Cosmological parameter estimation using Particle Swarm Optimization (PSO)

Jayanti Prasad* and Tarun Souradeep†
IUCAA, Post Bag 4, Ganeshkhind, Pune 411007, India.
(Dated: August 30, 2011)

Obtaining the set of cosmological parameters consistent with observational data is an important exercise in current cosmological research. It involves finding the global maximum of the likelihood function in the multi-dimensional parameter space. Currently sampling based methods, which are in general stochastic in nature, like Markov-Chain Monte Carlo(MCMC), are being commonly used for parameter estimation. The beauty of stochastic methods is that the computational cost grows, at the most, linearly in place of exponentially (as in grid based approaches) with the dimensionality of the search space. MCMC methods sample the full joint probability distribution (posterior) from which one and two dimensional probability distributions, best fit (average) values of parameters and then error bars can be computed. In the present work we demonstrate the application of another stochastic method, named Particle Swarm Optimization (PSO), that is widely used in the field of engineering and artificial intelligence, for cosmological parameter estimation from WMAP seven years data. We find that there is a good agreement between the values of the best fit parameters obtained from PSO and publicly available code COSMOMC. However, there is a slight disagreement between error bars mainly due to the fact that errors are computed differently in PSO. Apart from presenting the results of our exercise, we also discuss the merits of PSO and explain its usefulness in more extensive search in higher dimensional parameter space.

PACS numbers:

I. INTRODUCTION

In a typical CMBR data analysis pipeline firstly the time order data, obtained from an instrument like WMAP, is reduced into a set of sky maps from which angular power spectra are computed, and finally these spectra are reduced into a set of cosmological parameters representing a model usually using Bayesian analysis [1–7, 7–9]. The exercise of parameter estimation involves identifying a set of parameters which has the highest probability of giving the observed data i.e., finding a point in the multidimensional parameter space at which the likelihood function has the greatest value. Currently this exercise is done using some sampling based methods, like MCMC, in which the likelihood function is sampled at discrete points, which are further used to compute various statistics of parameters [3, 6, 10]. Apart from MCMC, non-sampling based methods inspired from artificial intelligence techniques, like artificial neural network, have also been successfully applied in cosmological parameter estimation from the CMBR data [11].

In the present work we demonstrate the use of another artificial intelligence inspired method, named Particle Swarm Optimization or PSO [12–14] , for cosmological parameter estimation using WMAP seven years data[15]. Being a stochastic method, PSO also has the interesting feature that the computational cost for searching the global maximum in the multi-dimensional space does not grow exponentially with the dimensionality of the search space. However, in this case also (like other stochastic methods) the probability of convergence to the global maximum is usually guaranteed only in the asymptotic limit. Based on a very simple idea and having very few design parameters, PSO is quite easy to program and can provide accurate results very fast. As compared to artificial neural network [11] all the calculations in PSO are exact, in the sense that no extrapolation or interpolation is done at any stage i.e., fitness function is computed exactly at all points. It has been found that PSO can outperform MCMC in certain situations, in particular when there are a large number of local maxima present and/or the dimensionality of the search space is very high [16].

Since the fitness function i.e., the function to be optimized, can be computed concurrently on a parallel platform for a large number of particles, PSO promises to give accurate results very fast, if implemented efficiently. Unlike MCMC which provides the full probability distribution from which marginalized values of parameters and error bars can be computed, PSO just gives the location of the best fit point, called the `Gbest` , and one needs to find some way to compute error bars. In the present work we fit the effective chi square ($\chi^2_{\rm eff} = -2\log\mathcal{L}$) by a multi-dimensional paraboloid around the best fit point and compute the error bars from the fitting coefficients of that.

We observed that in most cases (which we have considered) not only the `Gbest` approach towards the best fit point, the average location of particles also approaches towards that, as PSO progresses. The average location of PSO particles in the multi-dimensional search space can be used to check the robustness of PSO. Since PSO is designed mainly for finding the best fit point therefore the sampling done by PSO particles is not designed to be fair representation of the likelihood surface.

The plan of the paper is as follows. In § II we discuss a

*Electronic address: jayanti@iucaa.ernet.in
†Electronic address: tarun@iucaa.ernet.in

simple implementation of the Particle Swarm Optimization in detail. In particular, we focus on the dynamics of particles, setting-up initial conditions, boundary conditions and the convergence criteria. We also define all the design parameters and variables of PSO in § II. A very brief overview of parameter estimation in Bayesian formalism is discussed in §III. In place of discussing the full prescription of Maximum Likelihood (ML) estimation, we mainly focus on the computation of error bars from covariance or Hessian matrix in this section. We present our results of parameter estimation using PSO for WMAP seven year data in § IV. Apart from giving the best fit parameter estimates which we get from PSO, we also present a comparison of the results obtained from PSO and MCMC. In §V we summarize the PSO with discussing its advantages and disadvantages.

II. PARTICLE SWARM OPTIMIZATION

Formally proposed by James Kennedy and Russell Eberhart in 1995 [12] PSO has been successfully tested and applied in many engineering and artificial intelligence problems [17–19]. Recently it has been applied in astrophysical problems also [16, 20, 21]. In PSO, a set of "particles" driven by "cognition" and "social" factors explore the multi-dimensional search space by carrying out random walks, determined by a set of "design parameters". The accuracy and performance of the algorithm depends on the values of the design parameters as well as the function to be optimized i.e., the fitness function.

Particle Swarm Optimization is based on observations of social dynamics, bird flocks, fish schools and other forms of group behavior. Personal discoveries made by members of the group and shared with everyone else can help everyone to become more efficient in making new discoveries. Efficient personal search and communication with other members of the group can lead to rapid success for the group in search of some common goal i.e., food etc.

At present there exists many implementations of the Particle Swarm Optimization. Here we consider one of the simplest one[12] the elements of which are shared by many other implementations. Before describing the working of our PSO implementation, it is useful to define some of the key terms which are used to describe PSO.

