

Solving the CMBR Map Making Problem with a Stochastic Algorithm

Ricardo Varela Corrêa¹
Coordenadoria de Ciências Espaciais e
Atmosféricas
ricardo@cea.inpe.br
Thyrso Villela¹
Divisão de Astrofísica
villela@das.inpe.br
Airam Jonatas Preto¹
Laboratório Associado de Computação e
Matemática Aplicada
airam@lac.inpe.br

Stephan Stephany¹
Laboratório Associado de Computação e
Matemática Aplicada
stephan@lac.inpe.br
Carlos A Wuensche de Souza¹
Divisão de Astrofísica
alex@das.inpe.br
Jorge Mejia¹
Divisão de Astrofísica
mejia@das.inpe.br

¹Instituto Nacional de Pesquisas Espaciais - INPE

Abstract

The aim of this work is to study the feasibility of stochastic algorithms to produce Cosmic Microwave Background Radiation (CMBR) Anisotropy maps. In order to evaluate the proposed method, simulations for time-ordered data (TOD) with up to $O(10^3)$ and $O(10^5)$ measurements were performed. In this method, initial constraints were not imposed and the initial guess for the candidate solution was taken from an average of one hour data acquisition, instead of the standard approach of using a random initial guess. It is shown that the use of a simulated annealing algorithm can produce CMBR anisotropy maps in feasible processing time using a distributed memory parallel and a code implemented with calls to the communication library Message Passing Interface (MPI).

Keywords: Cosmic Microwave Background Radiation, simulated annealing, MPI, high performance cluster computing

1. Introduction

The process of data analysis for map construction of the Cosmic Microwave Background Radiation (CMBR) is a convenient way to represent distribution of the measured temperature of the sky[1]. The production of CMBR maps consists in reducing a large time-ordered data set (TOD) into a matrix that represents the sky temperature without any noise. The TOD series is composed by the chronological set of observations obtained by a CMBR experiment. Production of maps with high angular resolution is computationally challenging due to the performance requirements. Thus,

there is a need to develop efficient algorithms for the CMBR map production and further data analysis[2].

The aim of this work is to exploit a stochastic algorithms, the simulated annealing (SA) [3], to reduce the intrinsic noise of the TOD series for the Background Emission Anisotropy Scanning Telescope (BEAST) experiment[4]. Some standard methods employs the average temperature of the TOD for each pixel of the sky in order to obtain a lower computational complexity. Such schemes are called bin average methods. The use of a stochastic algorithm allows to employ all data of the TOD, aiming at improving the accuracy of the produced maps. Successive candidate solutions are generated and the noise vector is expected to decrease. Iterations stop when the noise reaches a near zero value. In our TOD, each region of the sky corresponds to a one hour sampling of the sky temperatures. For a given sky region, each TOD contains hourly samplings repeated every 24 hours. During an hour period, each pixel is sampled $s = 20,000$ and the considered region of the sky has $l = 8,000$ pixels. Synthetic sky temperature data was generated to simulate one TOD for a 30-day duration.

In a first attempt, a genetic algorithm (GA)[5] was used to generate an initial guess for the candidate solution. This solution was then further improved by a SA. The idea was to generate a global solution by the GA[6] and to improve it performing a local search by the SA. This approach produced a solution that was slightly better than the obtained with a bin average method, but with an increased processing time due to the GA. This led to the use of an initial guess taken from an one hour data pixel average. Iteration proceeds using only the SA, until convergence. Then, a re-annealing[7] scheme is applied to the solution in order to further reduce the intrinsic noise. This method produced good results in the CMBR map production using

simulated TOD data in a processing time slightly higher than the bin average method. It is important to emphasize that the proposed method obtained a better solution without reducing/averaging temperature data. Furthermore, this method was parallelized with calls to the Message Passing Interface (MPI) communication library[8] and executed in a distributed memory parallel machine[9], a cluster[10] of 16 PC's interconnected by a Fast Ethernet network.

