

# Revisiting Cosmological parameter estimation

Jayanti Prasad\*

IUCAA, Post Bag 4, Ganeshkhind, Pune 411007, India.

(Dated: December 11, 2014)

Constraining theoretical models with measuring the parameters of those from cosmic microwave background (CMB) anisotropy data is one of the most active areas in cosmology. WMAP, Planck and other recent experiments have shown that the six parameters standard  $\Lambda$ CDM cosmological model still best fits the data. Bayesian methods based on Markov-Chain Monte Carlo (MCMC) sampling have been playing leading role in parameter estimation from CMB data. In one of the recent studies [1] we have shown that particle swarm optimization (PSO) which is a population based search procedure can also be effectively used to find the cosmological parameters which are best fit to the WMAP seven year data. In the present work we show that PSO not only can find the best-fit point, it can also sample the parameter space quite effectively, to the extent that we can use the same analysis pipeline to process PSO sampled points which is used to process the points sampled by Markov Chains, and get consistent results. We also present implementations of downhill-simplex Method of Nelder and Mead and Powell's method of Bound Optimization BY Quadratic Approximation (BOBYQA) in this work for cosmological parameter estimation, and compare these methods with PSO. Since PSO has the advantage that it only needs the search range and does not need covariance-matrix, starting point or any other quantity which depend on the final results, it can be quite useful for a blind search of the best fit parameters. Apart from that, PSO is based on a completely different algorithm so it can supplement MCMC methods. We use PSO to estimate parameters from the WMAP nine year and Planck data and get consistent results.

PACS numbers:

## I. INTRODUCTION

Cosmological parameters characterizing the primordial curvature and tensor perturbations which were present at the end of inflation, background expansion of the universe, physical events (like reionization) that took place after the decoupling of CMB photons can be estimated from the temperature and polarization anisotropies present in the CMB sky [2–12]. CMB anisotropies are represented by a stochastic field on a 2-sphere and can be completely characterized by the correlations between different directions. If CMB anisotropies are statistically isotropic and Gaussian also, as most observations indicate, then they can be completely described by two-point correlations or the angular power spectrum [3, 8, 9, 13–17]. Since CMB anisotropies are quite small (one in one hundred thousand part for temperature) their evolution can be studied quite effectively using the theory of linear perturbations [3, 18–20]. There are numerical codes available which give us the angular power spectrum of CMB anisotropies at present for a given model of primordial fluctuations and background cosmology [21–23] and that can be compared with the observed angular power spectrum we get from CMB experiments like WMAP and Planck.

In general cosmological parameter estimation from CMB data involves finding the point in the multi-dimensional parameters space at which the likelihood function is maximum i.e., the best fit point. In Bayesian

framework this is done by sampling the joint posterior probability distribution and from that various statistical quantities are computed. Since grid based sampling is prohibitively expansive (computationally) for a large number of parameters (at least six in our case), stochastic methods are generally employed which scales linearly (at the most) with the number of parameters. Stochastic methods based on Markov-Chain Monte Carlo (MCMC) sampling employing Metropolis-Hastings algorithm [24, 25] have been leading cosmological parameters estimation from CMB data. MCMC method have an interesting property that they ensures that the number density of the sampled points is asymptotically proportional to the joint probability distribution [14, 26–29]. There has been presented some modification in the usual Metropolis-Hastings algorithm to achieve better performance [29]. However, all the methods based on MCMC share the common weakness that the step size has to be chosen very carefully, in particular when the likelihood surface has multiple local maxima.

In a recent study [1] we have shown that particle swarm optimization or PSO (we call our code **COSMOPSO**), which is an artificial intelligence inspired population based search procedure, can be used to estimate cosmological parameters from the WMAP seven year data. At that time it was not clear to us whether PSO can be used as a “sampler” also like MCMC, apart from an optimizer. By carrying out a large number of simulations, we have found that if the convergence of PSO can be delayed and we use multiple realizations of PSO then PSO can also sample the parameter space quite effectively to the extent that we can use the same analysis pipeline to process the PSO sampled points which is used to process the points

---

\*Electronic address: jayanti@iucaa.ernet.in

sampled in MCMC i.e., `GetDist`.

The plan of this paper is as follows. In section (2) we show that PSO particles in `COSMOPSO` do sample the parameter space effectively and present the results which we get for the WMAP nine year and Planck data. In section (3) we present an implementation of the downhill simplex method of Nelder and Mead [30] for cosmological parameter estimation (we call the code `COSMODSM`) for the WMAP nine year data. In section (4) we present an implementation of Powell's method of Bound Optimization BY Quadratic Approximation (BOBYQA) for cosmological parameter estimation (we call our code `COSMOBOBYQA`) and compare `COSMOMC`, `COSMOPSO`, `COSMODSM` with `COSMOBOBYQA` and present results for WMAP nine year data. In section (5) we present the discussion and conclusions of our work.

## II. PARTICLE SWARM OPTIMIZATION

PSO was proposed by Kennedy and Eberhart [31, 32] in 1995 and since then there have been proposed many modifications of PSO for different type of applications (for recent updates see [33, 34]). PSO is a population based search procedure which carry out local and global search in a multi-dimensional parameter space simultaneously [31–35]. PSO has been regularly used in engineering problems since its inception, however, recently it has become common in astronomy also [1, 36–39]. In PSO, a team of particles or computational agents is launched in the multi-dimensional parameter space which is driven by the exploration of all the individual particles and the team together. If the position and the velocity of  $i^{th}$  PSO particle at “time”  $t$  are represented by  $X^i(t)$  and  $V^i(t)$  respectively then according to the PSO algorithm the velocity  $V^i(t+1)$  at time  $(t+1)$  is computed as

$$\begin{aligned} V^i(t+1) = & wV^i(t) + c_1\xi_1[P^i - X^i(t)] \\ & + c_2\xi_2[G - X^i(t)], \end{aligned} \quad (1)$$

and the position is updated as:

$$X^i(t+1) = X^i(t) + V^i(t+1). \quad (2)$$

In Eq. (1)  $P^i$  is the point at which the  $i^{th}$  particle has found the maximum value of the cost function  $f(X)$  (in our case minimum value of  $-\log \mathcal{L}$  where  $\mathcal{L}$  is the likelihood function) and is called its **Pbest** or the “personal best” and  $G$  is the location of that **Pbest** which is highest and called the “global best” or **Gbest**. The coefficients  $c_1$  and  $c_2$  are called acceleration coefficients and their values determine the weight we want to give to the exploration of the individual particles and the team respectively. The coefficient  $w$  is called the inertia weight and its value decides the weight which we want to give the “inertial” motions of the particles and  $\xi_1$  and  $\xi_2$  are two uniform random numbers in the range  $[0, 1]$ .

The second and the third term in the right hand side of Eq. (1) resemble a simple harmonic motion (Hooke's

law) and corresponds to the force  $F^i(t)$  acting on particle  $i$  at time  $t$ :

$$F^i(t) = -k[X^i(t) - \tilde{X}], \quad (3)$$

where  $\tilde{X}$  is the location of the **Pbest** or **Gbest** and  $k = c_i\xi_i, i = 1, 2$ , is the force constant which is a stochastic variable here. Eq. (3) shows that PSO particle oscillate around the **Pbest** and **Gbest** with gradually decreasing random amplitudes. In the beginning of PSO particles exploration there are a very few particles which have their **Pbest** close to the **Gbest** (one particle has its **Pbest** and **Gbest** the same), however, when the particles approach close to the global maximum of the cost function all the particles have their **Pbest** close to the **Gbest** and they all oscillates around the points which are very close to each other.

In order to show the behavior of the cost function  $f(X)$  close to the best-fit point we Taylor expand  $f(X)$  around the best fit point  $X_0$ , in the following way:

$$f(X) = f(X_0) + \frac{\partial^2 f(X)}{\partial X^2}|_{X=X_0}(X - X_0^2) + \dots, \quad (4)$$

which shows that very close to the best fit point  $X_0$ , every cost function  $f(X)$  can be approximated as a quadratic function so it is not required that the sampling method we use have more complex strategy than what is needed to sample a quadratic function. Since step size of PSO particles is approximately proportional to their distance from the global maximum (the approximation becomes better when we reach close to the global maximum) therefore the density of sample points increases in a natural way. We support the claim that PSO also can sample the parameter space effectively by estimating the cosmological parameter estimation from the WMAP nine year and Planck data. We arrange the points which are sampled by PSO into “chains” and use the `COSMOMC` analysis pipeline `GetDist` on that.

In any PSO implementation we not only need to chose the values of the design parameters  $c_1, c_2, w$  and the number of PSO particles, we also need to select a method for setting the initial positions and velocities of particles, maximum velocity, boundary condition and a termination criteria. Apart from the termination criteria, we keep the values of the design parameters and other consideration the same as we used in [1]. For termination, we stop PSO exploration when find that the change in the cost function is smaller than a user given value for a given number of steps.