1. **Particles:-** The term "particles" in PSO is used for "computational agents" and has no relation with any form of physical particles. PSO particles neither have any mass nor occupy any volume (however, they can have weights called inertia weights which will be discussed later). PSO particles are distinguished from each other on the basis of their identification numbers or *id*s and have "positions" and "velocities". In our discussion we represent the position and velocity of a particle with id i at step ("time") t by vector $X[i,t]$ and $V[i,t]$ respectively.

2. **Fitness function:-** The function to be used for searching the global maximum is called the "fitness function" or "optimization function". In the present case we use $-2\log\mathcal{L}$ or $\chi^2_{\rm eff}$ as our optimization function, where $\mathcal{L}$ is the CMBR likelihood function.

3. `Pval` :- We represent the value of the optimization function for a particle i at time t by its `Pval` i.e., `Pval` $[i,t] = \mathcal{F}(X[i,t])$.

4. `Pbest` **:-** The largest value of `Pval` a particle i has obtained till the present step (time (t)) is called its `Pbest` .

$$\texttt{Pbest}\ [i] = \text{Max}\{\texttt{Pval}\ [i,t'], t' = 0,1,2...,t\} \quad (1)$$

the location of the `Pbest` is represented by the vector $P[i]$

$$X[i,t] = P[i] \ \text{ if } \ \mathcal{F}(X[i,t]) = \texttt{Pbest} \quad (2)$$

5. `Gbest` **:-** The largest value of `Pbest` among all particles is called the `Gbest` . The value of `Gbest` changes only when any of the particles find a new position at which the value of the fitness function is larger than any of the earlier values.

$$\texttt{Gbest}\ = \text{Max}\{\texttt{Pbest}\ [i], i = 0,1,2...,N_{part}\} \quad (3)$$

where N_{part} is the number of particles. The location of the `Gbest` is given by the vector G.

$$X[i,t] = G \ \text{ if } \ \texttt{Pbest}\ [i] = \texttt{Gbest}\ . \quad (4)$$

A. Dynamics of PSO particles

The following equation is used to update the positions of the particles [12],

$$X[i,t+1] = X[i,t] + V[i,t+1], \quad (5)$$

where velocity $V[i,t+1]$ for the particle i at step $(t+1)$ is computed in the following way

$$V[i,t+1] = wV[i,t] + c_1\xi_1\left\{X[i,t] - P[i]\right\} + c_2\xi_2\left\{X[i,t] - G\right\}, \quad (6)$$

where c_1 and c_2 are called acceleration coefficients, w is called the inertia weight and ξ_1 and ξ_2 are two uniform random numbers in the range $[0,1]$. The values of the acceleration coefficients c_1 and c_2 decide the contribution due to personal (cognitive) learning and social learning respectively.

The first factor in the right hand side of equation (6) moves the particle along a straight line and the second and third factors accelerate it towards the location of the `Pbest` and `Gbest` respectively. Kennedy and Eberhart [12] use $c_1 = c_2 = 2$ to give it a mean of unity, so that the particle would overfly the target about half of the time.

Although the equation (6) is most commonly used, however, the following version is also in use [20].

$$V[i,t+1] = K\{V[i,t] + C_1\xi_1\left(X[i,t] - P[i]\right) + C_2\xi_2\left(X[i,t] - G\right)\} \quad (7)$$

where K is called the constriction factor and is defined in the following way:

$$K = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (8)$$

where $\phi = C_1 + C_2$. Here the recommended values are $C_1 = C_2 = 2.05$ which gives $K = 0.729$. Equation (7) is equivalent to equation (6) with $c_1 = KC_1, c_2 = KC_2$ and $w = K$. Since there are many implementations (with different values of design parameters or with some new parameters) we have decided to work with PSO standard 2006 [22] which uses the following values of the design parameters in equation (6)

$$w = \frac{1}{2\log(2)} = 0.72 \quad (9)$$

and

$$c_1 = c_2 = 0.5 + \log(2) = 1.193 \quad (10)$$

since we were able to get quite accurate results with the values of design parameters suggested in PSO standard 2006[22], we decided to adopt these values in our implementation. We did try a few other values but could not find any significant improvement.

In particle swarm optimization all the particles can communicate with each other or the communication can be restricted between only subsets of particles. The first case is found to be more useful for intensive local search and the second one for global search. In our implementation we let every particle share the information about its `Pbest` with every other particle.

B. Maximum Velocity

In order to stop particles leaving the search space we need to limit the maximum velocity which particles can acquire. This can be done by setting the maximum velocity along various dimensions. It has been a common practice to keep the maximum velocity proportional to the search range.

$$V[i,t] = \begin{cases} V_{max}, & \text{if } V[i,t] > V_{max} \\ -V_{max}, & \text{if } V[i,t] < -V_{max} \end{cases} \quad (11)$$

where V_{max} is also a design parameter. We use $V_{max} = c_v(X_{max} - X_{min})$ with $c_v = 0.5$ where $[X_{min}, X_{max}]$ is our search range. This means that the biggest jump a particle can make is half of the size of the search range.

C. Initial Conditions

We assign random positions and velocities to particles in the beginning.

$$X[i, t=0] = X_{min} + \xi \times (X_{max} - X_{min}) \quad (12)$$

and

$$V[i, t=0] = \xi V_{max}, \quad (13)$$

where ξ is a uniform random number in the range $[0-1]$. Apart from the above initial conditions "particles on a grid" initial condition can also be used. From our trial runs we have found that the final outcome i.e., the location of the `Gbest` is not much sensitive to the initial condition.

D. Boundary condition

We use the "reflecting wall" boundary condition in which a particle reverses its velocity component perpendicular to the boundary when it tries to cross the boundary.

$$V[i,t] = -V[i,t] \quad (14)$$

and

$$\begin{cases} X[i,t] = X_{max} & \text{when } X[i,t] > X_{max} \\ X[i,t] = X_{min} & \text{when } X[i,t] < X_{min} \end{cases} \quad (15)$$

E. Termination criteria

In the early steps the PSO particles make large jumps and so there is a large change in the location of the `Gbest` at every step, however, when the location of the `Gbest` approaches close to the global maximum, particles move little. After a certain number of PSO steps the change in the location of the `Gbest` is very small (see Figure (3)) and we can stop the exploration if the value of the change in `Gbest` remains smaller than a small number (ϵ) for a given number of steps (N_{stops}). In the present case we use many different values for trial, however, we found that any set of values of ϵ and N_{stops} which allows between 200 and 300 PSO steps for 30 particles gives satisfactory results.