This article is organized as follows. Section 2 is a brief description of a typical bin average method and gives a general idea of the mathematical formalism employed in CMBR map making. Section 3 describes the employed SA algorithm and Section 4 shows the results obtained with the proposed method.

2. A bin average method and the CMBR map making formalism

The production of a CMBR map from a large set of sky observations (TOD) allows a further analysis in order to estimate cosmological parameters that are of great interest in Cosmology. Each map is composed of a pixel pattern, pixels are valued according to the sky temperatures and are stored in a vector m that represents a possible solution (map). The vector of measured temperatures d is assumed as being linearly dependent on the sky map through a known pointing matrix $P(2)$. Thus, both d and m have dimension l and P has dimension $l \times l$. Measured temperatures d can be mapped to the sky map temperatures m by the following equation[11]:

$$d = P m + n \quad (1)$$

Where n is a vector that expresses the random noise. The use of the BEAST experiment data (3) implies in a particular arrangement for the P matrix. The i -th row of P represents a scan and has only one non zero element in a given j -th column with a value of 1, that corresponds to the observed pixel:

$$P_{ij} = \delta_{pi,j} \quad (2)$$

In the equation above, δ is the Kroenecker delta function and the subscript p denotes the temporal sequence of observed pixels. The map making technique usually employs the generalized least square approach. The best candidate solution (map) must minimize the following quantity:

$$\chi^2 = n^t N^{-1} n = (d^t - m^t P^t) N^{-1} (d - P m) \quad (3)$$

The desired map[11] is then given by the following linear estimator:

$$\tilde{m} = (P^t N^{-1} P)^{-1} P^t N^{-1} d \quad (4)$$

The noise inverse covariance matrix N^{-1} is estimated from the observed data. Given the large amount of data, the inversion of the system in Eq. (4) is computationally heavy [2].

Once the noise vector is considered to have zero mean, the bin average method is used to produce the desired temperature vector without requiring the inversion of the noise covariance matrix[12]. This method averages the temperatures obtained at each pixel, then averages are applied at each one-hour set of data for every pixel. The obtained vector of temperatures is averaged continuously considering weekly intervals for a total year. The resulting vector is then plotted as a CMBR map.

The bin average method employed and applied to the simulation data of the BEAST experiment, as in this work, requires the computation of the mean and the standard deviation of the s measurements for each of the l pixels of a particular TOD. For a given one-hour set of data that corresponds to the same sky coverage region, the bin average is performed as a weighted sum with using the standard deviation[13] for each pixel:

$$\mu_i = \sum_{i=0}^l \bar{x}_i \sigma_i^{-2} \left(\sum_{i=0}^l \sigma_i^{-2} \right)^{-1} \quad (5)$$

Where μ is the resulting average, \bar{x} is the average of the temperature vector at each pixel i and σ is the standard deviation for each pixel i .

The proposed method uses the above bin average estimation as an initial guess for the SA. This initial candidate solution is considered to be a global solution that is further refined by means of a local search performed by the SA. This process is described in the following section.

3. The employed simulated annealing based method

Simulate Annealing is a heuristic approach for optimization problems. The name derived from a metallurgy technique involving the heating of a material followed by its controlled cooling in order to achieve molecular stability. The heat causes the molecules to move from their position through states of energy. The slow cooling increase the probability of reaching a more stable, lower energy state in comparison to the initial state.

The SA algorithm associates an energy to each point of the search space according to its fitness and try to reach a minimum energy point. It starts at an arbitrary point of the search space. At every step, it randomly chooses a neighboring point and moves to that point according to a certain probability. This probability is a function of the energy difference between the two points and a global time-dependent parameter defined as the temperature of the system.