Our cosmological parameter estimation code based on particle swarm optimization named `COSMOPSO` has three main components which are used for computing theoretical  $C_{ls}$ , computing likelihood and evolving PSO particles. Since we have already discussed the third component, here we would like summarize the first and the second components.

For computing the angular power spectra of CMB temperature and polarization anisotropies for a given theoretical model (set of cosmological parameters) we use

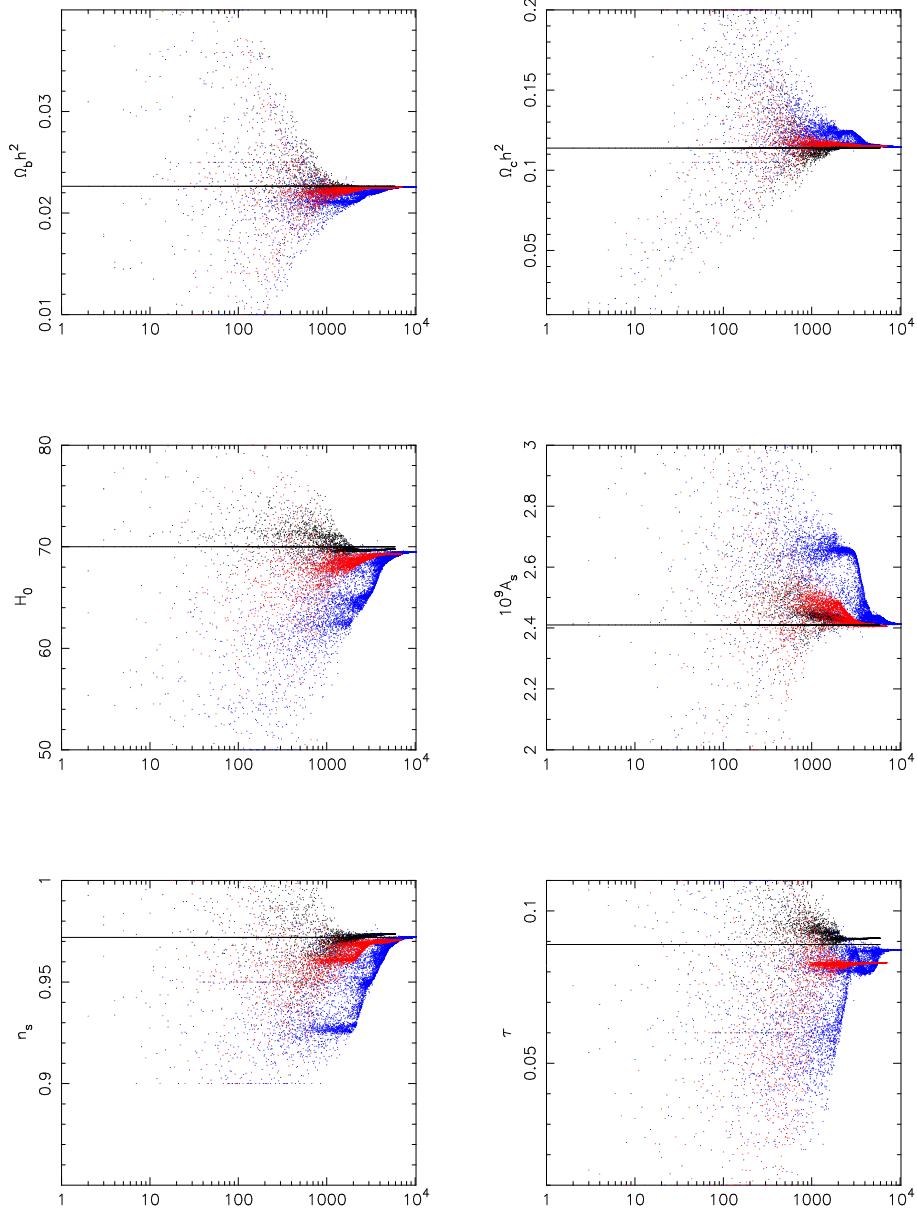


FIG. 1: The figure shows the positions of PSO particles along different directions with iterations (x-axis) for three different realizations (shown by different colors). PSO particles not only finally reach the point at which the cost function (likelihood) is maximum they also make random walks in way that the size of the jump at every stage is roughly proportional to their distance from the global maximum. As a result of this type of random walk the density of sample points increases when we approach towards the global maxima. In the above figure the distribution of particles along x-axis is not important since the number of steps which are needed for convergence are different for different realizations.

publicly available code **CAMB** [40] which employs a line of sight integration approach as was given in [21]. We use April 2014 version of **CAMB** which is different than what was used in WMAP nine year and Planck data analysis so some difference are expected in the values of cosmological parameters we report here.

In **CAMB** we vary only six cosmological parameters named physical densities of baryons ( $\Omega_b h^2$ ) and dark matter ( $\Omega_c h^2$ ), the Hubble parameter at present ( $H_0$ ),

the amplitude ( $A_s$ ) and spectral index ( $n_s$ ) of primordial scalar perturbations at some pivot scale ( $k_*$ ), and the optical depth of reionization ( $\tau$ ) epoch. Note that we consider  $H_0$  as a fitting parameter and not the angular size ( $\theta$ ) of the last scattering surface as is commonly done (in **COSMOMC**) and have some advantages over  $H_0$ . We do not consider dark energy density ( $\Omega_\Lambda$ ) as a fitting

parameter and compute that from the flatness condition:

$$\Omega_\Lambda = 1 - \frac{\Omega_b h^2 + \Omega_c h^2}{(H_0/100)^2}. \quad (5)$$

We use pivot scale for the scalar and tensor power spectrum  $0.002\text{Mpc}^{-1}$ , when using only WMAP nine year data and use pivot scale  $0.05\text{Mpc}^{-1}$ , when using WMAP nine year and Planck data combined. We use `l_max_scalar=2600`, `l_max_tensor=1500`, `l_eta_max_scalar=4000`, `l_eta_max_tensor=3000`, the width of reionization redshift  $\Delta z = 0.5$ , CMB temperature  $T_{cmb} = 2.72548$ , Helium fraction  $Y_{He} = 0.24$ , massless neutrino species  $n_\nu = 3.04$ . We keep logical variables `AccurateReionization` and `AccurateBB` true and do not compute tensor perturbations and keep `WantTensors` false in `CAMB`.

The procedure which we follow to estimate cosmological parameters is as follows. We use eight different realizations (different seeds for random number generator) of PSO which have been converged for sampling. For every realization we sorted the sample points on the basis of the likelihood value and wrote them in the form of chains as are needed by the `GetDist` program of `COSMOMC` to compute the best-fit values, 68% limits and confidence regions. In order to make one-dimensional probability distribution (marginalized) and two-dimensional contours (68% and 95% confidence regions) smooth, smoothing parameters are passed to `GetDist` parameter file. We found that PSO needs more smoothing than what is needed for MCMC samples.

### A. WMAP nine year data

In order to estimate cosmological parameter with `COSMOPSO` and demonstrate that PSO can also sample the parameter space quite effectively, apart from finding the best fit values, here we present the result for the WMAP nine year data which has been made publicly available by the WMAP team [41]. WMAP team has also made a FORTARN code available with the data to compute likelihoods for CMB temperature and polarization data for a user given theoretical model i.e., theoretical  $C_{ls}$ . There have been a few changes in the way likelihoods are computed for the nine year release [27], however, the basic approach has been the same which was outlined in [42]. WMAP nine year likelihood code computes likelihood differently at low and high- $l$  for temperature and polarization power spectra [43]. At high- $l$  ( $l > 32$ ) the likelihood is no longer computed using `MASTER` [44] (as was the case for earlier data releases) and use an optimally estimated power spectrum and errors based on the quadratic estimator [45]. At low- $l$  ( $l < 32$ ) the TT-likelihood is computed using the Blackwell-Rao estimator which employs Gibbs sampling and uses bias-corrected Internal Linear Combination (ILC) map as an input [46].

WMAP nine year likelihood does not have any extra parameter and needs only the theoretical power spec-

tra for computing the likelihood. Some of the parameters which are used in the WMAP likelihood code are `l_max=1200` and 800 and for TT and TE respectively. We also set `use_TT`, `use_TE`, `use_lowL_TT`, `use_lowL_TE` true in the WMAP nine year likelihood code.

The prior which we use for cosmological parameters are given in Tab. (I). Note that range which we consider here are different than what are considered in `COSMOMC` mainly because all the three methods which we consider in the present work demand that we should be able to compute the likelihood function at all the points in the search space.