III. COSMOLOGICAL PARAMETER ESTIMATION

Cosmic Microwave Background Radiation (CMBR) temperature and polarization anisotropies observed in the sky, represent the fluctuations in the baryons-photons fluid at the epoch of last scattering i.e., when electrons were last time scattered by photons before they combined

Parameter	Description	Value used
w	Inertia weight	0.72
c_1	Acceleration parameters (personal)	1.193
c_1	Acceleration parameters (social)	1.193
$c_v = V_{max}/\Delta X$	maximum velocity parameter	0.5
N_{part}	Number of particles	30
N_{dim}	Search dimensions	6
N_{stops}	The number of steps for stopping	
ϵ	Convergence parameter	

TABLE I: PSO design parameters

with protons and formed hydrogen atoms, contain a lot of information about the cosmological parameters [7, 23–25]. Due to the Gaussian nature of the density fluctuations at the epoch of last scattering (primordial fluctuations) as predicted by inflationary models, the most important information about the cosmological parameters is encoded in the angular two point correlation function or power spectrum.

It is a common practice to represent the temperature anisotropies in the CMBR sky in spherical harmonic expansion

$$\frac{\Delta T(\hat{n})}{T_0} = \sum_{lm} a_{lm} Y_{lm}(\hat{n}) \tag{16}$$

where T_0 is the average temperate i.e., the monopole term. The angular two point correlation function is given by

$$C(\theta) = \left\langle \frac{\Delta T(\hat{n})}{T_0} \frac{\Delta T(\hat{n}')}{T_0} \right\rangle = \sum_l \frac{2l+1}{4\pi} C_l P_l \cos\theta \tag{17}$$

where θ is angle between directions $\hat{n}$ and $\hat{n}'$ in the sky. The angular power spectrum C_l is defined as

$$C_l = \langle a_{lm} a^*_{lm} \rangle = \left\langle |a_{lm}|^2 \right\rangle . \tag{18}$$

As mentioned above, the angular power spectrum C_l (or angular two point correlation function $C(\theta)$) depends on a large number of cosmological parameters representing various energy densities in the universe, primordial fluctuations, and the physical process relevant in the early universe like reionization and recombination [23, 26–28]. Many of the cosmological parameters affect the angular power spectrum in the same way, i.e., degeneracies. However, it is possible to form a set of parameters, called "normal parameters" which affect the angular power spectrum in a unique way [27, 29]. The most common cosmological parameters which have been used (we also use these) to fit observational C_l are density parameters for cold dark matter ($\Omega_c h^2$), baryons ($\Omega_b H^2$), and cosmological constant (Ω_Λ), primordial scalar power spectrum index (n_s) and normalization (A_s), and reionization optical depth (τ). In our analysis we do not consider the "Hubble parameter" h as a free parameter and compute its value from other parameters for a spatially flat model.

$$h = \sqrt{\frac{\Omega_b h^2 + \Omega_c h^2}{1 - \Omega_\Lambda}} \tag{19}$$

It is now a common practice to follow a " line of sight " integration approach for computing the angular power spectrum C_l for a set of cosmological parameters which is a computationally expensive process. Publicly available code CAMB [30, 31] based on an earlier code named CMBFAST [32] which can compute the angular power spectrum C_l on a shared memory platform in a short time.

A. Bayesian Analysis

In the framework of Bayesian analysis the probability of obtaining a set of parameters Θ which is consistent with a data set D for a given prior I is given by

$$P(\Theta|D, I) = \frac{P(D|\Theta, I) P(\Theta|I)}{P(D|I)}. \tag{20}$$

In the above equation $P(\Theta|D, I)$ is the posterior probability distribution, $P(D|\Theta, I)$ is the likelihood function (will be represented by $\mathcal{L}$) and $P(\Theta|I)$ is the prior. The denominator $P(D|I)$ called "evidence" is used for normalization purpose and does not depend on the parameters Θ so can be ignored for the present purpose.

The likelihood function for a CMBR experiment with N_p pixel is given by [7]

$$\mathcal{L}(\Delta|\Theta) = \frac{1}{(2\pi)^{N_p/2}} \frac{1}{|C|^{N_p/2}} \exp\left[-\frac{1}{2} \Delta C^{-1} \Delta \right] \tag{21}$$

where Δ represents an estimator of the observed data vector, having N_p entries, and C is the joint covariance matrix i.e., sum of the signal and noise covariance matrix

$$C = S + N. \tag{22}$$

The noise covariance matrix N can be approximated by a diagonal matrix and the signal covariance matrix S is given by [6]

$$S_{ij} = \sum_l \frac{2l+1}{4\pi} C_l P_l(\cos\theta) \tag{23}$$

where θ is the angle between the directions $\hat{n}_i$ and $\hat{n}_j$ representing pixel i and pixel j respectively.

Exact likelihood computation by a brute force method is computationally expensive since it involves inversion of $N_p \times N_p$ matrix which is a formidable task for an experiment with very large number of pixels. There have been proposed many approximations which reduce the cost of likelihood computation significantly [6, 33, 34].

In the present work we use the likelihood code provided by the WMAP team for computing the likelihood function which takes the theoretical angular power spectrum computed by CAMB, and the power spectrum estimated by the WMAP experiment[35], as inputs.

The exercise of obtaining the best fit cosmological parameters involves finding a point in the multidimensional parameter space, at which the value of the likelihood function $\mathcal{L}$ is maximum or $-2\log\mathcal{L}$ is minimum. Apart from the best fit values one is generally interested in error bars also on the estimated parameters which involves knowing the shape of the likelihood function around the best fit values for which we follow a fitting procedure as discussed below.

