

Review Article

On Particle Swarm Optimization for MIMO Channel Estimation

Christopher Kniewel and Peter Adam Hoehner

Information and Coding Theory Laboratory, University of Kiel, 24143 Kiel, Germany

Correspondence should be addressed to Christopher Kniewel, chk@tf.uni-kiel.de

Received 7 July 2011; Revised 30 November 2011; Accepted 14 December 2011

Academic Editor: Lisimachos P. Kondi

Copyright © 2012 C. Kniewel and P. A. Hoehner. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Evolutionary algorithms, in particular particle swarm optimization (PSO), have recently received much attention. PSO has successfully been applied to a wide range of technical optimization problems, including channel estimation. However, most publications in the area of digital communications ignore improvements developed by the PSO community. In this paper, an overview of the original PSO is given as well as improvements that are generally applicable. An extension of PSO termed cooperative PSO (CPSO) is applied for MIMO channel estimation, providing faster convergence and, thus, lower overall complexity. Instead of determining the average iterations needed empirically, a method to calculate the maximum number of iterations is developed, which enables the evaluation of the complexity for a wide range of parameters. Knowledge of the required number of iterations is essential for a practical receiver design. A detailed discussion about the complexity of the PSO algorithm and a comparison to a conventional minimum mean squared error (MMSE) estimator are given. Furthermore, Monte Carlo simulations are provided to illustrate the MSE performance compared to an MMSE estimator.

1. Introduction

Multiple-input multiple-output (MIMO) transmission is considered as a key technology to reach the challenging goals of upcoming wireless standards, such as long-term evolution advanced (LTE-A) [1]. A wide range of MIMO detectors are known in the literature offering a performance close to the channel capacity. Precise channel state information (CSI) is required to obtain this performance. Correspondingly, the performance of the detection algorithms highly depends on the accuracy of the CSI. Minimum mean squared error (MMSE-) based channel estimation reaches optimum performance with computational cost that may become infeasible for practical implementation.

Advanced iterative receivers, which jointly detect the data symbols and estimate the channel, offer a close-to-optimum performance at often reduced computational complexity. However, the majority of joint receivers need good initial channel estimates to reach their ultimate performance. The space alternating generalized expectation (SAGE) algorithm in [2] and the graph-based iterative receiver in [3] are examples of iterative receivers which need proper initialization.

Generally, channel estimation can be seen as an optimization problem, that is, to minimize the Euclidean distance between the estimated and the true channel coefficients. The straightforward solution to this problem incorporates matrix inversion and leads to the well-known least-squares (LS) and/or MMSE estimator.

Heuristic, nature-inspired algorithms, such as particle swarm optimization (PSO) [4, 5] or genetic algorithms (GA) [6, 7], are attractive low-complexity solutions to facilitate MIMO channel estimation. PSO is a population-based heuristic global optimization algorithm, which originated in modeling the social behavior of bird flocks and fish schools. It has been applied to a variety of technical optimization problems, including channel and parameter estimation [8–13] as well as data detection [14] and multiuser detection [15]. Unfortunately, a fair evaluation of PSO is rather difficult due to the wide range of available modifications and the fact that the algorithm is often tuned to optimum performance for a specific optimization problem by empirical measures.

Genetic algorithms are inspired by natural evolution. Accordingly, population members are termed chromosomes. Based on an optimization metric, a subset of chromosomes

is selected to breed a new generation, which is subsequently used to generate a new generation by means of crossover and/or mutation.

PSO and GA share many similarities as both start with a randomly initialized population and both use a fitness value to evaluate their population members. The main difference lies within the selection of leaders (in terms of PSO) or parents (in terms of GA) as well as the update of position and/or generation of new members, respectively. Population members within PSO are updated iteratively and influence themselves directly by their personal best position. On the contrary, population members in GA pass characteristic information to their children.

It is difficult to compare the performance of PSO and GA in general as both depend on the specific optimization problem. Additionally, a similar variety of possible implementations exists also for GA. However, several publications in the field of digital communications come to the conclusion that PSO is advantageous compared to GA in terms of computational complexity, convergence speed, and accuracy [16–18]. Additionally, fewer parameters need to be set for the PSO algorithm.

PSO is a viable alternative to replace the closed-form solution of standard LS/MMSE estimators if it can provide similar performance at lower complexity. Nevertheless, the overwhelming variety of implementations makes a performance/complexity analysis cumbersome. This paper evaluates the applicability of PSO for MIMO channel estimation with respect to mean squared error (MSE) performance and computational complexity. The paper comprises the following central points.

- (i) An overview of the original PSO is given as well as general improvements which lead to a modified update function. Although PSO is widely adopted in many fields, mainly the original version of PSO is applied. Nevertheless, in many cases the performance can be improved and/or the required number of iterations can be reduced by more advanced versions. The focus is hereby on strategies which can be applied in general and do not have to be tuned for a specific optimization problem.
- (ii) While PSO has already been applied to MIMO channel estimation in the literature (e.g., [9]), an extension known as cooperative PSO (CPSO) [19] is introduced in this paper. Although PSO is directly applicable for some cases, advanced strategies, such as CPSO, are necessary either to provide optimum performance and/or a lower complexity.
- (iii) The performance of PSO and CPSO is compared with a conventional channel estimation algorithm, namely, the optimum MMSE estimator. Additionally, PSO/CPSO and an MMSE estimator are used to provide initial channel estimates for a graph-based iterative receiver. BER results illustrate the advantage of CPSO over PSO.
- (iv) A general rule to determine the maximum number of iterations is developed on the basis of the generalized

extreme value distribution. This approach allows the actual computation of the required number of iterations to reach a predefined target. To the authors best knowledge, a similar method to predict the required number of iterations does not yet exist, although it may be essential for the evaluation of the complexity.

- (v) The complexity of PSO/CPSO is discussed and compared to the conventional MMSE estimator. It is known that the complexity of PSO/CPSO per iteration is low. However, depending on the optimization problem several hundred iterations are required until convergence is achieved. Utilizing the proposed criterion to determine the maximum number of iterations, the complexity of PSO/CPSO can be evaluated for a wide range of parameter settings, and an optimum tradeoff between iterations and complexity per iteration can be determined.