One of the results which we would like to highlight is that the sampling of the parameter space done by PSO particles. In Fig. (1) we show the distribution of particles in PSO along the directions of different cosmological parameters for three different realizations. From the distribution of particles we can see that in a single realization most likely we will not have a symmetric distribution of particles around the global best position. However, when we use multiple realizations of PSO which have been converged we get more and more symmetric distribution. Which is important if we want to sample the parameter space close to the global maximum fairly. The final confirmation of the distribution of particles can be done by looking at the one dimensional marginal distribution of cosmological parameters in Figs. (2) and (3).

We show the best fit cosmological parameters which we estimate using `COSMOMC` and `COSMOPSO` for the WMAP nine year data with their 68% limits in Tab. (II). Note that there is a slight difference between the `COSMOMC` best-fit parameters which we present in Tab. (II) and published in [47] which can be attributed to the different versions of codes we are using. From the table we can see that PSO is not only able to find the best-fit point, it is also able to find the mean and 65 % limits of the parameters quite close to what we get from `COSMOMC` for the same data. We would like to mention that the sampling done in `COSMOPSO` is not as good as in `COSMOMC` and as a results that we get slightly non-Gaussian one-dimensional marginal probability distributions as compared to `COSMOMC`. We also find that the minimum value of  $-\log \mathcal{L}$  which we get in `COSMOPSO` is very close to what we get in `COSMOMC` (in fact `COSMOPSO` gives us slightly better likelihood).

In Fig. (2) we show the one dimensional marginal probability distributions for the six cosmological parameters in the diagonal panels and 68% and 98% confidence regions in the rest of the panels. From these panels we see that the marginal probability distributions and two-dimensional joint probability distribution as shown by the contour plots show expected behavior. This clearly shows that the way PSO particles make random walks in the parameter space can be used for sampling the parameter space quite effectively.

S. No	Parameter	Prior	Description
1	$\Omega_b h^2$	[0.01, 0.04]	Baryon density today
2	$\Omega_c h^2$	[0.01, 0.20]	Cold dark matter density today
3	$H_0$	[50, 80.0]	Hubble parameter at present
4	$10^9 A_s$	[2.0, 4.0]	Primordial curvature perturbations
5	$n_s$	[0.9, 1.0]	Scalar spectrum power-law index
6	$\tau$	[0.01, 0.11]	Thomson scattering optical depth due to reionization

TABLE I: Cosmological parameters and their priors we used in **COSMOPSO**, **COSMODSM** and **COSMOBOBYQA**. Note that the range we have considered is different than what is generally used (in **COSMOMC**) mainly because all the three methods we considered demand that we must be able to compute the likelihood function at every point in the prior range.

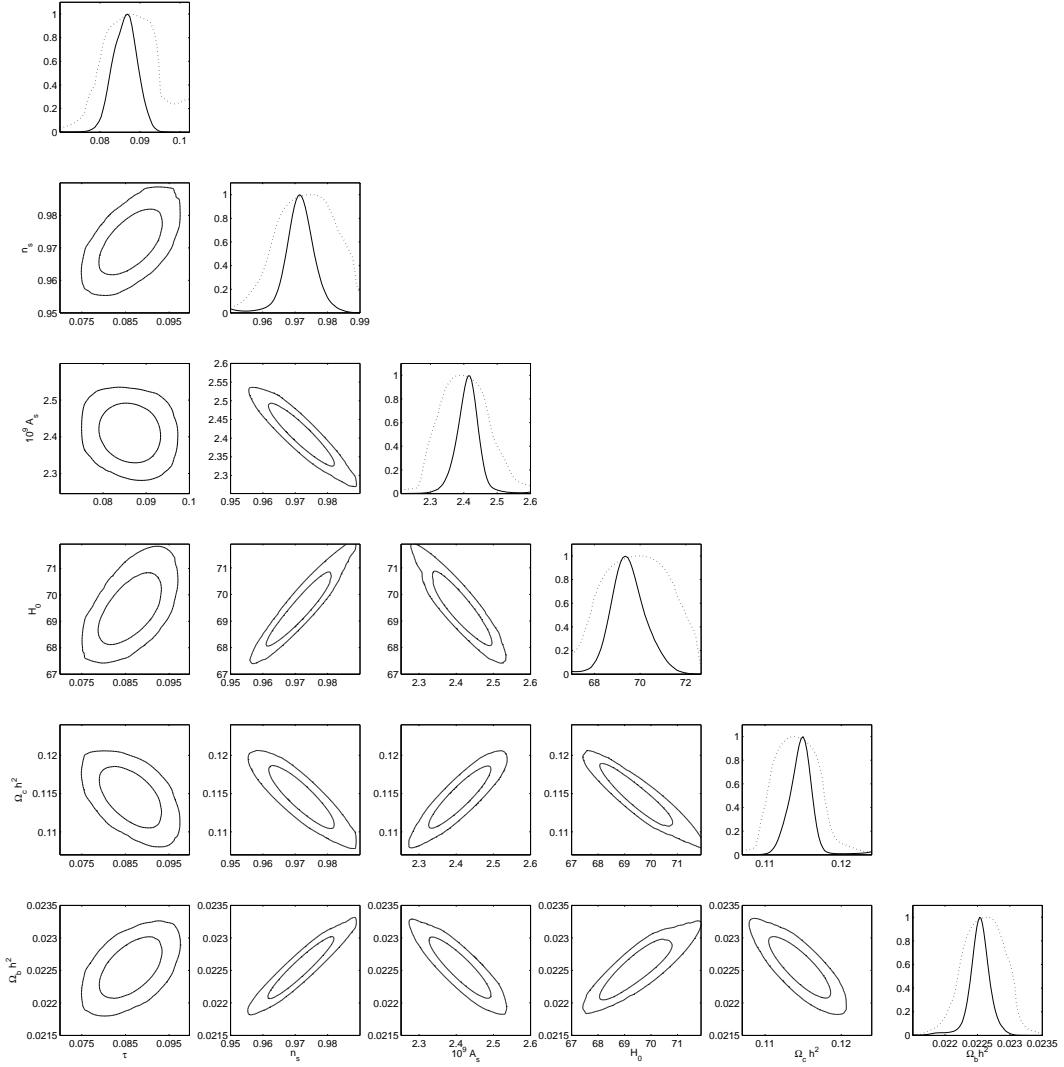


FIG. 2: The diagonal panels in this figure show the one-dimensional marginalized distributions and other panels show two-dimensional marginalized constraints (68% and 95% CL) for the standard six cosmological parameters for the WMAP nine year data which we obtained by combining the sampled points of eight different realizations of PSO and using the **Getdist** (the program which is used for analysis in **COSMOMC**) on those.

### B. WMAP nine year + Planck data

In this section we present our analysis for the WMAP nine year + Planck data. A detail description of the

Planck project and the data product is given in [48], however, here we would like to mention that Planck has more frequency channels over wide frequency range, higher an-

Parameter	PSO		COSMOMC	
	Best Fit	68% Limit	Best Fit	68 % Limit
$\Omega_b h^2$	0.02260	$0.02252 \pm 0.00018$	0.02259	$0.02265 \pm 0.00048$
$\Omega_c h^2$	0.1140	$0.1148 \pm 0.0018$	0.1122	$0.1137 \pm 0.0046$
$H_0$	69.827	$69.397 \pm 0.9758$	70.059	$69.660 \pm 2.178$
$10^9 A_s$	2.395	$2.416 \pm 0.044$	2.369	$2.402 \pm 0.068$
$n_s$	0.9733	$0.971 \pm 0.005$	0.978	$0.9730 \pm 0.013$
$\tau$	0.086	$0.086 \pm 0.003$	0.086	$0.088 \pm 0.014$
$-\log \mathcal{L}$	3778.7840		3779.0220	

TABLE II: Cosmological parameters estimated using **COSMOMC** and PSO for the WMAP nine year temperature and polarization data. Note that the latest version of **COSMOMC** computes  $A_s$  at pivot scale  $k_0$ ,  $0.05 \text{ Mpc}^{-1}$  so we convert the best fit and mean values to  $0.002 \text{ Mpc}^{-1}$  for comparison.