B. Likelihood fitting

It has been argued that one can construct a set of parameters, called normal parameters, from the cosmological parameters in which the the likelihood function is roughly a multivariate Gaussian around the best fit values θ_0, and so, $\log\mathcal{L}$ can be well approximated by a quadratic function [36].

$$\mathcal{L}(\theta) = \mathcal{L}(\theta_0)\exp(-\sum_{ij} F_{ij}\delta\theta_i\delta\theta_j/2) \tag{24}$$

where Fisher matrix F is given by [27]

$$F_{ij} = \sum_l (\Delta C_l)^2 \frac{\partial C_l}{\partial\theta_i}\frac{\partial C_l}{\partial\theta_j} \tag{25}$$

where

$$\Delta C_l = \sqrt{\frac{2}{(2l+1)f_{sky}}}\left[C_l + w^{-1}e^{-l(l+1)/l_s^2}\right] \tag{26}$$

where w^{-1} is the noise power for each channel, f_{sky} is the fraction of the sky observed and l_s is the Gaussian beam filter scale.

Identifying $-2\log\mathcal{L}$ with $\chi^2_{\rm eff}$, equation (24) can be written as

$$\Delta\chi^2_{\rm eff} = -2\log\mathcal{L} + 2\log\mathcal{L}_0 = \sum_{ij} F_{ij}\delta\theta_i\delta\theta_j \tag{27}$$

Error (1σ) on parameter θ_i is computed from the covariance matrix M

$$\Delta\theta_i = \sqrt{M_{ii}} \quad \text{where} \quad M = F^{-1} \tag{28}$$

IV. RESULTS

We compute the best fit cosmological parameters from the WMAP seven years data for a six parameter model with model parameters $\Omega_b h^2, \Omega_c h^2, \Omega_\Lambda, n_s, A_s$ and τ using PSO. Before presenting our results quantitatively, we consider it useful to present a qualitative comparison of the way parameters are estimated in the Markov-Chain Monte Carlo methods and in Particle Swarm Optimization. In particular, we want to highlight the way parameter space is explored and sampled in PSO and MCMC methods.

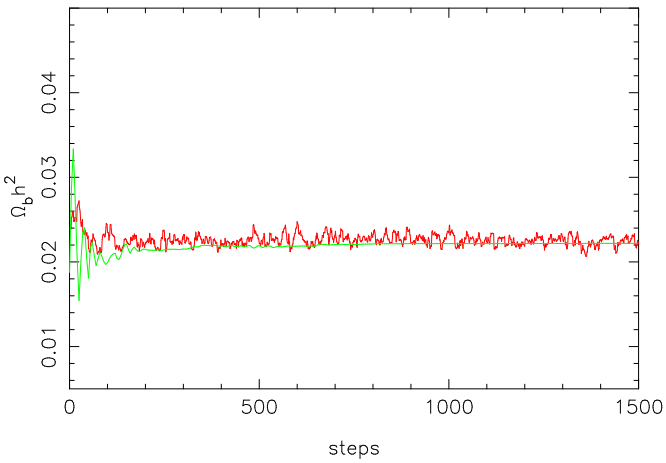

FIG. 1: In this figure the red line shows a Markov chain which has been obtained from a typical run of COSMOMC and the green line shows the trajectory of a PSO particle, along the same dimension i.e. $\Omega_b h^2$. The "burn-in" phase is also shown to highlight the initial behavior of the PSO particles exploration and that of the Markov chain. From the figure it can be noticed that the Markov chain starts from a far away location and then progressively moves towards the best fit location. In the case of PSO also the particle starts from a far away point and then moves towards the best fit location. However, in the case of PSO the particle approaches towards the best fit location (`Gbest`) in an oscillatory manor with successively decreasing amplitude. Only after a sufficient number of PSO steps the particle positions and the Markov chain converge. Since there are more number of points for the Markov chain as compared to the PSO, we use x-scale such that we have five Markov points for very PSO point.

A. Markov Chain and PSO exploration

The nature of exploration by a Markov chain and that by a set of PSO particles is completely different. In the first case two successive steps are correlated [3] which does not happen in the second case. However, there are some similarities also, for example, in both the cases the random walk is governed by the optimization function or the fitness function, in MCMC proposal density is directly related to the function to be sampled. In the case of MCMC exploration of a chain is completely local, in the sense that whether a step will be selected or rejected depends only on the values of the fitness function at the current location and the next location. However, in the case of PSO, particles always have some informa-

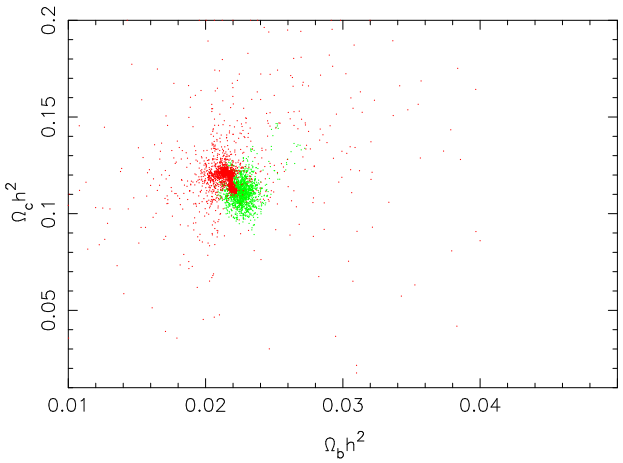

FIG. 2: In this figure the red and the green points show the distribution of the positions of PSO particles and samples from a Markov chain respectively, in the same plane. From the figure it can be noticed that in the initial stage the scatter of PSO particles is very large (see Figure 1 also), however, close to the convergence all particles get confined in a very compact region. The distribution of the sample points in the case of Markov chain is much more symmetric than in PSO. We suspect that this is due to different role played by the stochastic variables (random numbers) in PSO as compared to that in the Markov chains. Non-symmetric distribution makes PSO less favorable if we want to find the shape of the likelihood close to the best fit values (in order to report errors) in comparison to Markov Chain.

tion about the global maximum `Gbest` , which keeps changing. In general the step size does not change in MCMC, however, in the case of PSO it rapidly falls as `Gbest` approaches close to the global maximum.