The remainder of this paper is organized as follows: PSO and the extension to cooperative PSO is elaborated in Section 2. The application to MIMO channel estimation is described in Section 3. A performance as well as a complexity comparison of PSO/CPSO with an MMSE estimator is given in Sections 4 and 5, respectively. The extension to multiple objectives is discussed in Section 6. Finally, Section 7 draws the conclusion.

Throughout the paper, the following notation conventions are adopted. Bold-face capital and lower-face letters stand for matrices and vectors of appropriate dimensions, respectively. \mathbf{I}_{N_T} denotes a $N_T \times N_T$ identity matrix. Furthermore, $()^\dagger$ represents the Hermitian operator.

2. Particle Swarm Optimization

PSO is a population based, heuristic, iterative optimization algorithm. Due to the heuristic approach, no gradient information is required to converge to the global optimum. Hence, it can easily be adopted to a wide range of technical optimization problems.

In the following a general overview of PSO is given in Section 2.1. An extension termed cooperative PSO is introduced and applied for MIMO channel estimation in Section 2.2.

2.1. Standard PSO. The standard PSO is described by Algorithm 1. Initially all N_p particles of a swarm are randomly set throughout the feasible search region $[\mathbf{S}_{\min}, \mathbf{S}_{\max}]$, where $\mathbf{S} \in \mathbb{R}^D$. The particles of a swarm “fly” through a D -dimensional search space, which is gradually explored by adjusting the trajectory of each particle at each iteration. Within each iteration the current position of a particle $\mathbf{p}_i = [p_1, \dots, p_D]$ is used as a candidate solution for the optimization metric termed fitness function. The fitness value of a particle is distributed to all particles within the swarm. The previously best position of a particle is termed personal best \mathbf{p}_i^{PB} , whereas the previous best position of the

```

Initialize swarm
Locate leader
i = 1
while i < imax or convergence do
  for each particle do
    Update position using (1)/(2), (4)
    Evaluation using (5)
    Update pBest
  end for
  Update leader
  i++
end while

```

ALGORITHM 1: Standard PSO algorithm.

swarm is called global best \mathbf{p}^{GB} . The velocity vector of a particle i is updated according to [5, 20]:

$$\mathbf{v}'_i = \omega \mathbf{v}_i + c_1 \varepsilon_1 \circ (\mathbf{p}_i^{\text{IB}} - \mathbf{p}_i) + c_2 \varepsilon_2 \circ (\mathbf{p}^{\text{GB}} - \mathbf{p}_i), \quad (1)$$

where \circ denotes the entrywise product. The variables ε_1 and ε_2 denote random numbers in the range of $[0, 1]$. The inertia weight ω typically decreases from 0.9 to 0.4 over the course of iterations. The social and cognitive parameters c_1 and c_2 are acceleration coefficients towards the personal and/or global best position, respectively. The velocity vector of a particle is, similar to the search space, restricted within certain boundaries $[\mathbf{v}_{\min}, \mathbf{v}_{\max}]$. Particles which are beyond the boundaries of search space and velocity, respectively, are reset to the boundary limits.

The update function (1) was published in 1998 as part of an already revised version of the PSO algorithm. The original update function of PSO published in 1995 [4] did not include the inertia weight or the cognitive and social parameters. Since then, an overwhelming amount of variations have been proposed. However, no standard algorithm or set of parameters has yet emerged, which delivers optimum performance independent of the optimization problem. Hence, parameters are tuned for each specific problem and settings tuned by means of empirical measures are often applied. The authors of [21] propose a so-called standardized version of PSO which incorporates several general applicable improvements, that is, bound handling, swarm size, and an update equation replacing the inertia weight with a constriction factor. The standardized version improves the performance for most optimization problems compared to the original version. We restrict ourselves to general applicable optimization for PSO, although adaptive versions [22] are also able to improve the performance of the standard PSO, but their parameters may be optimized for each optimization problem.

The update rule based on the constriction factor is given by

$$\mathbf{v}'_i = \chi \{ \mathbf{v}_i + c_1 \varepsilon_1 \circ (\mathbf{p}_i^{\text{IB}} - \mathbf{p}_i) + c_2 \varepsilon_2 \circ (\mathbf{p}^{\text{GB}} - \mathbf{p}_i) \}, \quad (2)$$

with

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi} \right|}, \quad (3)$$

where $\varphi = c_1 + c_2$, $\varphi > 4$. The factors c_1 and c_2 are constraints on the velocity towards the global and the personal best position. According to [23], suitable values for a wide range of test functions are as follows: $c_1 = 2.8$ and $c_2 = 1.3$, which results in $\chi \approx 0.7298$. The standardized update function (2) as well as the above-mentioned parameters are applied throughout all simulations. The position of a particle is updated subsequently according to

$$\mathbf{p}'_i = \mathbf{p}_i + \mathbf{v}'_i. \quad (4)$$

The updated velocity vector \mathbf{v}'_i is added to the current position \mathbf{p}_i of a particle. The new position \mathbf{p}'_i is used as a candidate solution for the optimization metric. The optimization performed by PSO is described by

$$p^{\text{OPT}} = \arg \min_{\mathbf{p}_i} f(\mathbf{p}_i). \quad (5)$$

The so-far emerged personal and/or global best \mathbf{p}_i^{IB} and \mathbf{p}^{GB} , respectively, are replaced by the updated position \mathbf{p}'_i , if the fitness value p^{OPT} is improved compared to the values of the personal and the global best position. This procedure is repeated until PSO is converged or the maximum number of iterations i_{\max} is reached. i_{\max} is chosen to be sufficiently large to prevent that the algorithm is stopped before the global optimum could be found. Frequently, the optimum solution is found with just a fraction of i_{\max} . Therefore, a stopping criterion is necessary to reduce the average number of iterations needed for convergence. An overview of suitable stopping criteria is given in [24]. In this paper, PSO is said to be converged if p^{OPT} is below a certain threshold t_h for γ iterations.

In case PSO converges, all particles \mathbf{p} of the swarm are located at the same position minimizing (5). Without loss of generality, only minimization problems are considered.