The pseudo-code below performs the simulated annealing algorithm[14] starting from an initial state S_0 and initial temperature T_0 . The *newstate* function provides a move to a new state S_i with a new temperature T_i . The *objective* function then computes the new energy E_i for the state S_i . The *fitprob* function returns a probability at which a transition to a state of higher energy takes place. The *schedule* function returns a new temperature at each iteration provided by a counter of the annealing schedule.

```

Initialize( $S_0, T_0$ )

 $E := \text{objective}(S_0)$ 
 $i := 0$ 

while not stop_condition

     $S_i := \text{newstate}(S, T)$ 
     $E_i := \text{objective}(S_i)$ 
    If [ $E_i < E$ ] or [ $\text{random}() < \text{fitprob}(E, T)$ ]
         $S := S_i$ 
         $E := E_i$ 
         $T := \text{schedule}(T, i)$ 
         $i := i + 1$ 

end while
end program

```

Figure 1. Pseudo-code of the SA

The pseudo-code presented in figure 1 denotes an initial energy state E_0 at temperature T_0 . A random perturbation is applied altering its energy state to E_1 . If this energy state is lower the transition is accepted. If it is higher, it might be accepted or not according to a probability function. The change of energy from E_0 to E_1 is performed by the *fitprop* function and the probability of acceptance of the new state is given by the Boltzmann probability distribution[15]:

$$p = \exp[-\Delta E / \kappa T] \quad (6)$$

A random number in the range $[0 < r < 1]$ is generated by the *random* function and compared to p . If it is greater, the new state is accepted. In the current work, the above distribution was replaced by a Cauchy-Lorentz distribution. This is known as the fast simulated annealing algorithm[16].

SA algorithms may be improved by including a re-annealing phase. This phase occurs when the evaluation of the objective function remains unchanged for a chosen number of iterations. In that case, the standard deviation of the current candidate solution is evaluated and used as a correction factor to all elements of this solution. The proposed SA implementation employs re-annealing.

An additional improvement is used in the current SA. As the algorithm stabilizes, i.e. converges to an optimal

solution, the standard deviation of the final solution is evaluated and employed to correct it. This standard deviation is given by:

$$\alpha_i = \left\langle \left| x_i - \mu_i \right| \right\rangle \text{ where } \mu_i = \langle x_i \rangle \quad (7)$$

Where x represents the temperature vector sampled (candidate solution) for each pixel i . The resultant vector η for each pixel i is then corrected by an empirical *factor* (in this work, taken as 0.4):

$$\eta_i = \mu_i * \text{factor} * \left(\frac{\sim}{m_i} - 1 \right) \quad (8)$$

The *stop_condition* in a SA algorithm can be given by a limit number of iterations or when the error decreases below a given threshold. This is the case of the current implementation. In addition, the algorithm was parallelized by means of calls to the Message Passing Interface (MPI) communication library in order to be executed in a distributed memory parallel machine.

The proposed SA employs an initial guess given by the bin average method was tested using a set of simulated data and its results are shown in the next section.

4. Numerical results with simulated data

In order to generate the simulated data, a full sky map was created using the SYNFAST routine of the HEALPIX package [17]. A smaller patch with 8,000 pixels was selected for the tests and corrupted with white Gaussian noise in a way to obtain a signal-to-noise ratio of 0.1. Each simulated TOD series is composed of 20,000 temperature values for each pixel, representing an one hour data acquisition. Figure 1 shows the first 50 values of temperatures of a TOD for a given pixel with and without noise.

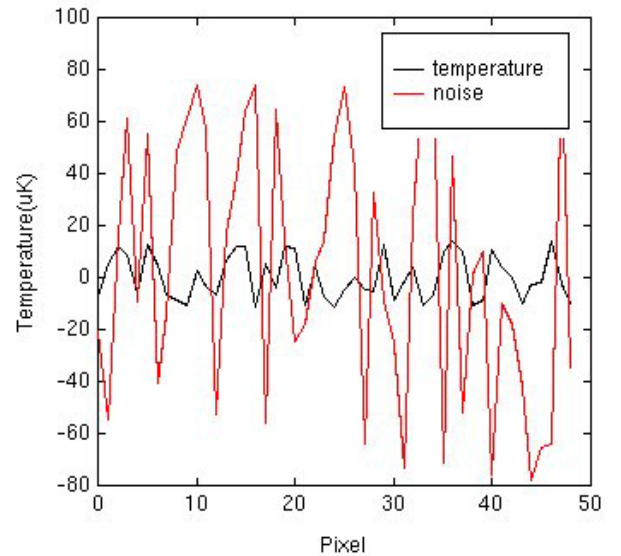


Figure 1. Temperature values with and without noise.

