IUCAA-17/93

IUCAA -17 /93
June ' 93

Algorithm for Optimally Distributing quantised load on Transputers with Unequal Speeds: An Application to the Detection of Gravitational Wave Signals from Coalescing Binaries

18

By

Sangita N. Pitre and S.V. Dhurandhar

Fermilab Library
0 1160 0036988 8

FERMILAB
SEP 13 1993
LIBRARY

Inter-University Centre for Astronomy & Astrophysics

An Autonomous Institution of the University Grants Commission

Preprint Request (Please quote the preprint number)
email : preprn@iucaa.ernet.in
Fax : 0212 - 335760
Post : IUCAA, Post Bag 4, Ganeshkhind, Pune 411 007, India.

Algorithm for Optimally Distributing quantised load on Transputers with Unequal Speeds: An Application to the Detection of Gravitational Wave Signals from Coalescing Binaries

Sangita N. Pitre and S. V. Dhurandhar

IUCAA, Post Bag 4, Ganeshkhind, PUNE-411007, INDIA

June, 1993

Abstract

In a parallel computing system, we work with a network of a large number of processors wherein the performance characteristics each processor may have are different. This leads to a situation that when there is equal load on all the processors, some complete the job before the others. To make the optimum use of the available computing facilities and optimise on time, it is necessary to balance the load on the processors according to their characteristics like speed etc. Here we present an algorithm to optimise on 'time' when different processors have different speeds and the load is quantised in integral multiples of a given unit of load. The algorithm distributes the load in such a manner that all the processors work optimally and the processing time is minimal. The optimal distribution of the load is achieved by employing the well known bisection technique for finding the roots of an equation. We discuss this algorithm in the context of our application for filtering the coalescing binary gravitational wave signals. Numerical results are finally discussed for the 64 transputer machine (PARAM).

IUCAA - Preprint - 17 ; June. 1993

(Submitted to Parallel Computing)

1. Introduction: In a parallel environment, we work with a network of a large number of processors. In the network available to us, we worked with a cluster of 64 processors and the processors were transputers T800 from INMOS Ltd. In general, the transputers have different processing speeds. Our particular cluster consisted of 4 types of transputers as follows: T800 25 MHz with 3cycles for accessing external memory , T800 25 MHz with 4 cycles and T800 20 MHz with 3 and 4 cycles. Running an application software in minimum possible time which would help in an on-line data analysis is our main concern. We observed that due to the above described differences in the performance characteristics of transputers, even though equal work load is placed on all the transputers, some complete their task before the others. To make the optimum use of the available computing facilities and also to optimise on time, it thus became necessary that we come upon an algorithm which distributes the work load in such a way that as far as possible no processor is left idle and all of them finish the task almost at the same time. The algorithm basically puts more load on the faster processors and less on the slower ones. More specifically the load balancing algorithm described in this paper distributes the load on the processors in proportion to their speed. Since the load is quantised (discrete), to obtain an optimal distribution of the work load on the processors is a non-trivial task as we cannot have the load in exact proportion to the speed of the transputer.

Our application involves the filtering of gravitational wave signals from coalescing binary stars. A binary system consists of two stars which spiral together emitting gravitational waves and eventually coalescing with a burst of waves [1]. This is one of the most promising sources of gravitational waves for broad band detectors [2]. The signal depends on several parameters such as the masses of the stars, the phase, the time of arrival etc. A bank of filters is constructed which scans the astrophysically relevant range of parameters. The matched filtering technique is employed to filter out the signal from the noise [3, 4, 5]. This technique consists of correlating a data segment with every filter in the bank so that the correlation statistic exceeds a certain preset threshold. This scheme has been amply discussed in the literature. See [6] and references cited therein.

In a parallel network the scheme involves distributing the filters corresponding to

different values of the parameters on different processors. The tasks of computing correlations with several filters are independent of each other and the problem therefore is highly parallelizable. This parallel algorithm has been given in [7] which consists of distributing filters in *equal* numbers on each transputer[8]. The computation time involved in computing a given correlation is the same for every filter in the bank since all the filters are of equal length. This scheme produces the result in the shortest possible time if all the transputers have the *same* speed. However, if the transputers have *different* speeds, the faster among them will be left idle while the slower ones are still processing, thus resulting in a non-optimal situation. Hence the idea is to distribute the load (the number of filters) in such a manner that the task is performed in the least possible time by the network. If the load had not been quantised then the solution is trivially given by distributing the load in proportion to the speed of each transputer. Since only entire filters can be put on each transputer, the load on each transputer comes in integral multiples of a basic load (the load is quantised) and then the problem of distributing the load becomes involved.