2.2. Cooperative PSO. In general, population-based optimization algorithms are searching for a region of small, specified volume in a D -dimensional search space, surrounding the global optimum. In order to converge to the global optimum, an optimization algorithm needs to create a sample within this region. The probability of generating a sample within the region is the volume of the region divided by the volume of the search space [19]. This probability decreases exponentially with increasing dimensionality of the search space. This effect is often termed ‘‘curse of dimensionality.’’

Separating the high-dimensional search space into sets of smaller dimension improves the performance significantly. PSO is known to perform rather poor for high-dimensional problems. A large variety of solutions is proposed to solve this problem. In [25] the update function (1) is adapted to take adaptive parameters into account. These parameters are changed over the course of iterations and improve the

converge behavior of PSO algorithm. However, the optimum set of parameters remains problem dependent. An alternative solution to improve the performance of the original PSO algorithm is given by a so-called cooperative approach to particle swarm optimization (CPSO) presented in [19]. The CPSO approach relies on the original update equation and is described in the following. The pseudocode describing CPSO is given by Algorithm 2. The N_p particles of the PSO swarm are now separated into N_s swarms with N'_p particles. The number of particles for both PSO and CPSO should be chosen within a certain range. Too few particles ($N_p, N'_p < 5$) lead to a deteriorated performance, while too many are not able to increase the performance ($N_p, N'_p > 100$). About 15 particles is a good tradeoff between complexity and performance [23].

Accordingly, the D -dimensional problem is split into N_s subsets and optimized separately by an individual swarm of particles $\mathbf{s} = [s_1, \dots, s_{D/\delta}]$, where δ is the number of dimensions for each swarm. The position of a particle i of swarm s is given by $\boldsymbol{\rho}_{s,i} = [\rho_1, \dots, \rho_\delta]$. The separation of the dimensions is mitigating a drawback of the standard PSO algorithm: Since the standard PSO considers the full-dimensional vector in the update function, it allows that some dimensions move further away from the solution as long as the overall fitness value is improved. On the contrary, cooperative PSO is evaluating subsets of the D -dimensional vector. The probability that single components are deteriorated in favor of other dimensions is thus reduced.

For $N_s = 1$ swarm, CPSO is equivalent to PSO since all dimensions are optimized by one swarm. In case of $N_s > 1$, the evaluation of the optimization metric is not directly possible since a particle represents only a subset of dimensions of the optimization problem. Consequently, a context vector $\phi_{s,i}$ is necessary. In order to construct a D -dimensional vector, the $D - \delta$ missing dimensions are replaced by the global best positions of the remaining swarms: $\phi_{s,i} = [\boldsymbol{\rho}_1^{\text{GB}} \dots \boldsymbol{\rho}_{s,i} \dots \boldsymbol{\rho}_{D/\delta}^{\text{GB}}]$. The optimization function (5) is changed accordingly:

$$p^{\text{OPT}} = \arg \min_{\phi_{s,i}} f(\phi_{s,i}) . \quad (6)$$

3. Application to MIMO Channel Estimation

We consider a MIMO system with N_T transmit and N_R receive antennas. The received signal vector at time index k , $\mathbf{y}[k] \in \mathbb{C}^{N_R \times 1}$, is modeled as

$$\mathbf{y}[k] = \mathbf{H}\mathbf{x}[k] + \mathbf{n}[k], \quad (7)$$

where $\mathbf{x}[k] \in \mathbb{C}^{N_T \times 1}$ is the transmitted signal vector at time index k . The entries of the channel matrix $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ are assumed to be independent and identically distributed (i.i.d.) according to $\mathcal{C}\mathcal{N}(0, 1)$. We consider a quasi-time-invariant channel for the numerical analysis of PSO and CPSO. The application to time-varying channels is given briefly by the description of a multiple-objective PSO in Section 6. Furthermore, only a memoryless channel is considered in the numerical results in order to simplify the optimization metric and the discussion of the results.

```

Initialize  $N_s$  swarms with  $N'_p$  particles
Locate leader
 $i = 1$ 
while  $i < i_{\text{max}}$  or convergence do
  for each swarm do
    for each particle do
      Update position using (1)/(2), (4)
      Evaluation using (6)
      Update pBest
    end for
  end for
  Update leader
   $i++$ 
end while

```

ALGORITHM 2: Cooperative PSO algorithm.

The complexity analysis is directly applicable to frequency-selective channels as well, since only the dimensionality of the problem is discussed here. Dimensions can be increased by either transmit and receive antennas and/or the channel memory length L . The MSE performance for a channel memory length $L > 0$ will follow the MSE performance of a least-squares estimator. For detailed information about the application of PSO to channels with memory, interested readers are referred to [12]. Furthermore, $\mathbf{n}[k]$ denotes the noise vector at the receiver whose entries are i.i.d. modeled as $\mathcal{C}\mathcal{N}(0, \sigma_n^2)$.

Training symbols are transmitted to support pilot-aided channel estimation (PACE). Stacked in a matrix, the transmit vector $\mathbf{x}[k]$ can be written as $\mathbf{X} \in \mathbb{C}^{N_T \times N_T}$. A minimum of N_T training symbols are transmitted to ensure a full rank. The training matrix consists of orthogonal sequences subject to $\mathbf{X}\mathbf{X}^\dagger = \mu\mathbf{I}_{N_T}$, where μ is related to the signal power assigned to the training matrix [26].

In the following, we assume that the transmit vector $\mathbf{x}[k]$ of length N_T consists of training symbols only.

In case of a quasi-invariant (block-fading) channel, the maximum-likelihood metric (fitness function) for PSO can be written as follows:

$$f(\mathbf{p}_i) = \sum_{k=1}^{N_T} \|\mathbf{y}[k] - \mathbf{P}_i\mathbf{x}[k]\|^2. \quad (8)$$

The position of the i th particle \mathbf{P}_i is used as a potential solution for the metric. For a consistent notation in line with (7), the previously used vector notation of the position of the particle is changed here to a matrix notation with $\mathbf{P}_i \in \mathbb{C}^{N_R \times N_T}$. Thus, a position of a particle represents a hypothesis of the channel matrix $\tilde{\mathbf{H}}$. It is of importance to note that each dimension of a particle is real valued. As a particle needs to estimate $N_R \times N_T$ complex-valued channel coefficients, the dimensions of the search space results in $D = 2 \cdot N_T \cdot N_R$.