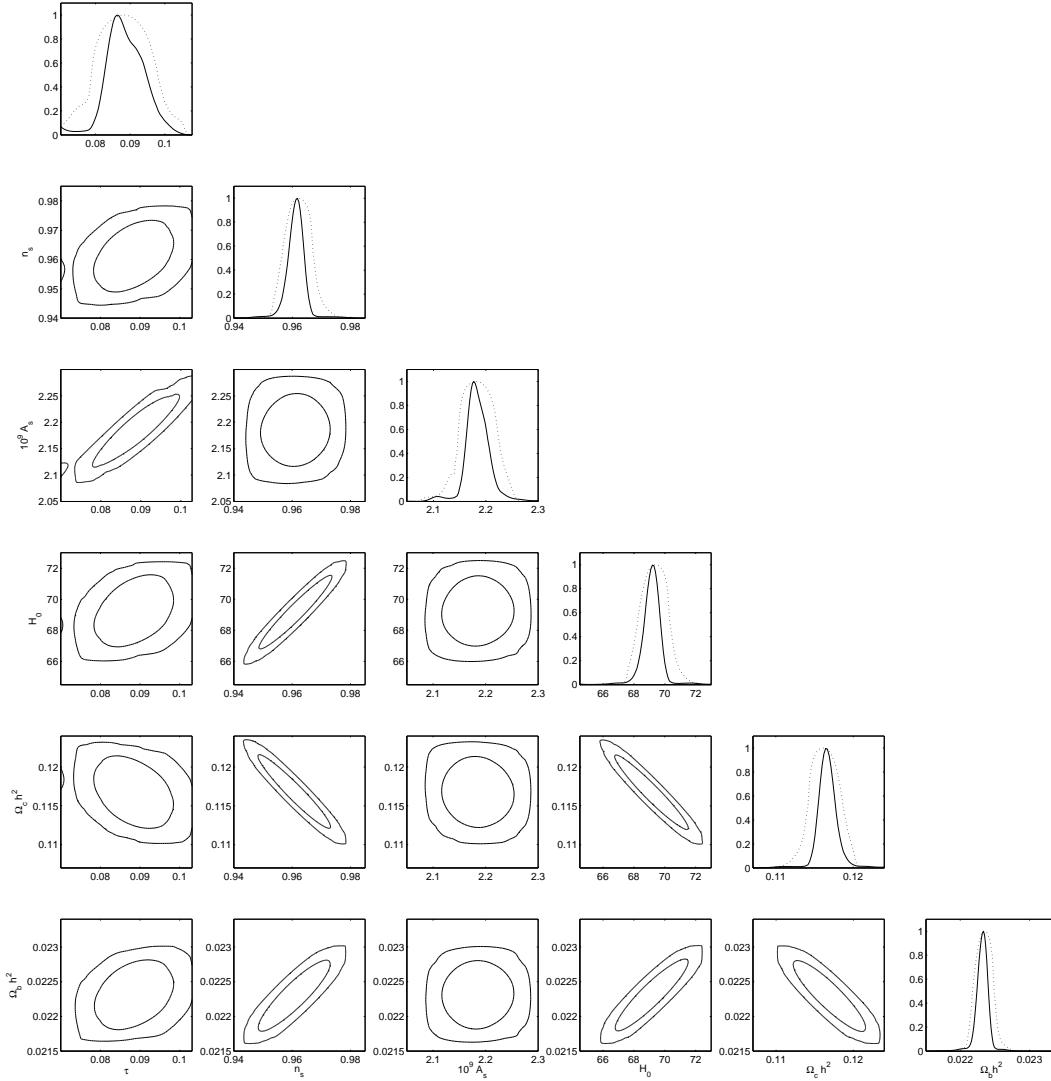


FIG. 3: The same as in Fig. (2) for the WMAP nine year + Planck data.

gular resolution and better noise sensitivity which makes it far better equipped to handle different type of systematic (foreground etc.) than WMAP. At present Planck has made only temperature data publicly available which

can be used alone or with a combination of other CMB data sets to constrain theoretical models. Planck collaboration has also made a code named **P1c** publicly available for computing the likelihood for temperature, polariza-

Parameter	PSO			COSMOMC	
	Best Fit	68% Limit		Best Fit	68 % Limit
$\Omega_b h^2$	0.02237	$0.02231 \pm 0.00001$		0.02233	$0.02237 \pm 0.00020$
$\Omega_c h^2$	0.1159	$0.1167 \pm 0.0014$		0.1166	$0.1163 \pm 0.0020$
$H_0$	69.508	$69.138 \pm 0.683$		68.688	$68.835 \pm 0.975$
$10^9 A_s$	2.1759	$2.1838 \pm 0.023$		2.1741	$2.1832 \pm 0.037$
$n_s$	0.9628	$0.9610 \pm 0.003$		0.9614	$0.9618 \pm 0.005$
$\tau$	0.087	$0.088 \pm 0.005$		0.086	$0.088 \pm 0.009$
$-\log \mathcal{L}$	8694.4260			8693.6840	

TABLE III: Cosmological parameters estimated using COSMOMC and PSO for the WMAP nine year (temperature and polarization data) + Planck (temperature data).

tion and lensing data. The code can be downloaded from [49] and the detail explanation about the methodology of power spectrum and likelihood estimation can be found in [50].

In order to compute likelihood for a given cosmological model, which is represented by a set of six angular power spectra (TT, EE, BB, TE, TB, EB), we use Planck likelihood software `P1c`. In Planck likelihood the high-l (up to  $l = 2500$ ) TT likelihood is computed using `CAMspec` which needs 14 extra (nuisance) parameters which we fix to their best fit values given in [51] and vary only the six cosmological parameters which directly affect  $C_{ls}$ . Planck likelihood software computes low-l ( $l = 0$  to  $l = 49$ ) TT likelihood using `commander` and low-l polarization likelihood is computed using WMAP nine year data which needs TT, EE, BB and TE power spectra from  $l = 0$  to  $l = 32$ . Since power at  $l = 0$  and  $l = 1$  does not have any sense so we keep their values  $-1$  in the Planck likelihood. In `COSMOPSO` we add all the three Planck likelihoods (after changing the sign) to the WMAP nine year likelihood for our optimization function or the cost function. In order to cross check that we get consistent results we compare those with `COSMOMC` results.

The best fit cosmological parameters which we get for the WMAP nine + Planck data which we get for `COSMOMC` and `COSMOPSO` are given in Tab. (III). From the table we can re-confirm that PSO not only can find the best fit cosmological parameters, which gives the maximum value of the likelihood function, it can also find consistent estimates of the 68% limits of the cosmological parameters. For the WMAP nine year + Planck data also we have consider eight different realization of `COSMOPSO` which have been converged and create chains out of the sample points to process with `GetDist`. In Fig. (3) we show one dimensional marginalized probability distribution and 68% and 95% confidence regions of the estimated parameters and for those we can see that PSO gives consistent result for the joint data also.

### III. DOWNHILL SIMPLEX METHOD (DSM)

Downhill simplex method was introduced by Nelder and Mead [30] in 1965 exactly for the purpose we are

using it here i.e., “maximization of a likelihood function, in which the unknown parameters enter non-linearly”. On the basis of Nelder and Mead’s algorithm [52] has written a quite efficient program named ‘amoeba’. A great detail about the algorithm and implementation is also given in [52]. In place of directly using amoeba we have written our code from the scratch on the basis of the algorithm given in the original paper [30]. Since there are some minor difference between the algorithm given in [30] and implemented in [52], therefore we would like to give an overview of the algorithm which is quite close to what is given [53] also.

#### A. Basics

A geometric object with  $(n + 1)$  vertices in a  $n$ -dimensional space is called a “simplex”. For example, a triangle in a two-dimensional flat surface is a simplex. In one-dimensional optimization (root finding for example) we can “bracket” the solution between two points and keep making our bracket smaller and smaller following some algorithm and finally reach close to the solution. In more than one dimensional space there are no “bracketing” methods available and so the problem is much more challenging.

In the downhill simplex method we begin with an initial simplex i.e., set of  $(n + 1)$  points in a  $n$  dimensional space and replace the point at which the cost function is maximum (the worst point) with a new point which we obtain by carrying out three type of geometrical operations named reflection, expansion and contraction. We keep repeating the process until the values of the function at all the  $(n + 1)$  points do not become close to each other (within a user defined tolerance). In downhill simplex method the simplex adapts the local geometry, in the sense, that it gets elongated when encounter a long inclined place, reflect when face a valley at an angle and contract close to a minimum.

The most important consideration of downhill simplex method are (1) the choice of the initial simplex (2) tolerance and (3) the values of the expansion and contraction coefficients. We keep the values of reflection coefficients  $\alpha$ , expansion coefficient  $\beta$  and the contraction coefficients  $\gamma$  to their standard values  $-1$  and  $2$  and  $1/2$  respectively.

Choosing tolerance is easy (we can chose a sufficiently small number) but choosing an initial simplex is a bit harder since we want to make sure that the simplex always remain within the allowed range without imposing any boundary condition.

## B. The Algorithm

Since the downhill simplex algorithm strictly demands that we should be able to compute the (cost) function at all the vertices at every step therefor we must make sure that the values of the parameters always remain in the permitted range. We use the same range of cosmological parameters in **COSMODSM** which we use for **COSMOPSO** and is given in Tab. (I). Since there is no standard prescription for choosing the starting simplex in Downhill-Simplex method therefore we chose an ad-hoc prescription which is as follows.