In section 2, we first analyse and then describe the algorithm. In section 3 we describe our application in brief and then discuss the results obtained on the 64-node machine called PARAM available to us at the Centre for the Development of Advanced computing (C-DAC), Pune, INDIA.

2. The Load Balancing Algorithm

A. The Mathematical Analysis

We define the parameters used in the analysis as follows:

- N is the total number of transputers used which is in powers of 2.
- SP(i)=Speed of the i'th transputer where speed refers to the reciprocal of the time taken to process one filter or the number of filters processed in 1 second;
- F_TR(i) is the number of filters on the i'th transputer;
- T(i) is the time required by the i'th transputer to process it's work load which is F_TR(i) filters;
- N_TOT is the total number of filters to be distributed on N transputers;

The speed array and N_TOT are given. The aim is to find a distribution F_TR(i) such that the max T(i), where $1 \leq i \leq N$, is the least. We have the following relations between the quantities defined above:

The time taken by the i'th transputer is given by the load on it divided by the speed of the i'th transputer. This gives the relation,

$$\frac{F_TR(i)}{SP(i)} = T(i) \tag{2.1}$$

We also have the constraint that the total number of filters should add up to N_TOT. Thus

$$\sum_{i=1}^{N} F_TR(i) = N_TOT \tag{2.2}$$

Actually the load on each transputer should be in exact proportion to it's speed. This condition optimises the time since no transputer will remain idle. But owing to the quantised nature of the load, this cannot always be done. In the ideal case,

$$F_TR(i) = \alpha SP(i) \tag{2.3}$$

where α is a constant. From equation (2.2) we get,

$$\alpha = \frac{N_TOT}{\sum_{i=1}^{N} SP(i)} = \alpha_{min}(\text{say}) \tag{2.4}$$

We call this value of α as α_{min} since this is the minimum time needed to process N_TOT filters with a total speed of $\Sigma SP(i)$ which is the speed of the entire network. In a realistic situation wherein the load comes in multiples of a given load, equation(2.3) is not in general satisfied. In the general case equation (2.3) is modified to

$$F_TR(i) = [\alpha SP(i)] \tag{2.5}$$

where $[a]$ denotes the integer part of a i.e. the largest integer $\leq a$. Therefore if we now choose $\alpha = \alpha_{min}$ in equation(2.5) then $\Sigma F_TR(i)$ will not add up to N_TOT but will in general be less than it. To get the total number of filters at least as much as N_TOT or

more, a value of $\alpha > \alpha_{\min}$ will have to be chosen. The aim is to choose that value of α which is the lowest and also satisfies the following condition:

$$\sum_{i=1}^{N} F_TR(i) \geq N_TOT$$

Equation (2.5) then determines the distribution of the filters. In general, the total number of filters will add up to a number greater than N_TOT. However, if the exact number N_TOT is required then a few filters may be removed arbitrarily from the distribution to satisfy this constraint. This will not in any way reduce the maximum time taken by the network to process the load. In our particular application, we let the extra number of filters remain since these can be used to make the filter bank finer or increase the range of parameters, without paying any extra cost in time. This can be explained as follows: If the total number of filters to be distributed are N_TOT and after applying this algorithm we get the sum of the filters to be N_F which is more than N_TOT, it means that the computation time will remain the same for both N_F and N_TOT number of filters. In our particular application, we can use this fact to our advantage.

In the following , we prove that such a value of α exists and can be found by elementary means. Let,

$$N_I(i;\alpha) = [\alpha SP(i)] \tag{2.6a}$$

$$N_F(\alpha) = \sum_{i=1}^{N} N_I(i,\alpha) \tag{2.6b}$$

For a given α, $N_I(i,\alpha)$ is the number of filters on transputer i and $N_F(\alpha)$ the total number of filters. We immediately have the inequalities,

$$N_I(i;\alpha) \leq \alpha SP(i) \tag{2.7a}$$

$$N_F(\alpha) \leq \alpha \sum_{i=1}^{N} SP(i) \tag{2.7b}$$

Clearly from equation (2.4)

$$N_F(\alpha_{\min}) \leq N_TOT \tag{2.8}$$

We now get bounds on the optimal value of α. There is already a lower bound $\alpha_{\min}$. We now display an upper bound. From equation (2.6 a) we have the following relation:

$$(\alpha \mathrm{SP(i)} - 1) < \mathrm{N_I(i;}\alpha) \leq \alpha.\mathrm{SP(i)} \tag{2.9}$$