Initially, the SA method was tested for map making using a random initial guess solution. Figure 2 shows the convergence of the SA for this case using a set of 30 TODs in order to represent a 30-day acquisition period of a given region of the sky. It can be noted that more than 100 iterations of the SA were required.

On the other hand, Figure 3 shows the convergence when the bin average method was employed to generate the initial guess solution. The number of required iterations was about 40. It is interesting to note that it is faster to use this scheme to generate the initial guess solution than using a randomly generated one.

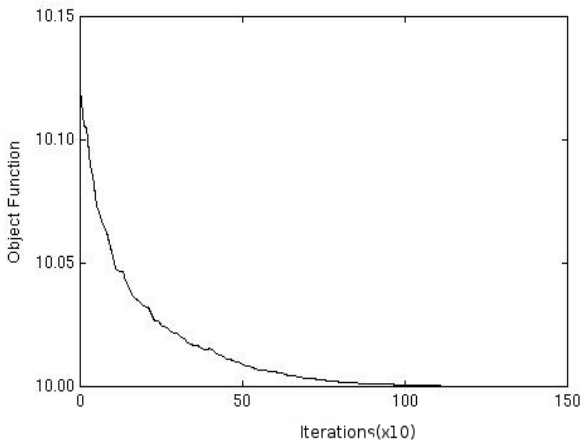


Figure 2. Convergence using a random vector as initial guess

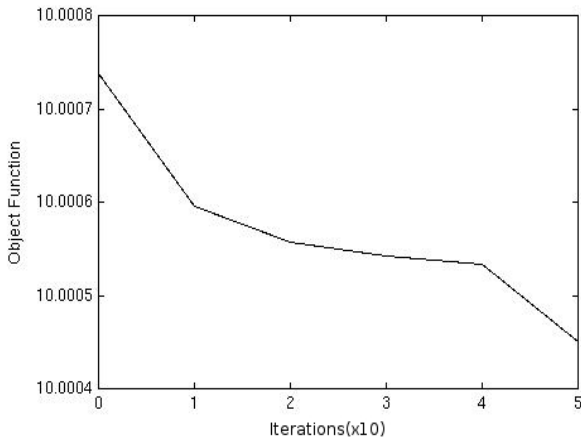


Figure 3. Convergence of SA using a data vector as initial guess

The inflexion at the 40th iteration shown in Figure 3 is caused by the re-annealing phase in the SA process after 10 iterations with no improvement of the solution.

Figure 4 shows a comparison of the standard deviations of the solution vector obtained for the two methods (the standard bin average method and the proposed SA method), for the first 100 pixels. However, this comparison is similar for the rest of the 8,000 pixels.

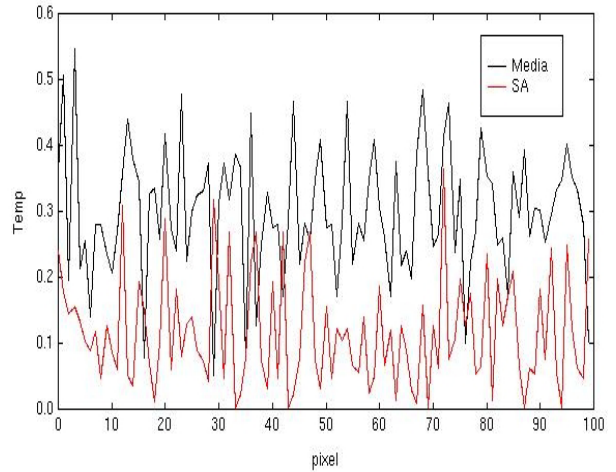


Figure 4. Standard deviation of the solution vector for the proposed and the bin average methods.