We represent the  $i^{th}$  vertex of the simplex at iteration (or “time”)  $t$  by  $X^i(t)$  and the minimum and maximum value of it are given by  $X_{\min}$  and  $X_{\max}$ . Note that here  $X$  is a vector which has the dimensions of the parameter space (in our case it is six for the standard cosmological model).

The first vertex of the initial simplex is set as :

$$X^1(0) = X_{\min} + a(X_{\max} - X_{\min}), \quad (6)$$

where  $a$  is a user defined parameter in the range  $[0, 1]$  (we recommend it to be less than 0.5).

The other remaining  $n$  vertices of the initial simplex are set as:

$$X^j(0) = X^1(0) + \lambda_j e^j, \quad (7)$$

where  $j = 2, \dots, n+1$  and  $e^j$  are  $n$  unit vectors. The coefficients  $\lambda_j$  depend on the search range:

$$\lambda_j = b(X_{\max}^j - X_{\min}^j), \quad (8)$$

where  $b$  is a user defined parameter in the range  $[0, 1]$  (we recommend it to be less than 0.5) and  $j = 1, \dots, n$ . Note that there is some degeneracy between  $a$  and  $b$  and in any ideal parametrization this should be avoided. Our parameterization is as good as choosing the initial points by trial and error.

In **COSMODSM** at every iteration we find the vertices which have the largest and smallest values of the cost function, called these  $i_h$  and  $i_l$  respectively. For better readability we also represent  $X_{i_h}$  by  $X_h$  and  $X_{i_l}$  by  $X_l$ . The second highest point is also needed and we represent that by  $X_{nh}$ . We also compute the centeroid of  $n$  vertices (keeping the  $i_h$  aside) and represent that with  $\bar{X}$  with respect to which all the geometric operations, reflection, expansion and contraction are carried out. We represent the value of the cost function by  $f(X)$  and so :

$$f(X_h) \geq f(X_i) \quad \text{for } i = 1, 2, \dots, n+1, \quad (9)$$

and

$$f(X_l) \leq f(X_i) \quad \text{for } i = 1, 2, \dots, n+1. \quad (10)$$

The centeroid  $\bar{X}$  is defined as:

$$\bar{X} = \frac{1}{n} \sum_{i=1, i \neq i_h}^{i=n+1} X^i \quad (11)$$

We use the same convergence criteria which is used in [52] for which we compute:

$$\epsilon = 2 \frac{f(X_h) - f(X_l)}{|f(X_h)| + |f(X_l)|}, \quad (12)$$

and stop the search when  $\epsilon < \epsilon_0$  where  $\epsilon_0$  is a user defined number.

In [30] three different equations are used for reflection, expansion and contraction, however, we use a single function  $g(X, \xi)$  as defined below to carry out all the three operations:

$$g : X \longrightarrow X' = \bar{X} - \xi(\bar{X} - X), \quad (13)$$

where  $\xi = \alpha, \beta$  or  $\gamma$  with  $\alpha = -1, \beta > 1$  and  $\gamma < 1$ . We consider  $\beta = 2$  and  $\gamma = 1/2$  in the present work.

The meaning of  $\xi$  can be easily understood by writing:

$$\xi = \frac{X' - \bar{X}}{X - \bar{X}}, \quad (14)$$

which means that  $\xi$  is the ratio of the distance of the new point  $X'$  and the old point  $X$  from the centeroid  $\bar{X}$ .

The main steps of the Nelder-Mead Algorithm can be summarized in the following way:

1. Set up the initial simplex  $X^i(0), i = 1, 2, \dots, n+1$ , compute the values of the function  $f_i(0) = f(X^i(0))$ , and find out the highest (worst) and lowest (best) point  $X_h(0)$  and  $X_l(0)$ , and the centeroid  $\bar{X}$  of the best  $n$  points. We also need to find the second worst (highest) point  $X_{nh}$ .
2. If the convergence is not reached reflect the worst point  $X_h$  about the centeroid:

$$X_r = \bar{X} - \alpha(\bar{X} - X_h), \quad (15)$$

and go to the next step.

3. If we find  $f_l \leq f_r < f_{nh}$  than that means we have found a point which is better than the worst point  $X_h$  so we replace  $X_h$  with  $X_r$  and recompute  $X_l, X_h$  and  $\bar{X}$  and go to step (2).
4. If we find  $f_r < f_l$  that means we are going in the right direction and it is useful to expand the new point  $X_r$  along the same direction and get a new point  $X_e$ :

$$X_e = \bar{X} - \beta(\bar{X} - X_r) \quad (16)$$

If we find that  $f_e < f_r$  than we replace the worst point  $X_H$  with  $X_e$  otherwise we replace the worst point with  $X_r$  and go to step (2).

S. No	(a,b)	iterations	$\Omega_b h^2$	$\Omega_c h^2$	$H_0$	$A_s$	$n_s$	$\tau$	$-\log \mathcal{L}$
1	(0.20,0.60)	428	0.022546	0.113998	69.689714	2.400900	0.971840	0.085522	3778.8044
2	(0.20,0.40)	330	0.022648	0.114334	69.782342	2.395106	0.974042	0.086739	3778.7934
3	(0.25,0.40)	348	0.022603	0.114112	69.811296	2.396340	0.973377	0.086554	3778.7840
4	(0.21,0.42)	356	0.022601	0.114102	69.804714	2.396605	0.973349	0.086532	3778.7839
5	(0.40,0.20)	306	0.022609	0.113243	70.223014	2.334731	0.973016	0.074727	3779.3774
6	(0.30,0.40)	364	0.022601	0.114110	69.807095	2.399997	0.973447	0.087404	<b>3778.7830</b>
7	(0.35,0.25)	374	0.022588	0.113895	69.868690	2.402573	0.972920	0.087475	3778.7909
8	(0.50,0.20)	222	0.022345	0.115229	68.930597	2.412138	0.965558	0.077042	3779.1554

TABLE IV: Best fit cosmological parameters for eight different initial conditions for Downhill-Simplex methods. Apart from the values of parameter  $a$  and  $b$  we have kept other parameters fixed i.e.,  $\epsilon = 10^{-8}$  and  $\alpha = -1, \beta = 2$  and  $\gamma = 1/2$ . How rapidly the algorithm converges depend on the values of the parameters  $a$  and  $b$  i.e., the initial simplex. The highest value of the likelihood function is shown in bold.

5. If we find  $f_r \geq f_{nh}$  that means the simplex is too big and we need contraction. However, before that we test:

- (a) If  $f_r \geq f_h$  i.e., the reflected point is worse than the worst or is as worse as the worst we contract

$$X_c = \bar{X} - \gamma(\bar{X} - X_h). \quad (17)$$

If we find that the point  $X_c$  is worse than the worst  $X_h$  we shrink the whole simplex around the best point  $X_l$ :

$$\tilde{X}_i = \frac{1}{2}(X_i + X_l), \quad (18)$$

with  $i = 1, 2, \dots, n+1$  and go to step (2)

- (b) If  $f_r < f_h$  then we contract the reflected point  $X_r$ :

$$X_c = \bar{X} - \gamma(\bar{X} - X_r), \quad (19)$$

and if  $f_c \leq f_r$  then we replace  $X_h$  with  $x_c$  otherwise again shrink the simplex around the best point  $X_l$  and go to step (2).

In any iterative algorithm like PSO and downhill simplex in general it is not that important what convergence criteria we choose since most of them look for the progress made in each iteration and stop the search when the improvement falls below a tolerance limit or remains below a tolerance limit. All the three methods which discussed in the present work have different criteria, however, they all have a very good agreement with the final results.

In order to demonstrate that the downhill simplex method of Nelder and Mead can successfully find the best fit cosmological parameter we considered WMAP nine year data for our analysis. Apart from the programs for the exploration of parameter space, we used the same program to compute the cost function (likelihood) which we use in COSMOPSO so there is no need to further elaborate on that here and so we will only present the results here.

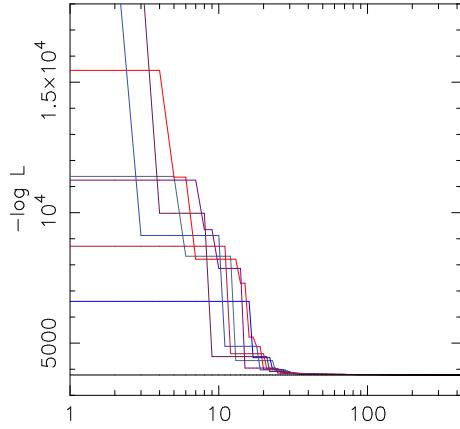


FIG. 4: Change in the cost (likelihood) function for the seven vertices we have (shown by different colors) as the Downhill Simplex algorithm iterates (x-axis shows the iterations) for the WMAP nine year data in COSMODSM. The minimum (best) value of the cost function is also shown by the straight line.