The maximum-likelihood metric for CPSO is very similar to the PSO metric. Instead of using the position of a particle, a context matrix $\phi_{s,i}$ is utilized, since a position

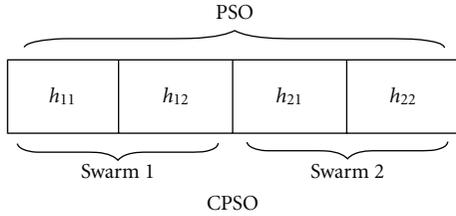


FIGURE 1: Possible separation of an 8-dimensional problem into a set of lower-dimensional problems by the CPSO compared to PSO.

of a particle of a subswarm represents only a fraction of the channel matrix:

$$f(\phi_{s,i}) = \sum_{k=1}^{N_T} \|\mathbf{y}[k] - \phi_{s,i} \mathbf{x}[k]\|^2. \quad (9)$$

In case of MIMO channel estimation, $N_R \cdot N_T$ channel coefficients are estimated assuming a flat-fading time-invariant channel. As mentioned before, the performance of PSO is degraded with increasing dimensions. The dimensionality of the optimization problem is not only determined by the number of transmit and/or receive antennas but also by the channel memory length. Thus, channel estimation of a realistic MIMO system may result in a high-dimensional optimization problem.

Figure 1 illustrates the difference between PSO and CPSO for channel estimation of a 2×2 MIMO system. PSO optimizes all channel coefficients with one swarm. CPSO is able to separate the $D = (2 \cdot N_R \cdot N_T)$ -dimensional problem into subsets and optimizes each subset with a separate swarm. In this example two swarms are shown; however, the number of possible subswarms is in the range of $[1, D]$. In the case of D subswarms, a single swarm would optimize either real or imaginary part of one channel coefficient. While the number of subswarms N_s is directly related to the number of dimensions, there is no such relation for the number of particles. A minimum number of particles need to be assigned for each subswarm in order to allow convergence. Additionally, the performance of CPSO cannot be improved by increasing the number of particles once a threshold is reached. The number of particles depends again on the optimization problem. A good tradeoff between complexity per iteration and performance is $N_p = N'_p = 15$ [23].

4. Performance Comparison

A performance comparison of PSO and CPSO with an optimum MMSE channel estimator in terms of mean squared error (MSE) is given in Figure 2. The simulation setup consists of an MIMO system and a quasi-time-invariant Rayleigh fading channel. Pilot-aided channel estimation is conducted with orthogonal training sequences. The number of particles is fixed to $N_p = 60$ particles for PSO algorithm. The CPSO algorithm consists of $N_s = 4$ subswarms with $N'_p = 15$ particles for each swarm. The overall complexity of CPSO compared to PSO for one iterations is thus the same.

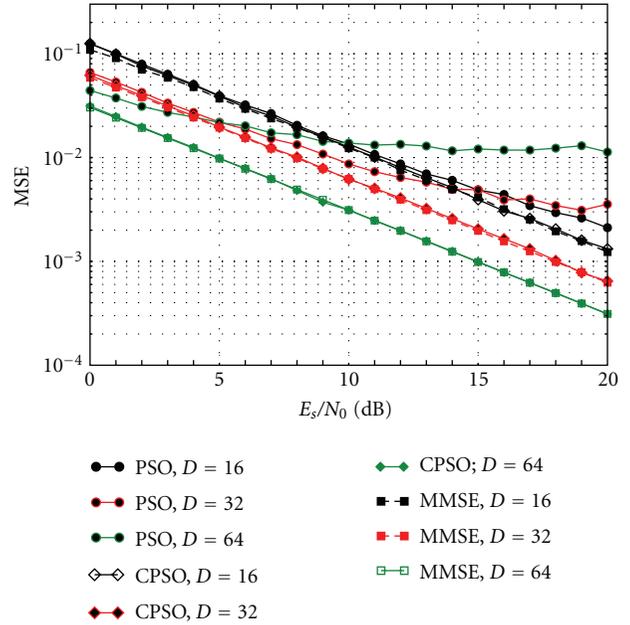


FIGURE 2: MSE performance of PSO and CPSO compared to MMSE channel estimation depending on the number of dimensions for a quasi-time-invariant Rayleigh fading channel.

The MMSE estimate of \mathbf{H} for a time-invariant channel is given by

$$\hat{\mathbf{H}} = \mathbf{YX}^\dagger \left[\sigma_n^2 \mathbf{I}_{N_T} + \mathbf{X}\mathbf{X}^\dagger \right]^{-1}. \quad (10)$$

PSO exhibits an inferior performance compared to CPSO and/or MMSE estimation with increasing SNR and dimensions D as illustrated in Figure 2. In this case, the dimensions are increased by increasing the number of transmit antennas with a constant number of receive antennas. The larger the dimensions the earlier PSO converges to an error floor. Large dimensions ($D > 32$) can easily be reached with settings defined in upcoming wireless standards, that is, long-term evolution advanced (LTE-A) [1]. For example, for an 8×8 -MIMO-OFDM system with a channel memory length of $L = 0$ the dimensions results in $D = 8 \cdot 8 \cdot 2 = 128$. Only the dimensions are studied to focus the reader on the problem of large dimensions for PSO. The performance of the original PSO can be further improved by adapting the variables in (1) and/or (2) over the course of iterations and/or by applying more advanced bound handling mechanisms. These optimization methods for PSO have to be tuned to the specific optimization metric. On the other hand, the performance of CPSO reaches the optimum MMSE performance with general settings.

It is worth mentioning that, although GA does not suffer from the curse of dimensionality and is thus able to converge in general, the required number of iterations increases, which renders a practical implementation unfeasible.