Summing

$$\alpha \Sigma \mathrm{SP(i)} - \mathrm{N} < \Sigma \mathrm{N_I(i;}\alpha) \leq \alpha \Sigma \mathrm{SP(i)}$$

From equation (2.4) we have

$$\frac{\alpha}{\alpha_{\min}}(\mathrm{N_TOT}) - \mathrm{N} < \mathrm{N_F}(\alpha) \leq \frac{\alpha}{\alpha_{\min}}(\mathrm{N_TOT}) \tag{2.10}$$

The first inequality produces an upper bound for α. If we choose an α such that

$$\frac{\alpha}{\alpha_{\min}}(\mathrm{N_TOT}) - \mathrm{N} > \mathrm{N_TOT} \tag{2.11}$$

then automatically $\mathrm{N_F}(\alpha) > \mathrm{N_TOT}$. Solving for α we obtain

$$\alpha_{\max} = \alpha_{\min}(1 + \frac{\mathrm{N}}{\mathrm{N_TOT}}) \tag{2.12}$$

The optimum value of α will lie in the closed-open interval $[\alpha_{\min}, \alpha_{\max})$. The closed-open interval for α is the direct consequence of the fact that the function f(x)=[x], where [ ] denotes the integer part of x is an upper semi-continuous function i.e.

$$\lim_{x \to x_{0+}} f(x) = f(x_0)$$
$$\lim_{x \to x_{0-}} f(x) \neq f(x_0) \text{ if } x_0 \text{ is an integer}$$

(Since if x_0 is an integer $f(x_0) = x_0$ but $f(x_0 - \epsilon) = x_0 - 1$ for $\epsilon < 1$) i.e. f is continuous from above but not from below. We observe that although α can be made to vary continuously, $\mathrm{N_F}(\alpha)$ is a discontinuous function and varies in steps i.e. $\mathrm{N_F}(\alpha)$ is a step function of α, with the following properties:

(i) $\mathrm{N_F}(\alpha)$ is an increasing function of α.

(ii) The steps are of size at most N (the number of transputers). This is clear from the definition.

(iii) N_F(α) retains a particular constant value over a closed open interval.

These simple properties of N_F(α) enables us to find easily the optimum value of α(say) α_{opt}, by employing the well-known bisection routine which always converges to a root. [Figure (i) depicts the function N_F(α) as it appears in our application of Signal Analysis.] Given a value of N_TOT, the optimum value of α say α_{opt} is found as follows: Consider that value of N_F(α) which satisfies the inequality N_F(α) $\geq$ N_TOT but is the least among such N_F(α). This value of N_F(α) will correspond to an closed open interval say, $[\alpha_1, \alpha_2)$ then the optimal value of α is the least in this interval i.e. $\alpha_{opt} = \alpha_1$. Clearly α_{opt} is the optimal value because if we consider an $\alpha < \alpha_{opt}$, α will lie in the preceding interval say $[\alpha_0, \alpha_1)$, which will correspond to a lower value of N_F(α) and which will be strictly less than N_TOT.

B. The Algorithm

We give below the steps in the algorithm. The steps are also displayed in the flow chart in figure(2). This is just the bisection method used to obtain α_{opt}.

Step 1. Compute α_{min} and α_{max}. Set

$$\alpha_{\text{min}} = \frac{\text{N_TOT}}{\text{SUM}}$$

where

$$\text{SUM} = \sum_{\text{i=1}}^{\text{N}} \text{SP(i)}$$

and

$$\alpha_{\text{max}} = \alpha_{\text{min}}(1 + \frac{\text{N}}{\text{N_TOT}})$$

Step 2. Set $\alpha_{\text{min}} = \text{A1}$ and $\alpha_{\text{max}} = \text{A2}$

Step 3. Find A3 which is the mid-point of the interval defined by A1 and A2.

$$\text{A3} = \frac{\text{A1} + \text{A2}}{2}$$

If N_F(A3) < N_TOT then A1 $\leftarrow$ A3, otherwise A2 $\leftarrow$ A3.