In Fig. (4) we show the change in the values of cost function at the vertices of the simplex as the simplex evolves. From the figure we can see that like in any other iterative method in DSM also in the beginning we have huge improvement in the cost function which falls rapidly which means in DSM also (like PSO) the convergence is reached only asymptotically.

We show the best fit cosmological parameters which we get for the WMAP nine year data using COSMODSM in Tab. (IV) for eight different initial conditions i.e, different values of the parameters  $a$  and  $b$ . We find that apart from having different number of iterations for convergence they all agree with the values of the best fit cosmological parameters with a reasonable limit.

We also tried to see if DSM also can sample the parameter space like PSO and found that it does a very poor job and also suffer from the drawback of early convergence. In order to show that in Fig. (5) and (6) we show the the vertices of simplex in COSMODSM and trajectories of a few PSO particles and COSMOPSO respectively.

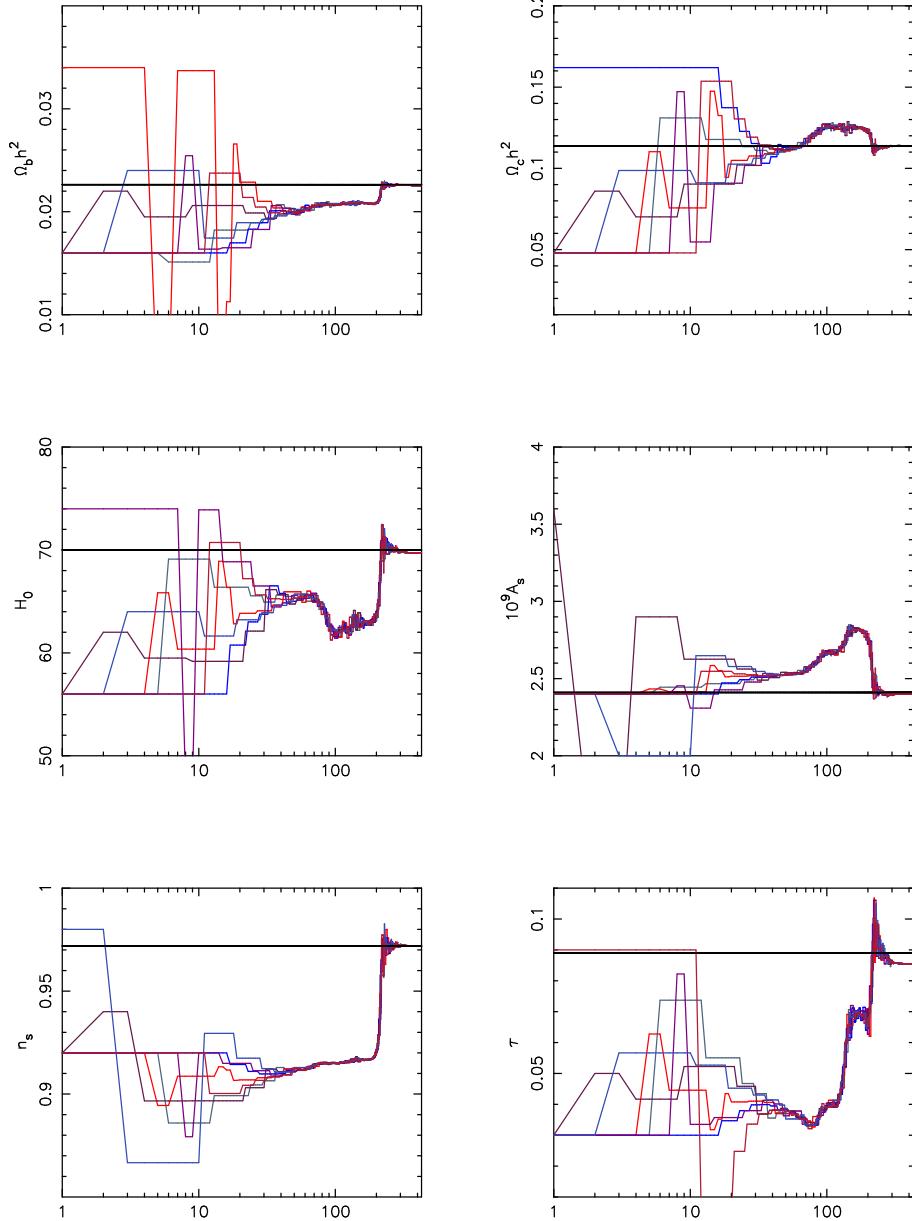


FIG. 5: Change in the vertices (shown by different colors) of the simplex with iterations (x-axis) along different directions i.e., cosmological parameters. The best fit values of the parameters are also shown by straight lines.

From these figures it is quite clear that PSO does sample a large volume of parameter space due to its stochastic nature which is not true for DSM.

#### IV. POWELL'S METHOD OF BOUND OPTIMIZATION BY QUADRATIC APPROXIMATION (BOBYQA)

Since M. J. D. Powell gave his method of unconstrained optimization by quadratic approximation in 1962 [54] there have been many changes in his approach of non-

linear optimization without derivative as is reflected by the software COBYLA, NEWUOA, BOBYQA and LINCOA [55] released by him time to time. In the present work we use BOBYQA FORTRAN 77 package which is available for downloading from [55].

Powell's method of Bound Optimization BY Quadratic Approximation (BOBYQA) is a technique to find the global maximum/minimum of a function  $f(X)$ ,  $X \in \mathcal{R}^n$ , with  $X_{\min} \leq X \leq X_{\max}$  without computing derivative of the cost function  $f(X)$ . The method is based on approximating the function  $f(X)$  by a quadratic function  $q(X)$  at a set of  $m$  points  $Y$  which are chosen and adjusted

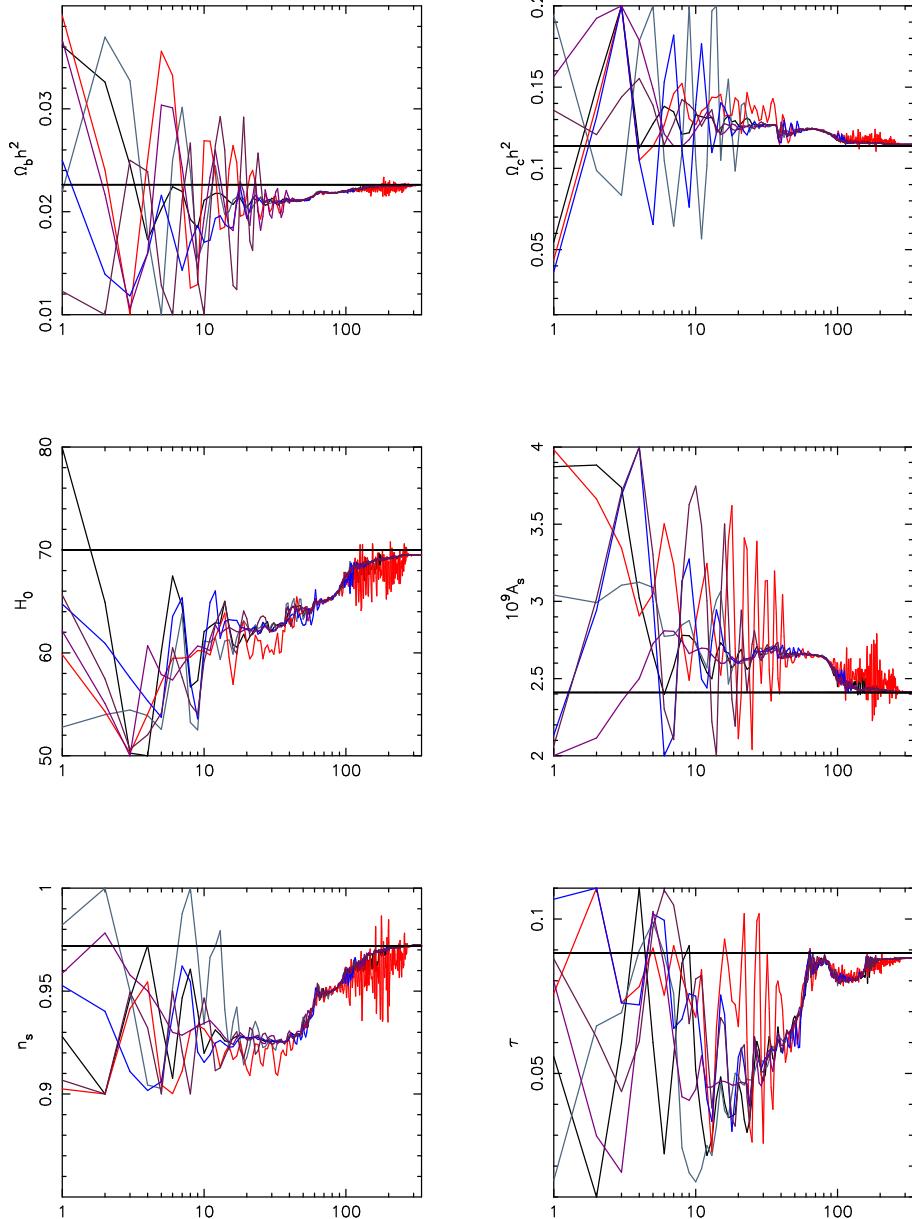


FIG. 6: Change in the positions of a few PSO particles with iterations (x-axis) along different directions. PSO particles sample much broader area of the parameters space along their journey towards the best fit point as compared to the vertices of simplex in Downhill Simplex method as shown in Fig. (5).