Besides reaching the optimum performance, another strength of the PSO algorithm is its fast convergence to a "reasonable" MSE. The fast converging nature of PSO/CPSO

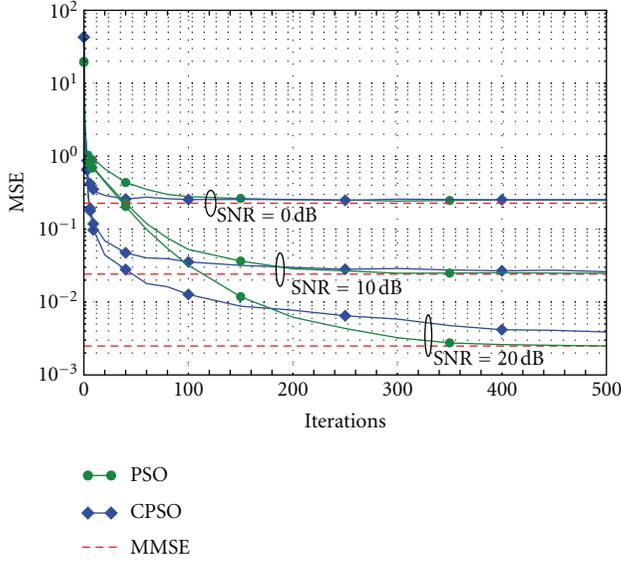


FIGURE 3: Convergence speed comparison of PSO and CPSO for different SNR values compared to the optimum MMSE performance with a 4×4 MIMO system.

can be seen in Figure 3, where a PSO algorithm with $N_p = 60$ particles as well as a CPSO algorithm with $N_s = 4$ and $N'_p = 15$ is used to estimate a 4×4 -MIMO channel. The MSE improves significantly in the first 20, 60, and 120 iterations for CPSO depending on the SNR. The subsequent iterations are needed for convergence to the optimum performance. Interestingly, PSO converges earlier to the optimum performance in case of an SNR of 20 dB. The CPSO algorithm is attracted to local optima which are created by the separation of the dimensions [19]. The convergence speed, is slower in this case. The fast convergence is especially beneficial when PSO/CPSO is used as an initialization algorithm where only a good starting value is needed instead of the optimum value. In order to further illustrate the advantages of CPSO over PSO, the graph-based soft iterative receiver (GSIR) of [3] is initialized by PSO, CPSO, and the MMSE estimator in the following. A 8×8 MIMO system is considered with QPSK modulation and a rate-1/2 repetition code. A fixed number of 10 iterations are used for the GSIR. A training preamble with orthogonal properties, as described in Section 3, is applied. A data sequence of length $K_D = 100$ is transmitted subsequently. Thus, the complete transmitted sequence can be represented as $\mathbf{X} = [\mathbf{X}^T \mathbf{X}^D]$. PSO, CPSO, and the MMSE estimator are applied once, utilizing only the training symbols given in \mathbf{X}^T and provide the initial channel coefficient estimates used by the GSIR. The maximum number of iterations i_{\max} for the PSO/CPSO is hereby restricted to keep the computational overhead of the initialization at a minimum. The BER results are shown in Figure 4. It is obvious that CPSO outperforms PSO in terms of convergence speed. A maximum of only 30 iterations is required for the CPSO to provide initial coefficients that are sufficiently well for the GSIR to converge to the same performance achieved

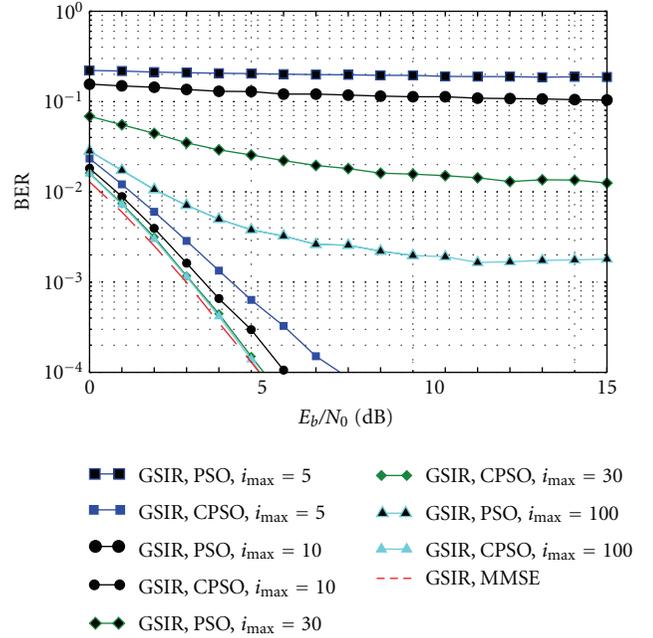


FIGURE 4: BER performance of the GSIR initialized by PSO, CPSO, and MMSE as a function of the maximum number of iterations.

with an MMSE initialization. As suggested by the results given in Figure 3, the performance of PSO/CPSO depends on the number of particles and subswarms. The relation between the number of particles/subswarms on the number of iterations is investigated in the following section. In order to keep the computational overhead at a minimum the maximum number of iterations for the PSO/CPSO should be set to a minimum.

5. Complexity Analysis

The complexity of PSO/CPSO is determined by the number of particles, subswarms, dimensions, and the required number of iterations for convergence. The number of particles and subswarms is a design parameter of the algorithm and is commonly chosen to achieve a good performance in terms of MSE for channel estimation. The number of dimensions is a fixed parameter depending on the optimization problem (e.g., number of transmit and receive antennas and/or channel memory length). In each iteration all particles N'_p of all subswarms N_s have to evaluate their current position and compare their current fitness value with their personal best as well as the global best, which results in a complexity of order

$$\mathcal{O}(\text{CPSO}^{(it)}) = \mathcal{O}(N'_p \cdot N_s \cdot D), \quad (11)$$

per iteration.

