CUSTOMIZED COMPRESSION ALGORITHMS FOR THE SCIENTIFIC PAYLOAD OF GAIA

Master Thesis of: Alberto González Villafranca
Advisors: Enrique García-Berro Montilla, Jordi Portell i de Mora
IEEC – UPC – UB

Abstract

Gaia is the new astrometric mission of the European Space Agency. It will measure the positions and proper motions of more than one billion stars and other objects with unprecedented accuracy, providing a sample of more than 1% of the stellar content of our Galaxy. Such a mission implies large technological and design efforts, since it will have to detect, select and measure hundreds of stars every second, sending their data to the Earth – more than 1.5 million kilometers away [(1)]. Thus, the data transmission system must be highly optimized in order to make an efficient use of the downlink. We have focused the master thesis on this aspect; more specifically, we have revised and optimised the existing pre-compressing algorithms of the different instruments. Also different compression methods are tested in order to increase the final compression ratio. Our main goal is to guarantee the correct transmission of the highest amount of instrument data to the ground station. Therefore, the final ratio is the key factor that shall be analysed here, but CPU consumption and transmission reliability shall be taken into account as well.

1. Introduction

1.1 The Gaia mission

Gaia is the most ambitious astrometric space mission currently envisaged, adopted within the scientific program of the European Space Agency (ESA) in October 2000. It aims to measure the positions and proper motions of an extremely large number of stars and other types of objects with unprecedented accuracy, complemented with multi-band photometry and spectrometry. As a result, the most complete and accurate three-dimensional map of our Galaxy will be obtained, also including Solar System objects and extragalactic sources. This space observatory, illustrated in Fig. 1, will be a technological challenge in all its aspects, from its instrumentation and on-board data handling to the on-ground data processing and analysis.

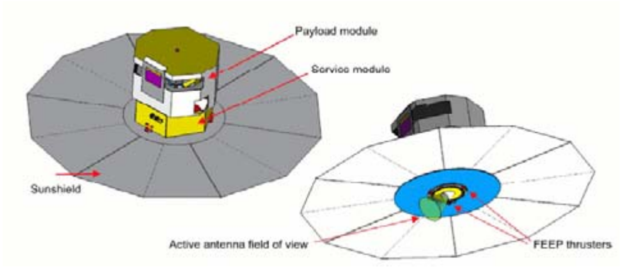


Figure 1: Gaia structure

The operation of Gaia is based on a continuous all-sky scanning. The satellite will spin around its own axis, with this axis performing a precession motion keeping a fixed angle with respect to the Sun. Although every astronomical object measurable by Gaia will be observed several times during the mission, this scanning law will lead to non-uniform sky coverage. The observations will be made in the visible spectrum, with two telescopes performing the astrometric and broad-band photometric measurements and a third telescope in charge of the spectrometric and medium-band photometric measurements.

The telemetry requirements of the Gaia mission are quite strict, since the instruments on-board will generate large amounts of information. The state-of-the-art CCD focal plane that will be used in the mission has almost 1 Gigapixel and will continuously operate in Time Delayed Integration (TDI) mode. Despite of the windowing mode (reading only the useful information for each star), the need of some sort of data compression algorithm is unavoidable. Furthermore, CCD output must be pre-processed before being compressed in order to achieve an adequate compression ratio, later feeding the scientific data into the telemetry stream. The data link is designed to transmit 3 Mbps while having visibility with the ground station. However, Gaia will only be visible about 8 hours per day, and thus the average transmission rate will be of about 1 Mbps. Furthermore the available bandwidth must be occupied also for attitude and housekeeping data, tightening even more the constraints for the scientific data.

1.2 The purpose

In this report we present the last improvements done to the pre-compression algorithms of some of the data generated by the scientific payload of Gaia. Some of the ideas presented here were suggested in previous studies [(2)]. Thus, the purpose of this thesis is to revise them and propose more elaborated algorithms for the pre-compression process, and also to explain the results obtained so far. Thus, the starting point of this note is the previous pre-compression algorithm already explained in those studies [(2),(3)].

The basis of the problem to be treated here has to do with the high amount of data generated by Gaia. It has already been shown in previous documents that it is not possible to feed it into the limited downlink [(4)] and the use of compression methods is mandatory. However, standard compression techniques alone are not enough. This problem has already been studied and some possible solutions have been proposed. In particular, the most feasible one seems to be the use of a pre-compression algorithm which consists of reorganizing the scientific data followed by some redundancy elimination. After this, the application of standard compression techniques further compresses the amount of telemetry. As this basic configuration seems to work fairly well, this approach will remain unchanged, although some improvements in the pre-compression algorithm can still be done.

Actual calculations show that a compression ratio of at least 2.8 is needed to never surpass the maximum downlink capacity. The situation can become risky when having both focal planes pointing towards high star density regions such as the galactic plane or the bulb. In these regions, the data rate gets naturally increased in order to transmit the information of all the stars measured in the focal plane. The Gaia payload has an on-board mass storage to deal with the data generated during slightly more than one day. It will keep the data until the ground station becomes available. Then all the data stored will be transmitted to the ground and the buffer will be filled with the new data until the following contact. Thus, high compression rates must be attained to preserve information integrity and avoid losses.

We must note that the Gaia design described here was upgraded on March 2006 with some changes to the instrumentation. Although we have done our study with such deprecated design, all of the ideas presented here are directly applicable to the latest design of the mission.