automatically.

$$q(X) = A + B^T(X - \tilde{X}) + (X - \tilde{X})^T C (X - \tilde{X}), \quad (20)$$

where  $A$  is a constant,  $B$  a column vector with  $n$  elements and  $C$  is a  $n \times n$  symmetric matrix and in total we have  $m = 1 + n + n(n+1)/2 = (n+1)(n+2)/2$  coefficients to fit the quadratic function for which we consider a set of  $m$  interpolating points  $Y$  at iteration  $k$  at which the approximation holds:

$$f_k(Y_j) = q(Y_j). \quad (21)$$

At every iteration  $k$  we select a point  $X_k$  from  $Y_j$  which has the property:

$$f(X_k) = \min(f(Y_j), j = 1, \dots, m) \quad (22)$$

which we keep updating just like we keep updating the worst vertex at every iteration as we do in the Downhill-Simplex method.

The quadratic approximation holds true only within a small region characterized by a positive number  $\Delta$  called the “trust region radius” for which we must give an initial  $\Delta_{\text{beg}}$  and final  $\Delta_{\text{final}}$  values. The value of  $\Delta$  keep

changing with iterations and once it falls below  $\Delta_{\text{final}}$  we stop the program. Quadratic approximation within a small region is a common technique and is used in many other algorithms also including one we published in [56].

At iteration  $k$  we construct a vector  $d_k$  from  $X_k$  such that, (1)  $\|d_k\| < \Delta_k$ , (2)  $X_{\min} \leq X_k + d_k \leq X_{\max}$ , (3)  $X_k + d_k$  is not one of  $Y_j$ . Once we have a valid  $d_k$  we can update  $X_k$ .

$$X_{k+1} = X_k + d_k, \quad (23)$$

if  $f(X_k + d_k) < F(X_k)$  otherwise we set  $X_{k+1} = X_k$  and one of the interpolation points  $Y_t$  is replaced with  $X_k + d_k$ . After this has been done we generate  $q_{k+1}(X)$  and  $\Delta_{k+1}$  for  $k+1$  iteration according to the prescription given in [55].

In order to use BOBYQA for cosmological parameters estimation from the WMAP nine year data we made the following considerations. We compute theoretical CMB angular power spectrum  $C_l S$  using the publicly available code **CAMB** and use WMAP nine year likelihood code for computing the likelihood for temperature and polarization anisotropies as we do in section (II).

Since BOBYQA is designed in such a way that all the parameters must have values of the same order for which we normalize the six cosmological parameters with some default values so that they have the same order of  $X_{\min}^i, X_{\max}^i$  and the same values of  $X^i$  (where  $i = 1, \dots, 6$ ) at every iteration. This step is important otherwise the parameters like  $H_0$  which has a large value will hardly get updated. We use default values  $\Omega_b h^2 = 0.018, \Omega_c h^2 = 0.1, H_0 = 65.0, 10^9 A_s = 2.2, n_s = 0.94, \tau = 0.08$  and avoided to give the exact values of the best fit cosmological parameters known to us.

There is no prescription available for choosing the number of interpolation points  $m$  and one must find the appropriate value by trial and error. We have found that  $m = n+10$ , which is in the range  $[(n+2), (n+1)(n+2)/2]$ , works fine for our case. For  $\Delta_{\text{beg}}$  and  $\Delta_{\text{final}}$  also there is no standard prescription so we must set their values by trying out different values. We use  $\Delta_{\text{beg}} = 10^{-2}$  and  $\Delta_{\text{final}} = 10^{-12}$  and found that these values give us acceptable convergence i.e., convergence is reached after 340 iterations.

Here we would like to mention that all the versions of **COSMOMC** [57] October 2012 onward also use Powell's BOBYQA with some modifications [57] to find the best-fit cosmological parameters when we need only those. Our implementation of Powell's method is completely independent and different than what is used in **COSMOMC** in terms of the uses and the values of design parameters  $m$  and  $\Delta_{\text{beg}}$  and  $\Delta_{\text{final}}$ . It is also important to keep in mind that MCMC methods themselves also can find the best-fit point without any external optimizer, however, can take more time than **COSMODSM** or **COSMOBOBYQA**.

In Tab. (V) we show the cosmological parameters which we have estimated with four different methods named **COSMOMC**, **COSMOPSO**, **COSMODSM** and **COSMOBOBYQA**

and found that all the methods are able to find the correct values of the best fit cosmological parameters within reasonable time.

## V. DISCUSSION AND CONCLUSION

Building on our earlier work [1] in the present work we have shown that particle swarm optimization not only can be used to find the best fit cosmological parameters from CMB data sets like WMAP and Planck, it can also be used to sample the parameter space quite effectively. We have found that when using PSO as a sampler we must avoid early convergence and use multiple realizations of PSO so that the parameter space is covered uniformly. We present the results of our analysis of PSO for the WMAP nine year and Planck data and show that these are quite consistent with the standard results. In order to show that the sampling of the parameter space is quite effective we show that we can use the same analysis pipeline **GetDist** to process the PSO sampled point which is used to process the MCMC sampled points in **COSMOMC** and get consistent results which is clear from the one dimensional marginalized probability distribution and 68% and 95% confidence regions we show in Figs. (2) and (3) for the WMAP nine year and WMAP nine year + Planck data respectively.

We ran **COSMOMC** and **COSMOPSO** on a Linux cluster with multiple nodes using Message Passing Interface (MPI) library and shared memory parallelization library **OpenMP**. We have found that **COSMODSM** and **COSMOBOBYQA** are faster than **COSMOPSO** and **COSMOMC**, however, need fine tuning of the design parameters and have higher chances of failure. **COSMOPSO** has the problem of early convergence but that is not an issue if our aim is just to find the best fit point and not to sample the parameter space. In general, for 30 PSO particles in **COSMOPSO** we can achieve convergence within 400 steps, or less than 12000 computation of the cost function (**CAMB** calls + likelihood computation). For **COSMODSM** we need 400 steps with 7 vertices or around 2800 computation of the cost function. For **COSMOBOBYQA** we achieve convergence in around 400 steps or 400 computation of the cost function. It should be noted that all the different methods we have discussed here have different convergence criteria so direct comparison is not fair, however, whatever we have mentioned is qualitatively true i.e., **COSMOBOBYQA** method is faster than **COSMOPSO**, **COSMODSM** and **COSMOMC**.

In the present work we use only basic formalism of PSO, however, PSO has scope for modifications also some of which have been already done for different type of problems (see [33, 34]). We believe that with some modifications PSO should be able to become a robust method for sampling also in the Bayesian data analysis. One of the interesting properties of PSO is that the trajectories of PSO particles can be compared with trajectories of point particles in classical mechanics since they also follow trajectories which minimize some function (action)

Parameter	COSMOMC	COSMOPSO	COSMODSM	COSMOBOBYQA
$\Omega_b h^2$	0.022597	0.022601	0.022601	0.022630
$\Omega_c h^2$	0.112282	0.114044	0.114110	0.114148
$H_0$	70.06592	69.82771	69.807095	69.87489
$10^9 A_s$	2.177424	2.395936	2.399997	2.414575
$n_s$	0.973842	0.973390	0.973447	0.974537
$\tau$	0.859690	0.086614	0.087404	0.091901
$-\log \mathcal{L}$	3779.0220	3778.7839	3778.7830	3778.8495

TABLE V: We show the best fit cosmological parameters, which give the maximum likelihood, for **COSMOMC**, PSO, Downhill-Simplex methods and Powell's **BOBYQA** in the second, third, fourth and fifth columns of the above table respectively.

like PSO particles. However, this analogy cannot be stretched too far since for PSO particles time is not continuous i.e.,  $t \in \mathbb{Z}$ .