The overall number of particles influences the number of iterations needed to converge. In case of using only one particle the required number of iterations until convergence is maximized and computational complexity per iteration is minimized, while, on the other hand, using an infinite

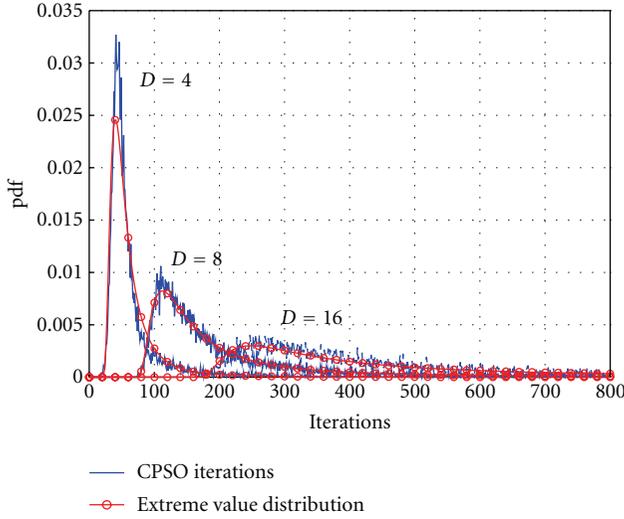


FIGURE 5: Histogram of the minimum number of iterations required to converge in dependence of the dimensionality.

number of particles is minimizing the number of iterations and maximizing the computational complexity per iteration. With an infinite number of particles, PSO is equivalent to exhaustive search. Hence, a tradeoff between the overall size of PSO/CPSO and the number of iterations has to be found. Furthermore, the required minimum number of iterations is depending on the optimization metric as well. In general, the more complex (higher dimensional) the optimization problem is, the more iterations are needed and vice versa.

A strategy often used to determine the maximum number of iterations i_{\max} is to find the minimum value of iterations at which the optimum MSE performance is reached. This approach requires extensive simulations over a variety of parameters in order to determine the optimum tradeoff between complexity and iterations.

In the following a general criterion to determine the maximum number of iterations based on the probability distribution function of the iterations required by PSO/CPSO for convergence is presented. The advantage of this strategy is that only a fraction of parameters need to be simulated while missing parameters can be reconstructed by means of an interpolation. PSO/CPSO is said to reach convergence if the fitness value p^{OPT} of (8)/(9) is below a certain threshold t_h for γ iterations. In this case the threshold is set to $t_h = 10^{-6}$ with $\gamma = 10$.

Monte Carlo simulations with a fixed parameter set for CPSO and varying number of transmit antennas are conducted. The iteration at which the stopping criterion is fulfilled is recorded. A histogram of the iterations fulfilling the stopping criterion for different dimensions is shown in Figure 5. Each histogram can be approximated by a generalized extreme value distribution. The characteristic shape of the function is in the steep slope once a certain value is exceeded and a slow decline after the maximum is reached. The probability density function (pdf) of the generalized extreme value distribution is described by (12). The

distribution is characterized by three parameters, namely, the shape parameter k , the scale parameter σ , and the location parameter μ :

$$p(k, \mu, \sigma) = \left(\frac{1}{\sigma}\right) \exp\left(-\left(1 + k \frac{(x - \mu)}{\sigma}\right)^{-1/k}\right) \times \left(1 + k \frac{(x - \mu)}{\sigma}\right)^{-1 - (1/k)}. \quad (12)$$

Given the pdf for a certain parameter set, the maximum number of iteration i_{\max} can be defined to cover a certain percentage of the pdf. The amount to which the pdf is covered defines the tradeoff between performance and complexity. Setting the maximum number of iterations too low reduces the complexity of the algorithm, but also implies a performance loss due to a premature stop of the algorithm. Vice versa, setting the maximum number of iterations too large is increasing complexity without a gain in performance. In case of $D = 4$ (cf. Figure 5 the location parameter results in $\mu = 45$, which resembles the most likely iteration at which the algorithm converges. In order to cover at least 90% of the required iterations the maximum number of iterations should be set to $i_{\max} \geq 105$.

The aforementioned tradeoff between the number of particles/subswarms and the number of iterations is evaluated in the following. The maximum number of iterations is defined to cover 90% of the pdf.

The number of iterations required by PSO/CPSO until convergence depends on the number of dimensions of the optimization problem and the allocated number of swarms and inherently particles. In Figure 6, the required number of iterations depending on the dimensions of the optimization problem is given for different swarm sizes. With a constant swarm size the iterations are increasing quadratically with the dimensions. On the contrary, with increasing swarm sizes, the required iterations are nearly constant over the dimensions, as can be seen from the similar starting points of the curves. The required number of iterations for PSO ($N_s = 1$) to converge, exceeds 8000 at 20 dimensions. Since the three parameters of the extreme value distribution are correlated over the number of particles and subswarms, not all swarm sizes need to be simulated but can be calculated by means of interpolation. The optimum tradeoff between swarm size and iterations can thus be determined with a minimum amount of simulations.

The overall complexity of CPSO depends on the complexity per iteration and the number of iterations:

$$\mathcal{O}(\text{CPSO}^{\text{(total)}}) = \mathcal{O}(\text{CPSO}^{\text{(it)}}) \cdot \mathcal{O}(I). \quad (13)$$

The number of iterations is taken into account here, as they are influenced by the dimensionality of the optimization problem. The complexity of the MMSE is dominated by the matrix inversion which has a complexity of order

$$\mathcal{O}(\text{MMSE}) = \mathcal{O}(N_T^3). \quad (14)$$

A fair comparison of the complexity is not straightforward, when for example the number of complex multiplications

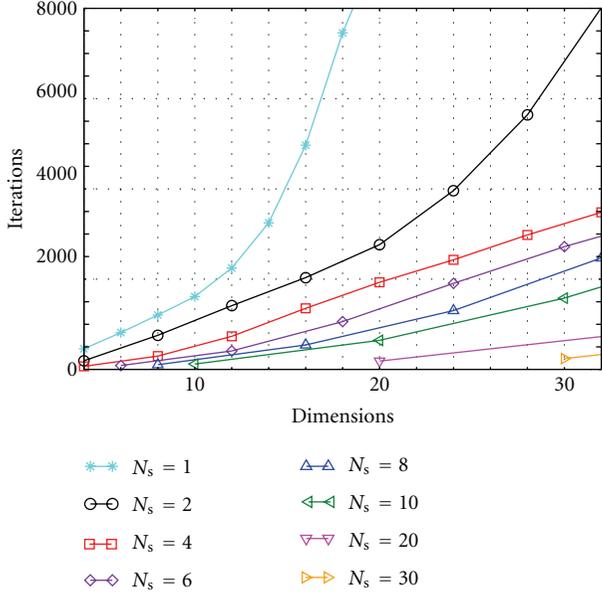


FIGURE 6: Required number of iterations of different swarm sizes as a function of the number of dimensions at an SNR of 10 dB.