Markov chains sample the fitness function in the multi-dimensional parameter space using methods like Metropolis Hastings. The sampling algorithm ensures that more points are sampled from regions in which the fitness function has large values and less points from the regions in which it has small values. The values of best fit parameters are obtained after marginalization. In case of PSO a set of "particles" explore the multidimensional space guided by their personal i.e., `Pbest` and collective i.e., `Gbest` discoveries (see equation (5) and equation (6)). The progress of a chain in MCMC and trajectory of a particle in PSO are very different. Starting from any arbitrarily point in the multi-dimensional space both approach towards the region where the probability of global maximum is high, however, the way they approach is very different.

Figure 1 shows a typical Markov chain and the trajectory of a PSO particle. Since there were greater number of steps in the Markov chain than in PSO, we have stretched the x-axis for PSO trajectory by a factor of five, i.e, there are five Markov chain points for every PSO point for the same range on the x-axis. From the figure it can be noticed that the PSO particle reach the global maximum by performing oscillatory motion with gradually decreasing amplitude. However, in the case of Markov chain the progress is very smooth.

One of the most common ways to present the results of a parameter estimation exercise is to make two dimensional scatter or contour plots. In MCMC it is done by marginalizing the sampled function along all other dimensions apart from the two for which we want scatter or contour plots. For a general case the location of the point at which the likelihood function peaks, may be different from the average location computed on the basis of the one dimensional probability probability distribution obtained after marginalization. In the case of PSO also, we present the average location also of the particles, apart from finding the point at which likelihood function peaks.

The red and green points in the Figure 2 show the projection of PSO particles position and a set of sample points from a Markov chain in a two-dimensional plane of the parameter space, respectively. From the figure it can be noticed that although the sampled points in both the cases cluster around the same point (`Gbest` in PSO) the distribution are completely different. In particular the points are more symmetrically distributed around the best fit value in the Markov chain case as compared to that in PSO. Since PSO results are always quoted in term of `Gbest` therefore the distribution of points does not change the results in any way. However, since in the present work we make explicit use of PSO particle distribution also (for fitting the likelihood function for computing errors), it can create some problem.

B. Best fit cosmological parameters

In order to test the working of our PSO module we considered a six dimensional cosmological model $(\Omega_b h^2, \Omega_c h^2, \Omega_\Lambda, n_s, A_s, \tau)$ (see Figures 1 of [9]) and tried to estimate its parameters from the WMAP seven years data. In order to be consistent with [9] we use a constant (0.52) value for the parameter A_{sz}. The range over which we tried to optimize our fitness function and the results are given in Table (II). In the Figure 3 we show the evolution of the fitness function $(-2\log\mathcal{L})$ with PSO steps. We also show the value for the fitness function for WMAP seven years best fit cosmological parameters (dashed line). From the figure it can be noticed that the value obtained by PSO finally converges to the WMAP seven year value. Since velocity with which particles move toward `Gbest` is proportional to their distance from the `Gbest` , we get large jumps in the fitness function in the beginning.

In PSO the values of the best parameters, location in the parameter space at which the likelihood function peaks, is represented by `Gbest` . For consistency and robustness we not only give the location of the `Gbest` , we also give the average location of the PSO particles. It is not a surprise that as PSO progresses, the average

Cosmological parameters from PSO							
Variable		PSO best fit			WMAP best fit[9]		Difference (Gbest -ML)
	Range	Gbest	Mean	std dev.	ML	Mean	
$\Omega_b h^2$	(0.01,0.04)	0.02219	0.02219	0.00049	0.02227	$0.02249^{0.00056}_{-0.00057}$	-0.0001 (0.36 %)
$\Omega_c h^2$	(0.01,0.20)	0.1109	0.1109	0.0058	0.1116	0.1120 ± 0.0056	- 0.0007(0.63%)
Ω_Λ	(0.50,0.75)	0.731	0.731	0.034	0.729	$0.727^{+0.030}_{-0.029}$	+0.0020 (0.27%)
n_s	(0.50,1.50)	0.962	0.962	0.012	0.966	0.967 ± 0.014	-0.0040(0.41%)
$A_s/10^{-9}$	(1.0,4.0)	2.43	2.43	0.13	2.42	2.43 ± 0.11	0.01(0.41%)
τ	(0.01,0.11)	0.084	0.084	0.012	0.0865	0.088 ± 0.015	-0.0021(2.43%)

TABLE II: The first columns in the above table shows the PSO fitting parameters and the second, third, fourth and fifth column show the search range, the location of the Gbest , the average position of PSO particles and the error (which computed by fitting the sampled function) respectively. In the sixth and seventh columns we give the best fit (ML) and the average values of the cosmological parameters derived from WMAP seven years likelihood estimation respectively. In the last column we give difference between our best fit parameters (PSO parameters) and WMAP team's best fit parameters (difference between ML and Gbest values). From this table it is clear that roughly there is good agreement between the PSO best fit parameters and WMAP team's best fit parameters from the seven years data.

