 0.3$	20.19	0.12	0.00	0.00	17.47	6.81	0.00	0.00	10.47	9.85	0.00	0.00
$df = 5,\ R^* = 0.5$	20.20	0.13	0.00	0.00	17.43	6.90	0.00	0.00	8.22	9.62	0.00	0.00
$df = 5,\ R^* = 0.7$	20.22	0.13	0.00	0.00	17.81	6.34	0.00	0.00	12.09	9.80	0.00	0.00
$df = \infty,\ R^* = 0.1$	20.23	0.13	0.00	0.00	20.25	0.11	0.00	0.00	16.99	7.15	0.00	0.00
$df = \infty,\ R^* = 0.3$	20.26	0.12	0.00	0.00	20.25	0.16	0.00	0.00	16.78	7.44	0.00	0.00
$df = \infty,\ R^* = 0.5$	20.22	0.09	0.00	0.00	19.87	2.79	0.00	0.00	17.97	6.14	0.00	0.00
$df = \infty,\ R^* = 0.7$	20.24	0.10	0.00	0.00	20.23	0.13	0.00	0.00	18.02	6.09	0.00	0.00
AT-BBPSOxp-MC												
$df = 1,\ R^* = 0.1$	2.19	5.99	0.00	0.00	0.49	2.73	0.00	0.00	1.00	4.02	0.04	0.00
$df = 1,\ R^* = 0.3$	1.41	4.69	0.00	0.00	1.28	4.75	0.00	0.00	0.07	0.16	0.04	0.00
$df = 1,\ R^* = 0.5$	1.11	3.97	0.00	0.00	0.14	0.39	0.00	0.00	0.06	0.15	0.00	0.00
$df = 1,\ R^* = 0.7$	0.63	2.81	0.00	0.00	0.53	2.66	0.00	0.00	0.54	2.87	0.04	0.00
$df = 3,\ R^* = 0.1$	18.23	5.28	0.00	0.00	14.49	8.28	0.00	0.00	11.62	9.28	0.00	0.00
$df = 3,\ R^* = 0.3$	18.68	4.03	0.00	0.00	14.48	8.29	0.00	0.00	9.35	9.18	0.00	0.00
$df = 3,\ R^* = 0.5$	18.17	5.21	0.00	0.00	15.52	7.62	0.00	0.00	8.76	9.22	0.02	0.00
$df = 3,\ R^* = 0.7$	15.52	7.99	0.00	0.00	15.31	7.87	0.00	0.00	11.95	8.96	0.00	0.00
$df = 5,\ R^* = 0.1$	19.39	2.12	0.00	0.00	19.35	2.68	0.00	0.00	15.93	7.26	0.00	0.00
$df = 5,\ R^* = 0.3$	19.82	0.10	0.00	0.00	19.58	0.28	0.00	0.00	16.12	6.99	0.00	0.00
$df = 5,\ R^* = 0.5$	19.43	2.68	0.00	0.00	19.35	2.42	0.00	0.00	17.39	5.38	0.00	0.00
$df = 5,\ R^* = 0.7$	19.81	0.12	0.00	0.00	19.29	2.74	0.00	0.00	15.17	7.86	0.00	0.00
$df = \infty,\ R^* = 0.1$	19.85	0.08	0.00	0.00	19.73	0.12	0.00	0.00	18.69	3.37	0.00	0.00
$df = \infty,\ R^* = 0.3$	19.87	0.07	0.00	0.00	19.75	0.11	0.00	0.00	18.88	2.97	0.00	0.00
$df = \infty,\ R^* = 0.5$	19.85	0.09	0.00	0.00	19.76	0.11	0.00	0.00	18.63	3.40	0.00	0.00
$df = \infty,\ R^* = 0.7$	19.86	0.08	0.00	0.00	19.77	0.10	0.00	0.00	19.47	0.30	0.00	0.00
DI-PSO												

References

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 *Swarm Intelligence: Introduction and Applications*, eds. C. Blum and D. Merkle. Springer.

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

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.

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

Hsieh, H.-I. and Lee, T.-S. (2010). "A modified algorithm of bare bones particle swarm optimization." *International Journal of Computer Science Issues*, 7, 11.

Magnus, J. R. (1988). *Linear structures*. No. 42 in Griffin's Statistical Monographs and Courses. New York: Oxford University Press.

Magnus, J. R. and Neudecker, H. (1980). "The elimination matrix: some lemmas and applications." *SIAM Journal on Algebraic Discrete Methods*, 1, 4, 422–449.

— (2005). *Matrix differential calculus with applications in statistics and econometrics*. 3rd ed. New York: John Wiley & Sons.

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