We also presented implementations of two new methods, Downhill simplex method of Nelder and Mead and Powell's method of Bound Optimization BY Quadratic Approximation (BOBYQA) in the present work which are much faster than PSO but are less robust, in the sense, PSO is very less sensitive to its design parameters and does not need any starting point (all the starting points are equally good) which is not the case for other algorithm. Since **COSMOPSO** is a completely parallel code (uses distributed memory parallelism using MPI and shared memory parallelism using OpenMP) therefore computational cost is not an issue for a cluster.

We have found that all the four methods (as are summarized in Tab. (V) are quite successful in finding the best-fit cosmological parameters in a reasonable time.

Cosmological parameter estimation from CMB data sets is still an active area of research in cosmology and optimization methods like PSO can be quite helpful not only in cosmological parameters but at other stages also of CMB data analysis pipeline. Being an independent and robust PSO has potential of becoming an essential

component of the CMB data analysis toolkit.

## Acknowledgments

The author acknowledges the use of codes **CAMB**, **COSMOMC** and **Planck** likelihood **P1c** for this work and also thank the WMAP and Planck team for making the data publicly available. The authors thanks Scott Dodelson for suggesting to compare PSO with the downhill simplex method and Gaurav Goswami for the feedback and comments on the drafts. The author would also like to thank Tarun Souradeep for giving insight about CMB physics time to time and introducing the problem of cosmological parameter estimation. All the computational work for this project was done on the IUCAA HPC facility. The author specially thank the Science and Engineering Research Board (SERB) of the Govt. of India for financially supporting this work via a Start-Up Research Grant (Young Scientists) with grant no SR/FTP/PS-102/2012.

- 
- [1] J. Prasad and T. Souradeep, Phys. Rev. D **85**, 123008 (2012), 1108.5600.  
[2] J. R. Bond, R. Crittenden, R. L. Davis, G. Efstathiou, and P. J. Steinhardt, Physical Review Letters **72**, 13 (1994), arXiv:astro-ph/9309041.  
[3] W. Hu and N. Sugiyama, Phys. Rev. D **51**, 2599 (1995), arXiv:astro-ph/9411008.  
[4] S. Dodelson and J. M. Jubas, Astrophys. J. **439**, 503 (1995), astro-ph/9308019.  
[5] G. Jungman, M. Kamionkowski, A. Kosowsky, and D. N. Spergel, Phys. Rev. D **54**, 1332 (1996), arXiv:astro-ph/9512139.  
[6] G. Jungman, M. Kamionkowski, A. Kosowsky, and D. N. Spergel, Physical Review Letters **76**, 1007 (1996), arXiv:astro-ph/9507080.  
[7] M. Zaldarriaga, D. N. Spergel, and U. Seljak, Astrophys. J. **488**, 1 (1997), arXiv:astro-ph/9702157.  
[8] J. R. Bond, G. Efstathiou, and M. Tegmark, Mon. Not. R. Astron. Soc. **291**, L33 (1997), arXiv:astro-ph/9702100.  
[9] J. R. Bond, A. H. Jaffe, and L. Knox, Phys. Rev. D **57**, 2117 (1998), arXiv:astro-ph/9708203.  
[10] A. Kosowsky, M. Milosavljevic, and R. Jimenez, Phys. Rev. D **66**, 063007 (2002), arXiv:astro-ph/0206014.  
[11] M. Chu, M. Kaplinghat, and L. Knox, Astrophys. J. **596**, 725 (2003), arXiv:astro-ph/0212466.  
[12] S. Dodelson, *Modern cosmology* (Academic Press, San Diego, U.S.A., 2003).  
[13] J. R. Bond and G. Efstathiou, Mon. Not. R. Astron. Soc. **226**, 655 (1987).  
[14] N. Christensen, R. Meyer, L. Knox, and B. Luey, Classical and Quantum Gravity **18**, 2677 (2001), arXiv:astro-ph/0103134.  
[15] A. Challinor, arXiv:astro-ph/0403344 (2004), arXiv:astro-ph/0403344.  
[16] S. Hamimeche and A. Lewis, Phys. Rev. D **77**, 103013 (2008), 0801.0554.  
[17] S. Hamimeche and A. Lewis, Phys. Rev. D **79**, 083012 (2009), 0902.0674.  
[18] J. R. Bond and G. Efstathiou, Astrophys. J. Lett. **285**,

- L45 (1984).
- [19] C.-P. Ma and E. Bertschinger, *Astrophys. J.* **455**, 7 (1995), astro-ph/9506072.
  - [20] W. Hu, U. Seljak, M. White, and M. Zaldarriaga, *Phys. Rev. D* **57**, 3290 (1998), astro-ph/9709066.
  - [21] U. Seljak and M. Zaldarriaga, *Astrophys. J.* **469**, 437 (1996), arXiv:astro-ph/9603033.
  - [22] U. Seljak and M. Zaldarriaga, CMBFAST: A microwave anisotropy code, 1999, 9909.004, Astrophysics Source Code Library.
  - [23] A. Lewis, A. Challinor, and A. Lasenby, *Astrophys. J.* **538**, 473 (2000), arXiv:astro-ph/9911177.
  - [24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
  - [25] W. K. Hasting, *Biometrika* **57**, 97 (1970).
  - [26] A. Lewis and S. Bridle, *Phys. Rev. D* **66**, 103511 (2002), arXiv:astro-ph/0205436.
  - [27] L. Verde *et al.*, *Astrophys. J. Suppl. Ser.* **148**, 195 (2003), arXiv:astro-ph/0302218.
  - [28] 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.
  - [29] S. Das and T. Souradeep, *JCAP* **7**, 18 (2014), 1403.1271.
  - [30] J. A. Nelder and R. Mead, *The Computer Journal* **7**, 308 (1965).
  - [31] J. Kennedy and R. C. Eberhart, IEEE International Conference on Neural Networks **4**, 1992 (1995).
  - [32] J. Kennedy and R. C. Eberhart, *Swarm Intelligence* (Morgan Kufmann, 2001).
  - [33] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications* (Springer, 2008).
  - [34] A. Lazinica, *Particle Swarm Optimization* (In-Tech., 2009).
  - [35] A. P. Engelbrecht, *Computational Intelligence, An Introduction* (John Wiley & Son, 2002).
  - [36] C. Skokos, K. E. Parsopoulos, P. A. Patsis, and M. N. Vrahatis, *Mon. Not. R. Astron. Soc.* **359**, 251 (2005), astro-ph/0502164.
  - [37] Y. Wang and S. D. Mohanty, *Phys. Rev. D* **81**, 063002 (2010), 1001.0923.
  - [38] A. Rogers and J. D. Fiege, *Astrophys. J.* **727**, 80 (2011), 1101.5803.
  - [39] L. Lentati *et al.*, *Phys. Rev. D* **87**, 104021 (2013), 1210.3578.
  - [40] <http://camb.info/>.
  - [41] <http://map.gsfc.nasa.gov/>.
  - [42] D. N. Spergel *et al.*, *Astrophys. J. Suppl. Ser.* **148**, 175 (2003), astro-ph/0302209.
  - [43] C. L. Bennett *et al.*, *Astrophys. J. Suppl. Ser.* **208**, 20 (2013), 1212.5225.
  - [44] E. Hivon *et al.*, *Astrophys. J.* **567**, 2 (2002), astro-ph/0105302.
  - [45] M. Tegmark, *Phys. Rev. D* **55**, 5895 (1997), astro-ph/9611174.
  - [46] H. K. Eriksen *et al.*, *Astrophys. J. Suppl. Ser.* **155**, 227 (2004), astro-ph/0407028.
  - [47] G. Hinshaw *et al.*, *Astrophys. J. Suppl. Ser.* **208**, 19 (2013), 1212.5226.
  - [48] Planck Collaboration *et al.*, *Astronomy & Astrophysics* **571**, A1 (2014), 1303.5062.
  - [49] [http://wiki.cosmos.esa.int/planckpla/index.php/Main\\_Page](http://wiki.cosmos.esa.int/planckpla/index.php/Main_Page).
  - [50] Planck Collaboration *et al.*, *Astronomy & Astrophysics* **571**, A15 (2014), 1303.5075.
  - [51] Planck Collaboration *et al.*, *Astronomy & Astrophysics* **571**, A16 (2014), 1303.5076.
  - [52] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in FORTRAN. The art of scientific computing* (Cambridge University Press, 1992).
  - [53] W. Forst and D. Hoffmann, *Optimization Theory and Practice* (Springer, New York, U.S.A., 2010).
  - [54] M. J. D. Powel, *The Computer Journal* **5**, 147 (1962).
  - [55] <http://mat.uc.pt/~zhang/software.html>.
  - [56] G. Goswami and J. Prasad, *Phys. Rev. D* **88**, 023522 (2013), 1303.4747.
  - [57] <http://cosmologist.info/cosmomc/>.