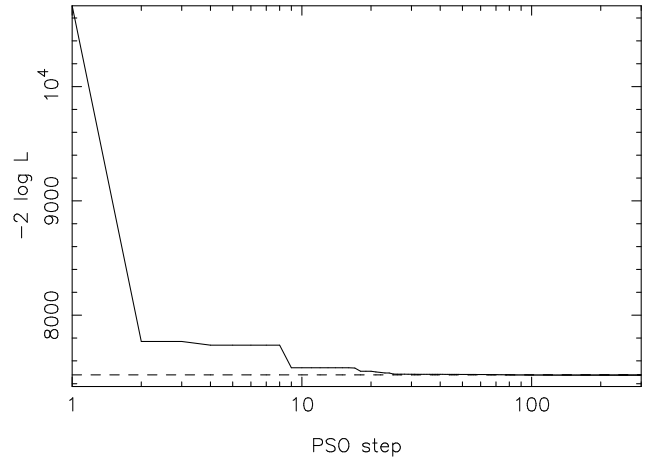

FIG. 3: The solid line in the above figure shows the change in the fitness function $-2\log\mathcal{L}$ as PSO steps progress, and the dashed line shows the value for the WMAP seven years data. From this figure it can be noticed that in the beginning improvement in the value of the fitness function is quite rapid, however, after some time it saturates, primarily because once particles reach close to the global maximum as given by the Gbest their velocities drop.

position of PSO particles and the location of the best point converge i.e., Gbest . In a case when there are local maxima also present, some of the PSO particles may trap in these, however, the average location of particles still follows the Gbest . In Figure 4 we show the location of the Gbest and the average position of the PSO particles in our six dimensional search space at at different steps. Note that in our model h is not a fitting parameter, we get its value from the flatness condition (see equation (19)).

The black, red and blue lines in Figure 5 show the best fit angular power spectrum obtained by MCMC analysis, from PSO code and the binned power spectrum (with error bars) provided by the WMAP team for the seven year data. From the figure it is clear that the power spectrum which we obtain from our PSO code closely follows other two curves.

C. Error estimates

In order to compute error bars on the parameters estimated using PSO, we fit (as discussed in § III B) a six dimensional paraboloid to a subset of sampled points to $\Delta\chi^2_{\rm eff} = -2\log(\mathcal{L}-\mathcal{L}_0)$ where $\mathcal{L}_0$ is the value of likelihood function or Gbest at the last PSO step.

$$\Delta\chi^2_{\rm eff} = [\tilde{\Theta}][\alpha][\tilde{\Theta}]^T \tag{29}$$

where $\Theta = (\Omega_b h^2, \Omega_c h^2, \Omega_\Lambda, n_s, A_s, \tau)$ and $[\alpha]$ is a 6×6 symmetric matrix with 21 independent coefficients.

The six dimensional vector $\tilde{\Theta}$ is defined as

$$\tilde{\Theta} = \frac{\Theta - \Theta_{\tt Gbest}}{\Theta_{\tt Gbest}}. \tag{30}$$

We used multi-parameter fitting subroutine of GNU scientific library for the fitting [37]. In order to limit our fitting to a subset of points, we consider only those points which are within a six dimensional hyper-sphere of radius i.e., $|\tilde{\theta}_i| < 0.1$ where θ_i is a component of the vector $\tilde{\Theta}$, and $\Delta\chi^2 < 10$.

After obtaining the fitting matrix $[\alpha]$ we invert it and compute the covariance matrix $[C] = [\alpha]^{-1}$ and compute error for the parameter θ_i from that

$$\Delta\theta_i = \sqrt{C_{ii}} \times \theta_{i,\tt Gbest} \ . \tag{31}$$

We present the error in various parameters in the fifth column of Table (II).