Test of the proposed parallel SA were executed in a distributed memory parallel machine, a cluster of 16 PC's with Pentium III 800 MHz and 256 MB of memory per node interconnected by a Fast Ethernet network.

5. Conclusions

A new method for CMBR map production is proposed using a stochastic algorithm, the SA. Numerical results were compared with a standard bin average method for simulated data that corresponds to a particular TOD series for a 30 day period. The proposed method is applied to the data analysis of the BEAST experiment.

White noise with a signal-to-noise ratio of 0.1 was applied to the simulated data in order to generate the TOD.

The use of the bin average method as a initial guess solution for the SA causes a faster convergence of the SA. Numerical results show a standard deviation lower (at least 35% for over 95% of the pixels) than the obtained by the standard bin average method. The Pearson correlation[12] of the solution with the initial simulated data is 0.999.

Although the time execution of the proposed method is slightly higher than that of the bin average method, the obtained solution is better. These simulations show that the proposed method solves the CMBR map making problem within a reasonable processing time using the SA algorithm in a parallel environment

6. References

- [1] Tegmark, M., "How to make maps from CMB without losing information" *Ap. J.*, 480, L87 (1997) (astro-ph/9611130).
- [2] Borrill, J. "The challenge of data analysis for future CMBR observations". 1999, *Phys. Rev. D*, 59, 027302
- [3] Kirpatrick, S.; Gellatt, C.D. & Vecchi, M.P. "Optimization by simulated annealing". *Science*, vol. 220, n. 4598. 1983, p.671-680.
- [4] Natoli, P. et al. "Imaging the Cosmic Microwave Background: The BEAST Experiment." *Cosmology and Particle Physics*, CAPP 2000, held 17-28 july, 2000. Verbier – Switzerland. AIP Conference Preceedings, vol.555. Melville, NY: American Institute of Physics, 2001, p.324.
- [5] Cantú-Paz, E. "A survey of parallel genetic algorithms". *IlligAL report n.97003*. University of Illinois, EUA, 1997, p.28.
- [6] Gefenstette, J.J. "Parallel adaptive algorithms for function optimization". Tech. Report n.C5-81-19. Vanderbilt University, EUA, 1981.
- [7] Ingber, L. "Very fast simulated reannealing". *J.Math. Comp. Modelling*, 12:967-973, 1989.
- [8] Pacheco, P.S. *Parallel programming with MPI*. Morgan Kaufmann Publishers, Inc. São Francisco, 1997.
- [9] Hwang, K.; Briggs, F.A. *Computer architecture and parallel processing*. McGraw-Hill Book Company, New York, 1984.
- [10] Spector, D.H. *Building Linux clusters*. O'Reilly & Associates, Sebastopol, 2000.
- [11] Meinhlod, P.R. et al. *A map of the Cosmic Microwave Background from the BEAST experiment*. Eprint arXiv:astro-ph/0302034 v1. 2003.
- [12] Dupac, X.; Giard, M. "Map-making methods for Cosmic Microwave Background experiments". *Mon.Not.Roy.Astron.Soc*, vol.330, 2002, p. 497.
- [13] Bevington, P.R. *Data reduction and error analysis for the physical sciences*. McGraw-Hill Book Company, New York, 1969.
- [14] Azencott, R. *Simulated Annealing: Parallelization Techniques*. John Willey & Sons, Inc. Canada. 1992.
- [15] Ingber, L. *Adaptive simulated annealing(ASA): Lessons learned*". *J. of Control and Cybernetics on Simulated Annealing Applied to Combinatorial Optimization*. 1995.
- [16] Szu, H. "Fast simulated annealing". In J.S. Dendekes, *AIP Conf. Proc. 151, Neural Networks for Computing*, Snowbird, Utah, AIP, 1986.
- [17] Gorski, E. K.; Hivon, E., & Wandelt, B. 1998, in *Proceedings of the MPA/ESO Garching Conference*, ed. A. J. Banday, K. Sheth, & L. Da Costa (Garching:ESO), inpress, <http://www.eso.org/science/healpix>