is considered, as different implementations as well as optimization techniques for both algorithms will end in different results. On basis of the \mathcal{O} -notation, CPSO and MMSE are compared to give an insight on the influence of the different parameters of CPSO. The total complexity of CPSO is separable into two parts as can be seen from (13), namely, into the complexity per iteration and the complexity introduced by the number of iterations. The complexity of CPSO per iteration is increased linearly with the number of transmit antennas. It is obvious that the complexity of the MMSE will eventually be larger than the complexity of CPSO per iteration with increasing number of transmit antennas.

However, the total complexity of CPSO will only be lower for a given number of transmit antennas if the number of iterations required by CPSO for convergence is small. An assumption that is fulfilled when CPSO is used for initialization.

With increasing number of transmit antennas a larger number of subswarms/particles can be supported with a lower complexity than the MMSE estimator. Using a criterion for optimum performance and not fast convergence (initialization) is further increasing the complexity of CPSO. Hence, optimum performance with a complexity lower than MMSE is only reached for larger MIMO systems.

The following conclusions can be drawn from this result.

- (i) CPSO is suitable for medium- to large-sized MIMO systems when used for initialization.
- (ii) Channel estimation with PSO/CPSO targeting optimum performance is computational feasible for MIMO systems with several dozens of transmit antennas. Such scenarios are often referred to as Large-MIMO systems [27].

6. Multiobjective PSO

High-dimensional optimization problems can be solved efficiently by the aforementioned cooperative particle swarm optimization algorithm. However, PSO as well as CPSO are limited to solve single-objective problems. A constraint that is fulfilled for channel estimation given quasi-time-invariant channels. With time-varying channels, the need for multi-objective optimization arises, since the optimization metric (8) and/or (9) can no longer be minimized by a single constant matrix. Specifically, this means that the position of a particle, which is used as a candidate solution, may be exact for one time index but will not for a second time index.

Considering the coefficients over the time as additional dimensions is infeasible due to the inherent increase of iterations and complexity. However, due to the correlation of adjacent channel coefficients in the time domain a multiobjective particle swarm optimization (MOPSO) can be applied. The fitness function is changed for MOPSO to minimize K objectives simultaneously:

$$f(\mathbf{p}_i[k]) = \|\mathbf{y}[k] - \mathbf{p}_i[k]\mathbf{x}[k]\|^2, \quad 1 \leq k \leq K, \quad (15)$$

where K represents the number of training symbols in case of pilot-aided channel estimation of a time-varying channel. Obviously, the candidate solution $\mathbf{p}_i[k]$ may minimize the k th objective but is not necessarily optimal for the $(k+1)$ th objective. This means that one objective cannot be optimized without neglecting the performance of at least one other objective. The previous concept of one global best position \mathbf{p}^{GB} is replaced by an archive \mathbf{F}^* with the so-called dominant solutions for all objectives. A particle \mathbf{p}_i is said to dominate another particle \mathbf{p}'_i , denoted as $\mathbf{p}_i \succ \mathbf{p}'_i$, if and only if

- (1) $\forall \lambda \in \{1, \dots, \Lambda\} : \mathbf{p}_i \leq \mathbf{p}'_i$,
- (2) $\exists \lambda' \in \{1, \dots, \Lambda\} : \mathbf{p}_i < \mathbf{p}'_i$.

In each iteration an updated candidate solution is compared to the solutions stored in the external archive. A candidate solution is added to the archive if it improves at least one objective compared to the already existing solutions. A candidate solutions replaces an existing solution if it improves all objectives of an existing solution. The set of solutions contained in the archive is termed Pareto set:

$$\mathbf{F}^* \doteq \{\mathbf{p}_i \in \mathbb{R}^D \mid \nexists \mathbf{p}'_i \in \mathbb{R}^D : \mathbf{p}'_i \succ \mathbf{p}_i\}. \quad (16)$$

It is of importance that all objectives are optimized equally well by the particles of a swarm. Without an additional control mechanism to ensure a certain degree of diversity (quality) within the archive, all particles may concentrate on one objective, equivalent to the single-objective PSO. A diverse solution set can be found by applying the so-called sigma method [28]. The idea of the sigma method is that each point of the D -dimensional search space is assigned a sigma value. The Euclidean distance of the sigma value of a current particle and the sigma value of the archive members is calculated. The leader of the current particle is determined by the smallest Euclidean distance between the sigma value of the current position and the sigma value of a member of the archive.

```

Initialize swarm
Locate leader in an external archive
i = 1
while i < imax or convergence do
  for each particle do
    Select leader from archive
    Update position using (1)/(2), (4)
    Mutation
    Evaluation using (15)
    Update pBest
  end for
  Update leaders in the external archive
  Quality (leaders)
  i++
end while

```

ALGORITHM 3: Multiobjective PSO algorithm.

The principle of MOPSO is described by Algorithm 3. The update function is hereby unchanged to PSO/CPSO. An additional mutation operator is recommended as the MOPSO algorithm occasionally converges prematurely. The mutation operator is used at certain iteration intervals and increases the velocity of a particle and/or reinitializes a particle. The maintenance and the additional mutation operator contribute to an increased complexity of the algorithm. Nevertheless, a multi-objective PSO (MOPSO) is successfully used in [29] to initialize a graph-based iterative receiver with only 5 MOPSO iterations to achieve the optimum performance with the subsequent graph-based receiver.

7. Conclusion

An overview of particle swarm optimization for MIMO channel estimation has been given. General applicable solutions for MIMO channel estimation are presented as well as the achievable performance of the algorithms is evaluated by means of Monte Carlo simulations. Furthermore, an analysis of the complexity based on the distribution of the required iterations until convergence is introduced. The proposed method allows the calculation of a maximum number of iterations with a minimum of simulation overhead, since missing parameters can be reconstructed by means of an interpolation.