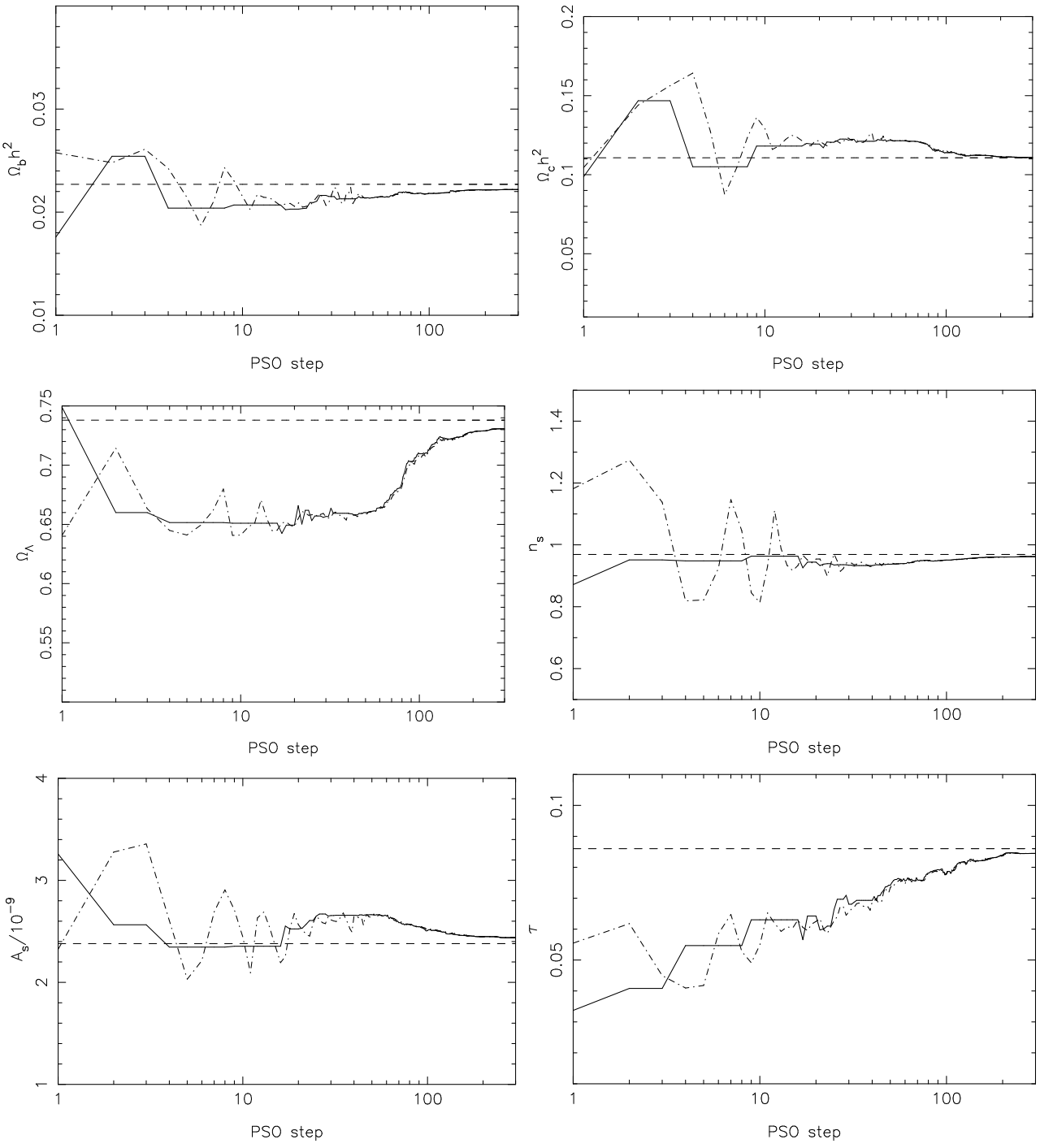

FIG. 4: In every panel of the above figure we show the location of the best fit points (location of `Gbest`) and the average location of PSO particles, by the solid and dot-dashed lines respectively. We also show the best fit values given by the WMAP team by dashed line. From the above figure it can be noticed that as PSO progresses the average location of PSO particles and the location of the `Gbest` converge, which can be used as a robust check. For most cases the best fit values obtained by PSO match well with the values computed by COSMOMC, however, there are some differences also.

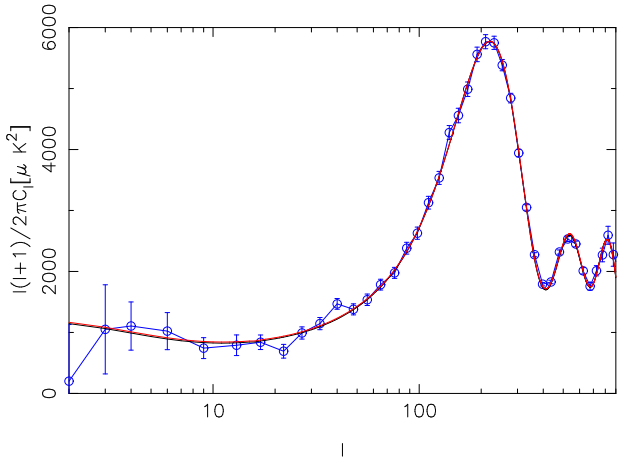

FIG. 5: The red, black and blue lines in the above figure represent the best fit angular power spectrum recovered from PSO, COSMOMC and the binned power spectrum of WMAP seven year data respectively. Note that the PSO best fit angular power spectrum is very close to that provided by the WMAP team. The small difference (see table (II)) between the PSO best fit parameters and the WMAP best fit parameters leads to a difference of 2 in $\chi^2_{\rm eff}$ (it is smaller by 2 for PSO).

D. Computational performance

Computing the fitness function i.e., $-2\log\ \mathcal{L}$, which is the most expensive part in the parameter estimation procedure, involves two steps. In the first step C_ls are computed for a set of cosmological parameters for which we use publicly available code CAMB [31] which employs OpenMP pragmas for doing computationally intensive steps in parallel on multi-processor shared memory systems. In the second step the likelihood function is computed from C_ls and the observational data i.e., WMAP seven years data, for which we use the likelihood code provided by the WMAP team. Since our PSO code shares two main modules (C_l and likelihood computation) with publicly available code COSMOMC, difference in the performance is expected only due to the number of times the fitness function is computed. Computationally a typical PSO run gives very good convergence with 30 PSO particles with 300 steps i.e., 9000 computations.

Here it is also important to mention that parallelizing our code is very straightforward. We use OpenMP to compute C_l for a point in the six dimensional parameter space and use MPI to distribute particles among different MPI nodes. At every step, particles are distributed among MPI nodes and after they return the value of the fitness function, the master node computes the `Pbest` and `Gbest` and update the positions and velocities of particles. We have tested our code on a Linux cluster using 15 MPI nodes, where each node has 2 AMD quad core Opetron 2.6 GHz processors. Within a node we use OpenMP for computing the angular power spectrum using as many number of threads as the number of cores are present. We have also ported our code on a Cray CX1 system with six nodes, where each node has 2 Intel Xeon 2.67 Ghz 6 core processors. In a case when the number of nodes can divide the number of PSO particles, there is no difficulty with load balance. Since at every PSO step very small amount of data has to be communicated among MPI nodes, we have found that MPI collective communications calls like "broadcast" and "gather" are much more efficient than regular "send" and "receive" calls.

It is not very straightforward to compare the performance of our PSO code and that of commonly used code COSMOMC mainly because (1) the convergence criteria in PSO is very different from in COSMOMC (2) angular power spectrum C_l computation in COSMOMC is optimized by selecting only a subset of particles which change their value, however, there is no such operation in PSO step (3) It is a common practice to "thin" chains in MCMC which means that not all sampled points are used for the final result which is not the case in PSO. In [6] four chains with each having 30,000 points are needed for convergence, however, in our typical run with six parameter model we never need more than 8-9000 computations. It is also important that the convergence in PSO also depend on value of design parameters.

V. DISCUSSION AND CONCLUSION

In the present work we have demonstrated application of the Particles Swarm Optimization for cosmological parameter estimation from CMBR data, which we believe has not been done earlier in any other study. Being a different method, PSO can be used an alternative method for parameter estimation, particularly when one is mainly interested in the location of the best fit point. The main focus of our present work was to demonstrate the method, in place of producing quantitative results and comparing those with publicly available code COSMOMC, which we leave for our future work. We have not only shown how to compute the values of the best-fit parameters, we have also proposed a method to quantify the error bars on the estimated values.

Based on a very simple algorithm, PSO has many interesting features some of them are as follows:

1. PSO has very few design parameters and the values of which can be easily fixed.

2. By tuning the values of the design parameters, PSO can be made more efficient for the global or the local search although it is more useful for the global search.

3. In PSO no approximation or extrapolation is made at any step., (like in artificial neural network) and the optimization function is computed exactly at every point.