Step 4. Continue this process of bisection till the difference between the two limits becomes less than a preassigned number ϵ. This process must converge after n steps where

$$n \sim \log_2\left(\frac{\alpha_{max} - \alpha_{min}}{\epsilon}\right)$$

The value of A3 after n steps is α_{opt}.

Step 5. Find the distribution of filters corresponding to this value of α_{opt}

$$N_F(i; \alpha_{opt}) = [\alpha_{opt} SP(i)]$$

$$\Sigma N_I(i; \alpha_{opt}) = N_F(\alpha_{opt})$$

Step 6. $N_F(\alpha_{opt})$ is generally a little more than N_TOT. Remove $N_F(\alpha_{opt}) - N_TOT$ filters arbitrarily if necessary to obtain the distribution F_TR(i).

Although the final solution is not unique, the time is optimal. We tested this program for different values of α_{max} and for different speed profiles and it gave the optimal distribution of filters. The bisection routine converges to the correct value of α_{opt} within a few iterations.

3.The Numerical Experiment

A.Application to gravitational wave signals from coalescing binaries

The above algorithm is applied to the problem of signal detection in the context of gravitational wave signals from coalescing binary stars. After Einstein predicted gravitational waves from his General theory of Relativity in 1916, it is only now with the advance in technology that it has become possible to detect these weak waves. Indirect evidence exists for the waves, in that, the decay of the orbit of the binary pulsar PSR 1913+16 is exactly as predicted by the General theory of Relativity[9]. Their direct detection on Earth however needs large scale interferometers, prototypes of which exist and several full scale ones are under way around the globe(The LIGO, VIRGO projects [10], [11]).

Coalescing binary systems are one of the most promising sources for the detection of gravitational waves for these broadband detectors. A compact coalescing binary consists of two stars, typically neutron stars or blackholes, which orbit around each other bound by their mutual gravitational attraction. The General Theory of Relativity predicts that such a system should radiate energy in the form of gravitational waves. They lose energy thus and spiral towards each other until they coalesce. The nature of the gravitational waves emitted by the system has a very characteristic waveform-the so called 'chirp' waveform. The important point here is that the waveform can be predicted with a good accuracy. The signal is however, buried in the noise. The matched filtering technique is a powerful tool for detecting known signals in noisy data. The idea is to construct a template which looks something like the signal (it looks exactly like the signal in case of white noise) and correlate it with the data containing the signal. The correlation then displays a peak which can be taken as an evidence of detection if the correlation peak exceeds a certain preset threshold determined from the distribution of the noise and the event rate.

Athough the technique is easily generalized to coloured noise, here for simplicity we consider only white noise. Then the filters and the signal are identical. The wave form from such a system of total mass M and reduced mass μ located at a distance r is given

by :

$$h_+(t) \equiv h(t) = N_h a(t)^{-\frac{1}{4}} \cos\left(2\pi \int_{t_a}^{t} f(t')\, dt' + \Phi\right), \tag{3.1}$$

where the quantities appearing above are defined as follows:

- t_a and Φ are respectively, the time-of-arrival and the phase of the signal when the instantaneous gravitational wavefrequency of the signal reaches some fiducial frequency, say f_a which is the lowest frequency in the bandwidth of the detector and is taken to be 100 Hz.

- $a(t)$ is the time-dependent normalised distance between the stars (normalised to $a(t_a) = 1$),

$$a(t) = \left(1 - \frac{t - t_a}{\xi}\right)^{\frac{1}{4}} \tag{3.2}$$

- $f(t)$ is the instantaneous gravitational wavefrequency given by

$$f(t) = f_a \left[1 - \frac{(t - t_a)}{\xi}\right]^{-3/8} \tag{3.3}$$

- ξ is the time taken for the two stars to theoretically coalesce starting from a time when the instantaneous frequency is f_a,

$$\xi = 3.00 \left[\frac{M}{M_\odot}\right]^{-5/3} \left[\frac{f_a}{100 \text{ Hz}}\right]^{-8/3} \text{sec.} \tag{3.4}$$

- $\mathcal{M} = (\mu^3 M^2)^{1/5}$ is called the mass parameter; the Newtonian waveform depends only on this parameter instead of the two individual masses of the stars.