It has been shown that cooperative PSO is able to approach the optimum MMSE estimator. Thus, for a potential implementation, the required number of iterations are of utmost importance. The presented MSE and BER results further illustrate that CPSO is also able to converge fast to a “reasonable” MSE, which allows an iterative receiver to converge to the same performance achieved with an MMSE-based initialization, with just a minimum of iterations required for the CPSO. The advantage of CPSO over MMSE is in the flexible tradeoff between complexity per iteration and required number of iterations, which makes it ideally suited for parallelization. Furthermore, the parameters for CPSO do not need to be tuned by empirical measures, which

is an advantage of CPSO, since the algorithm can directly be applied for MIMO channel estimation.

Although PSO/CPSO can be used to estimate time-varying channels by utilizing the extension to multiple objectives, the strength of PSO/CPSO lies in the estimation of time-invariant channels.

References

- [1] 3GPP TS 36.300 version 10.0.0, 2010.
- [2] J. Ylioinas and M. Juntti, “Iterative joint detection, decoding, and channel estimation in turbo-coded MIMO-OFDM,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1784–1796, 2009.
- [3] C. Kniewel, Z. Shi, P. A. Hoeher, and G. Auer, “2D graph-based softchannel estimation for MIMO-OFDM,” in *Proceedings of the IEEE International Conference on Communications (ICC ’10)*, Cape Town, South Africa, May 2010.
- [4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [5] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, 2001.
- [6] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.
- [7] M. Y. Alias, S. Chen, and L. Hanzo, “Multiple-antenna-aided OFDM employing genetic-algorithm-assisted minimum bit error rate multiuser detection,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 5, pp. 1713–1721, 2005.
- [8] Y. Gao, Z. Li, X. Hu, and H. Liu, “A multi-population particle swarm optimizer and its application to blind multichannel estimation,” in *Proceedings of the 3rd International Conference on Natural Computation (ICNC ’07)*, Haikou, China, August 2007.
- [9] S. Tao, X. Jiadong, and Z. Kai, “Blind MIMO identification using particleswarm algorithm,” in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom ’07)*, Shanghai, China, September 2007.
- [10] H. Bodur, C. A. Tunc, D. Aktas, V. B. Erturk, and A. Altintas, “Particleswarm optimization for SAGE maximization step in channel parameterestimation,” in *Proceedings of the 2nd European Conference on Antennas and Propagation (EuCAP ’07)*, Ankara, Turkey, November 2007.
- [11] W. Dong, J. Li, and Z. Lu, “Joint frequency offset and channel estimationfor MIMO systems based on particle swarm optimization,” in *Proceedings of the IEEE 67th Vehicular Technology Conference (VTC ’08-Spring)*, Singapore, May 2008.
- [12] K. Schmeink, R. Block, C. Kniewel, and P. A. Hoeher, “Joint channel andparameter estimation for combined communication and navigation usingparticle swarm optimization,” in *Proceedings of the 7th Workshop on Positioning Navigationand Communication (WPNC ’10)*, Dresden, Germany, March 2010.
- [13] C.-H. Cheng, H.-C. Hsu, Y.-F. Huang, J.-H. Wen, and L.-C. Hsu, “Performance of an adaptive PSO parallel interference canceller forCDMA communication systems,” in *Proceedings of the 5th Annual ICST WirelessInternet Conference (WICON ’10)*, Singapore, March 2010.
- [14] H. Palally, S. Chen, W. Yao, and L. Hanzo, “Particle swarm optimisationaided semi-blind joint maximum likelihood channel estimation and datadetection for MIMO systems,” in

- Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing (SSP '09)*, pp. 309–312, September 2009.
- [15] K. K. Soo, Y. M. Siu, W. S. Chan, L. Yang, and R. S. Chen, “Particle-swarm-optimization-based multiuser detector for CDMA communications,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 3006–3013, 2007.
- [16] L. D. Orazio, *Study and development of novel techniques for PHY layer optimization of smart terminals in the context of next-generation mobile communications*, Ph.D. thesis, University of Trento, Trento, Italy, 2008.
- [17] N. Liangfang and D. Sidan, “Evolutionary particle swarm algorithm based on higher-order cumulant fitting for blind channel identification,” in *Proceedings of the Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, Dalian, China, October 2008.
- [18] W. Qiang, Z. Jiashu, and Y. Jing, “Identification of nonlinear communication channel using an novel particle swarm optimization technique,” in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 1162–1165, December 2008.
- [19] F. van den Bergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [20] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, USA, May 1998.
- [21] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 120–127, April 2007.
- [22] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, “Adaptive particle swarm optimization,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 6, pp. 227–234, 2009.
- [23] A. Carlisle and G. Dozier, “An off-the-shelf PSO,” in *Proceedings of the Particle Swarm Optimization Workshop*, pp. 1–6, April 2001.
- [24] K. Zielinski and R. Laur, “Stopping criteria for a constrained single-objective particle swarm optimization algorithm,” *Informatica*, vol. 31, no. 1, pp. 51–59, 2007.
- [25] T. Hendtlass, “Particle swarm optimization and high dimensional problems,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, Trondheim, Norway, May 2009.
- [26] B. Hassibi and B. M. Hochwald, “How much training is needed in multiple-antenna wireless links?” *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 951–963, 2003.
- [27] S. K. Mohammed, A. Zaki, A. Chockalingam, and B. S. Rajan, “High rate space-time coded large-MIMO systems: low-complexity detection and channel estimation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 6, pp. 958–974, 2009.
- [28] S. Mostaghim and J. Teich, “Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO),” in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, Indianapolis, Ind, USA, April 2003.
- [29] C. Kniewel, P. A. Hoeher, G. Auer, and A. Tyrrell, “Particle swarm enhanced graph-based channel estimation for MIMO-OFDM,” in *Proceedings of the IEEE Vehicular Technology Conference (VTC '11-Spring)*, Budapest, Hungary, May 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