4. As is claimed in other studies also that PSO is very efficient in searching global maximum when dimensionality of the search space is very high or there are a large number of local maxima present. Adding extra search dimensions in PSO is quite straightforward.

5. PSO can be used for accelerated search of the global maximum since it always has some idea about the `Gbest` from very beginning.

As a result of very high quality CMBR data which has already been provided by the WMAP satellite [35] and will be provided by the Planck satellite [38], it has become an interesting exercise to consider much more complex models (with very high dimensionality than just six to eleven dimensional models). The main motivation behind considering such models has been to fit the "outliers" of the WMAP data (all years). In one of such exercise, in place of considering the primordial power spectrum just a power law, power in various bins can be left open for the fitting, which makes the dimensionality of the search space very large and we believe that PSO can be quite useful for such problems.

[1] G. Jungman, M. Kamionkowski, A. Kosowsky, and D. N. Spergel, Phys. Rev. D **54**, 1332 (1996), arXiv:astro-ph/9512139.

[2] G. Jungman, M. Kamionkowski, A. Kosowsky, and D. N. Spergel, Physical Review Letters **76**, 1007 (1996), arXiv:astro-ph/9507080.

[3] A. Lewis and S. Bridle, Phys. Rev. D **66**, 103511 (2002), arXiv:astro-ph/0205436.

[4] A. Kosowsky, M. Milosavljevic, and R. Jimenez, Phys. Rev. D **66**, 063007 (2002), arXiv:astro-ph/0206014.

[5] L. Verde *et al.*, Astrophys. J. Suppl. Ser. **148**, 195 (2003), arXiv:astro-ph/0302218.

[6] L. Verde, Statistical Methods in Cosmology, in *Lecture Notes in Physics, Berlin Springer Verlag*, edited by G. Wolschin, volume 800 of *Lecture Notes in Physics, Berlin Springer Verlag*, pp. 147–177, 2010, 0911.3105.

[7] S. Dodelson, *Modern cosmology* (Academic Press, San Diego, U.S.A., 2003).

[8] N. Jarosik *et al.*, Astrophys. J. Suppl. Ser. **192**, 14 (2011), 1001.4744.

[9] D. Larson *et al.*, Astrophys. J. Suppl. Ser. **192**, 16 (2011), 1001.4635.

[10] N. Christensen, R. Meyer, L. Knox, and B. Luey, Classical and Quantum Gravity **18**, 2677 (2001), arXiv:astro-ph/0103134.

[11] T. Auld, M. Bridges, M. P. Hobson, and S. F. Gull, Mon. Not. R. Astron. Soc. **376**, L11 (2007), arXiv:astro-ph/0608174.

[12] J. Kennedy and R. C. Eberhart, IEEE International Conference on Neural Networks **4**, 1992 (1995).

[13] J. Kennedy and R. C. Eberhart, *Swarm Intelligence* (Morgan Kufmann, 2001).

[14] A. P. Engelbrecht, *Computational Intelligence, An Introduction* (John Wiley & Son, 2002).

[15] http://lambda.gsfc.nasa.gov/.

[16] Y. Wang and S. D. Mohanty, Phys. Rev. D **81**, 063002 (2010), 1001.0923.

[17] A. Lazinica, *Particle Swarm Optimization* (In-Tech., 2009).

[18] D. W. Boeringer and D. H. Werne, IEEE Transc. on Antennas and propagation **52**, 771 (2004).

[19] J. Robinson and Y. Rahmat-Samii, IEEE Trans. on Ant. and Prop. **727**, 397 (2004).

[20] C. Skokos, K. E. Parsopoulos, P. A. Patsis, and M. N. Vrahatis, Mon. Not. R. Astron. Soc. **359**, 251 (2005), arXiv:astro-ph/0502164.

[21] A. Rogers and J. D. Fiege, Astrophys. J. **727**, 80 (2011), 1101.5803.

[22] www.particleswarm.info/Standard_PSO_2006.c/.

[23] J. R. Bond and G. Efstathiou, Astrophys. J. Lett. **285**, L45 (1984).

[24] A. Challinor, arXiv:astro-ph/0403344 (2004), arXiv:astro-ph/0403344.

[25] W. Hu, Concepts in CMB Anisotropy Formation, in *The Universe at High-z, Large-Scale Structure and the Cosmic Microwave Background*, edited by E. Martinez-Gonzalez & J. L. Sanz, volume 470 of *Lecture Notes in Physics, Berlin Springer Verlag*, pp. 207–+, 1996, arXiv:astro-ph/9511130.

[26] J. R. Bond and G. Efstathiou, Mon. Not. R. Astron. Soc. **226**, 655 (1987).

[27] J. R. Bond, G. Efstathiou, and M. Tegmark, Mon. Not. R. Astron. Soc. **291**, L33 (1997), arXiv:astro-ph/9702100.

[28] J. R. Bond, A. H. Jaffe, and L. Knox, Phys. Rev. D **57**, 2117 (1998), arXiv:astro-ph/9708203.

[29] M. Chu, M. Kaplinghat, and L. Knox, Astrophys. J. **596**, 725 (2003), arXiv:astro-ph/0212466.

[30] http://camb.info/.

[31] A. Lewis, A. Challinor, and A. Lasenby, Astrophys. J. **538**, 473 (2000), arXiv:astro-ph/9911177.

[32] U. Seljak and M. Zaldarriaga, Astrophys. J. **469**, 437 (1996), arXiv:astro-ph/9603033.

[33] S. Hamimeche and A. Lewis, Phys. Rev. D **77**, 103013 (2008), 0801.0554.

[34] S. Hamimeche and A. Lewis, Phys. Rev. D **79**, 083012 (2009), 0902.0674.

[35] http://map.gsfc.nasa.gov/.

[36] H. B. Sandvik, M. Tegmark, X. Wang, and M. Zaldarriaga, Phys. Rev. D **69**, 063005 (2004), arXiv:astro-ph/0311544.

[37] http://www.gnu.org/software/gsl/.

[38] http://www.nasa.gov/mission_pages/planck/.