- The constant N_h is given by,

$$N_h = 2.57 \times 10^{-23} \left[\frac{\xi}{3 \text{ sec}}\right]^{-1} \left[\frac{f_a}{100 \text{ Hz}}\right]^{-2} \left[\frac{r}{100 \text{ Mpc}}\right]^{-1} \tag{3.5}$$

Even though the waveforms of coalescing binaries are known, the experimenter will not know before hand what the values of the parameters are. The method followed consists of constructing a bank of filters which scans the astrophysically relevant range of parameters and correlating a data segment with each of these filters.

The Newtonian signal depends on the following three parameters:

1. the mass parameter which is a certain combination of the masses of the two stars in the binary or equivalently the coalescence time ξ which is the time taken for the two stars to coalesce starting from a certain fixed frequency ;

2. the phase ϕ of the signal;

3. the time at which the signal arrives t_a.

The statistic is the cross correlation function which is crucial in the construction of a lattice of filters. The filters are constructed for a discrete set of the values of the parameters while the actual signal could have any values for these parameters. In general, the values of the parameters of the signal differ from those of every filter in the set. However, if the lattice consists of a large number of closely spaced filters then the cross correlation between the signal and some filter(i) will be significant and will cross the threshold if the amplitude of the signal is sufficiently high. The time of arrival is determined when the correlation peaks. This leaves just the parameters ξ and ϕ. In general, one would have had to consider a 2 dimensional lattice of filters in (ξ, ϕ) space. However, this is not necessary. The simple dependence of the signal on the phase makes it possible to construct a 2 dimensional basis in ϕ space. A waveform $q(t, \xi, \phi)$ with an arbitrary phase ϕ can be written as a linear combination of two waveforms, one with phase equal to 0 and the other with phase equal to $\frac{\pi}{2}$. It is thus sufficient to construct a lattice of filters, two for each value of the mass parameter corresponding to the two independent values of the phase. The sampling is carried out at at least twice the frequency of the highest frequency of the signal and it is important to note that all the filters have equal length and thus require identical number of operations for their processing. Typically the masses of the stars can range from a fraction of the sun's mass to 10 solar masses. In the ξ parameter this corresponds to 0.1 secs to a few secs. The phase ranges from 0 to 2π. The filters are uniformly distributed in the ξ space and typically the number of filters is about 500 to 1000. We also pad the filters with zeros[4] to at least 75%. This means the filter is at least four times larger than the waveform. The correlation statistic is normalized to unity and the threshold is taken about 25% below this value. The threshold is normally chosen 7σ where σ is the standard deviation of the noise assumed to be Gaussian. See [6,7] for details.

B. The Numerical Results

The Procedure : We used a cluster of 64 transputers to run our program . We had two master programs and one slave program. A copy of the slave program was sent to each of the 64 transputers while the two master copies sat on the root transputer. The master1 program collects the results and checks the correlation value with that of the threshold while the master2 sends the data to all the workers. First the set of filters is generated with the appropriate values of parameters on each transputer(according to the number of filters on each transputer). The Fourier transform of the filters is taken and kept ready. Méanwhile, the program master2 finds the Fourier transform of the data segment. This data is then sent to all the slaves and the clock starts counting the processing time. The clock is stopped when the slaves send back the results. The parameters given as input to the program are as follows:

- $\xi_i = 4.00$ $(\mathcal{M}_i = 0.841 M_\odot)$ Here $\mathcal{M}_i$ is the initial mass parameter and ξ_i is the time left till coalescence starting from the time when the the frequency of the gravitational wave f_a and related to the mass parameter by $\xi \propto \mathcal{M}^{-5/3}$. $M_\odot$ is the mass of the Sun $\sim 2 \times 10^{33}$ gms.

- $\xi_f = 0.16$ which corresponds to $\mathcal{M} = 5.62 M_\odot$.

- $\Delta\xi = 0.02$secs. This is the spacing between the two filters in the ξ space.

- Threshold= 0.755 (The maximum value of the correlation has been normalised to unity)

- $N_p = 32768$, (the number of data points for a 16 second data train sampled at 2KHz)

- N_TOT= 384 (The total number of filters to be distributed)

1. The time taken by each transputer for processing a fixed work load (to find the correlation of the data train with 6 filters each) was recorded along with the node id. The speed array has been tabulated below:

Table 1		
Node id	Time	SP(i)
1–27	25	(6/25)=0.24
28–35	20	(6/20)=0.30
36–47	25	(6/25)=0.24
48–51	17	(6/17)=0.35
52–64	25	(6/25)=0.24

The cluster consisted of transputers with three different speeds.

2. The above calculated speed array was given to the program which calculated the following distribution of filters per node id. The basic unit of load here is two filters corresponding to the two values of phase. Hence only even number of filters appear on each node id. Hence N_TOT= 384/2= 192.

The input values for the bisection routine were as follows:

$\alpha_{\min} = 11.7851$

$\alpha_{\max} = 15.7134$

$\alpha_{\text{opt}} = 12.5006$

The total number of the distributed filters corresponding to this value of α_{opt} is N_F(α_{opt}) =196.

Nodeid	Number of filters
1–47	6
48–51	8
52–64	6

3. The comparison of computation times is as follows (here the computation time refers to the maximum time taken by a transputer to process the load on it):

Load Placed	Time (secs)
With equal load placed on all transputers (N_TOT = 192)	25
With load as per the load balancing algorithm (N_F = 196)	25

We would like to add that in the second case, the filters being processed totally are 196 while in the first case the total number of filters is 192. Thus the average time for processing one filter has reduced after load balancing.

Conclusion: We observe that the load balancing allows us to use the network more efficiently and process more load in the given amount of time alloted to the network. We have demonstrated this fact in our application. If the load for the transputer consists of more quanta then better efficiencies can be obtained since then it would be possible to distribute the load more 'evenly', meaning thereby that the time taken by each transputer to process the load would be more or less equal. For example, if we put 10 units of load on the slowest 52 transputers, 12 units on the transputers 28-35 and 15 units on 48-51, the time taken by the network would be ideal and no processor would remain idle.

We also note that the kernel of the algorithm can be applied to other problems which have different sets of quanta of load but the speeds of the processors are the same. It is basically the ratio of the load per processor divided by the speed which is the crucial variable in the algorithm.

Acknowledgement

One of us (SNP) thanks C-DAC for research assistantship. We would like to thank C-DAC for the use of their machine PARAM for our numerical experiment.

References

[1] P. C. Peters and J. Matthews, Phys. Rev. **131**, 435, (1963)

[2] K. S. Thorne, in *300 Years of Gravitation*, edited by S. W. Hawking and W. Israel (Cambridge University Press, Cambridge, England, 1987).

[3] C. W. Helstrom, *Statistical Theory of Signal Detection*, 2nd ed, (Pergamon Press, London, 1968)

[4] Problems concerning analysis of data from gravitational wave detectors is discussed at length in: B. F. Schutz, in *The Detection of Gravitational Radiation*, edited by D. G. Blair (Cambridge University Press, England, 1989).

[5] B. F. Schutz, in *Gravitational Wave data Analysis*, edited by B. F. Schutz, Kluwer, Dordrecht, 1989).

[6] B. S. Sathyaprakash and S. V. Dhurandhar, Phys. Rev. D, **44**, 3819 (1991)

[7] B. S. Sathyaprakash and S. V. Dhurandhar, *A parallel algorithm for filtering gravitational waves from coalescing binaries.* (to appear in *The Journal of Computational Physics*)

[8] J. Watkins *A Prototype System for Gravitational Wave Data Analysis*, Ph.D Thesis, University of Wales, 1991.

[9] T. Damour and J. Taylor, Astrophys. J. **366**, 501 (1991).

[10] LIGO: The Laser Interferometer Gravitational Wave Observatory. Science, **256**, (1992)

[11] A. Brillet et al, VIRGO Final Conceptual Design, INFN, Pisa, 1992, Australian International Gravitational Research Centre Proposal, The University of Western Australia, 1990

Figure Captions

Figure (1). The figure shows the plot of N_F(α) verses α. N_F(α) is a step function and increases in a discontinuous manner with α. In the figure N_F takes the values 140, 144, 196, 204, 208 over the intervals [11.000, 11.334), [11.334, 12.5006), [12.5006, 13.3339), [13.3339, 14.1673) respectively. For $\alpha = 12.50$, the sum N_F suddenly jumps to 196 which is the correct total in this numerical experiment. We wanted to distribute 192 filters. We make use of the extra 4 filters here to make our bank of filters finer or to increase the range of the parameters.

Figure (2). The figure shows the flowchart of the procedure described in section 2 (B).

Figure 1

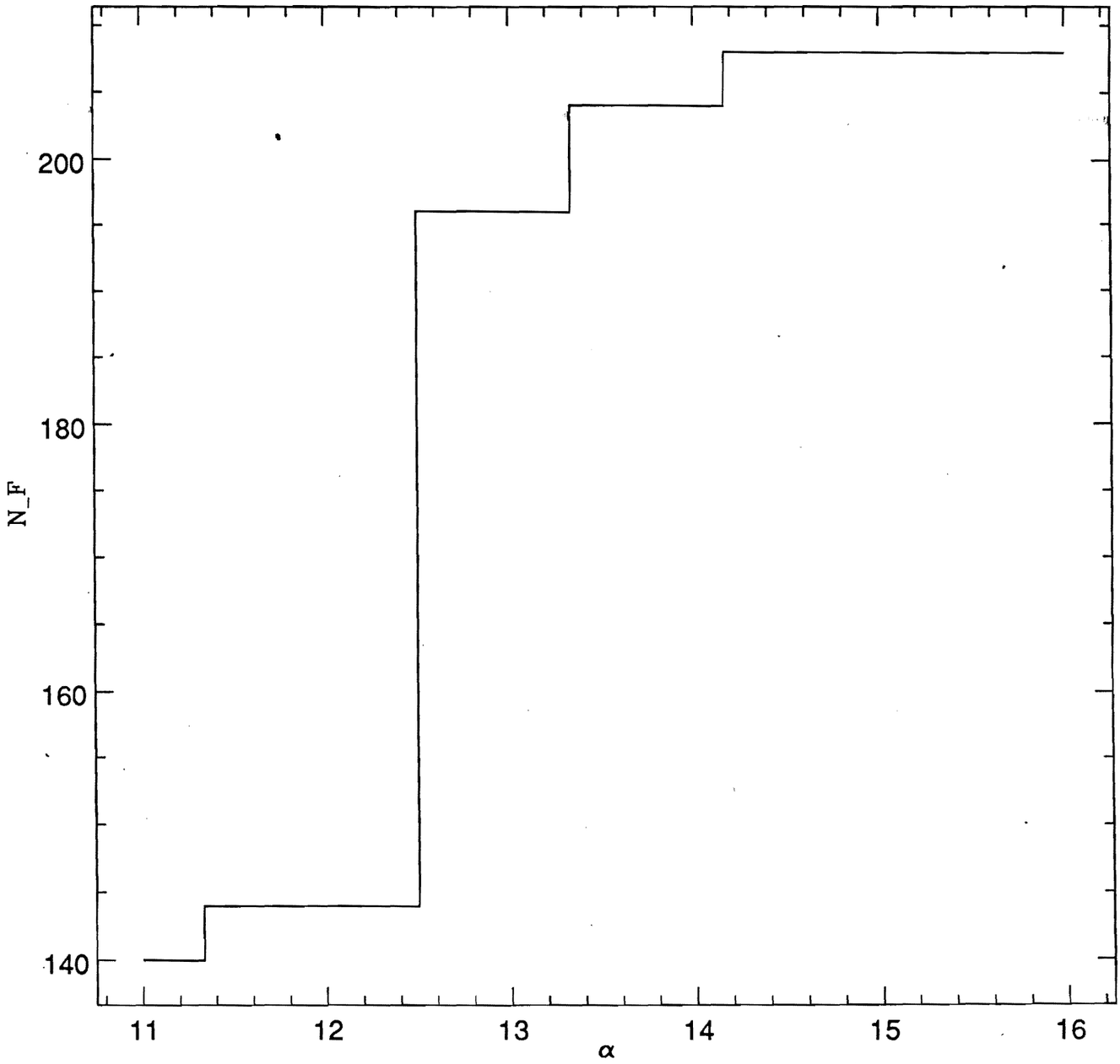
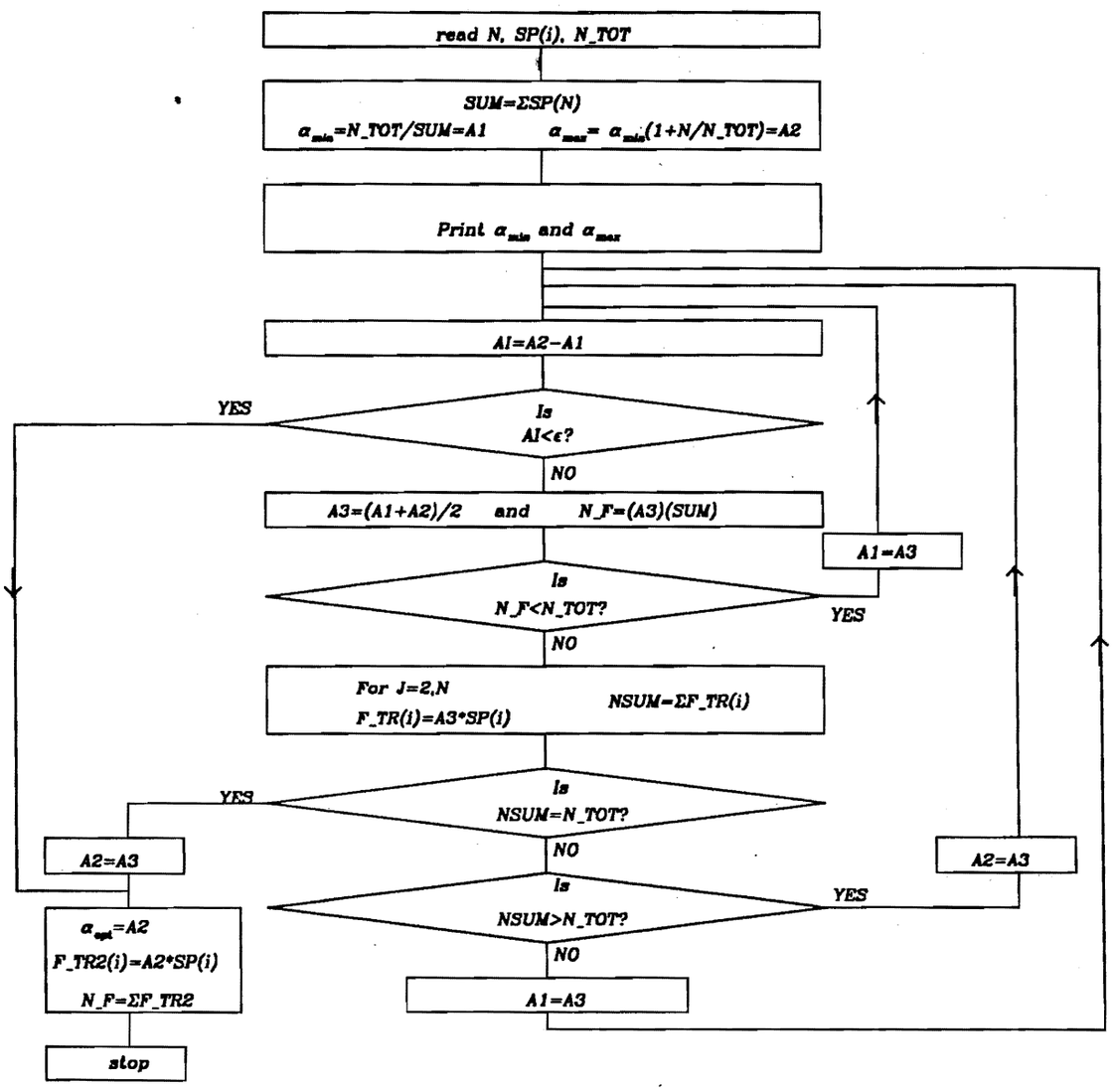


Figure 2.


read N, SP(i), N_TOT
SUM=ΣSP(N)
α_min=N_TOT/SUM=A1
α_max= α_min(1+N/N_TOT)=A2
Print α_min and α_max
AI=A2−A1
Is AI<ε?
YES
NO
A3=(A1+A2)/2 and N_F=(A3)(SUM)
A1=A3
Is N_F<N_TOT?
YES
NO
For J=2,N
F_TR(i)=A3*SP(i)
NSUM=ΣF_TR(i)
Is NSUM=N_TOT?
YES
NO
A2=A3
Is NSUM>N_TOT?
YES
NO
A2=A3
α_opt=A2
F_TR2(i)=A2*SP(i)
N_F=ΣF_TR2
A1=A3
stop