2. New adjustments to the pre-compression algorithms

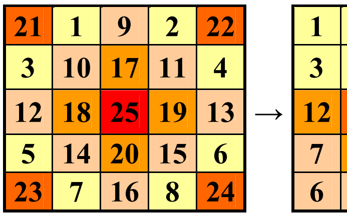
Here we propose some improvements for better pre-compressing the scientific data. The pre-compressing algorithm is lossless, in accordance with the mission requirements.

2.1. Offset correction

One of the points overlooked in the previous work was the difference between software and hardware binning. Each of the CCD samples read has an offset of 100 electrons to avoid negative values due to the read-out noise. The acquisition windows are two-dimensional, with pixel rows and columns, but in the end only one-dimensional samples (along-scan) are obtained in most cases since this is the most useful information that will be needed for the astrometric instruments of Gaia (ASM, AF & BBP) to estimate the position of every object. Thus, sets of pixels will be grouped in samples. The sample grouping depends on the instrument and window mode and there are two different ways of grouping them: by hardware or by software[5]. The main characteristic of the hardware binning is that the offset of the whole structure remains unchanged in 100 electrons. However, in the case in which software binning is performed the proximity electronics read the pixel values and outputs the total addition. This means that the offset of all the pixels has been summed. Having samples with different offsets could be a potential problem for data compression so they must be corrected.

2.2. ASM patch codification change

The Astrometric Sky Mapper (ASM) instrument is located at the beginning of the CCD and it is the first region where objects of both Fields of View (FOV) are focalised. There are two rows of ASM, each one dedicated to only one of the FOVs, but their operation is otherwise identical. They are read entirely, contrary to what happens in the AF and BBP instruments. When an object reaches the end of the ASM CCD it is detected and windowed. This ASM window data is one of the fields to be sent to the star packet (SP) and, usually, presents a Gaussian-like distribution pattern, centred in the right middle of the window: it has a maximum in its centre and vanishes towards the window edges. This patch is differentially coded and, consequently, it is mandatory to minimise the differences between the successive samples to be coded. It has been previously proposed [2] a codification scheme for these patches, but it did not take into account the offset corrections and some other details.



21	1	9	2	22
3	10	17	11	4
12	18	25	19	13
5	14	20	15	6
23	7	16	8	24

→

1	2	13	8	5
3	4	19	15	10
12	18	25	24	20
7	14	23	22	16
6	9	21	17	11

Figure 2: Old order of ASM samples and new proposal

As shown in figure 2, the basic idea of our new approach is the same already used but bearing in mind the new offset considerations. Also a bias towards the right-bottom corner of the ASM has been noticed. It is not clear whether this phenomenon is related to the actual design of GASS (we use GASS version 2) or it is indeed a real effect that will be encountered during the mission. Nevertheless, it has been taken as a true effect here and, consequently, the codification order has changed (note the stated right-bottom bias of the codification).

2.3. Improvement in the ROC and ROT values prediction

A pair of ROT (Read-Out Time) and ROC (Read-Out Coordinates) values are linked to each one of the AF and BBP measurements in each star packet. A ROT value consists in a quantity to be added to the reference time of the header of the star packet, giving as a result the exact time in which the measurement was performed by the CCD. The ROC value similarly describes the position of the top-right window corner within the CCD.

The nominal scanning law of Gaia is well defined. Consequently, the pixel windows will be acquired in a uniform and predictable way. It will only need the speed of the object images when projected on the focal plane, a value which can be obtained from two of the star packet header values doing a simple calculation. Hence, the differences between two successive ROTs (and ROCs) will remain stable with very slight variations (changes of 1 or 2 units of the absolute value in most occasions). In the pre-compression algorithm a differential coding scheme is adopted, but a series of modifications have also been introduced:

- Convergence of the ROP & ROT value to a reliable one by including a feedback in the algorithm. A slight correction in the prediction quantity has been performed to have the highest accuracy.
- Changes in the prediction value of the ROP and ROT in the first sample (AF1). As before, a small change has been done to increase precision.
- 3 bits codification instead of 7 or 6. The previous codification used either 7/6 bits or 16 bits to code the differential values. With the corrections applied to the code now it is possible to use only 3 bits. However, the final reduction will not be twice as the previous one because in this way there will be more fields with a value higher than 7 (16 bits codification required).

2.4. Prediction of the AF from the TotalFlux value

Gaia star packets have useful fields in their headers that can be used to improve the codification scheme. Thanks to them it is possible to significantly reduce some of the information generated by the AF and BBP instruments (and it is expected that for the ASM instrument this will be the case as well in the future). In particular, the headers of each star packet have one field called *TotalFlux*, which is an indication of the star apparent magnitude.

On the other hand, although differential coding has many advantages when dealing with easily predictable data, it is also true as well that a seed is generally needed. A reference starting point from which it could start the codification is required.

2.4.1. Graphical analysis of the AF1 wrt the TotalFlux

Now it seems a logical step to take advantage of this information and to code the AF samples accordingly using a simple strategy. The idea is to code them using a first-order approach. Consequently, a *slope* and a *y-intercept* term will be needed. This is not only because the data seems to adapt quite well to the linear distribution, but because keeping the prediction algorithm simple is desirable, given that on-board computing systems have tight limitations in their processing capabilities and power consumption. They must be preserved as much as possible.

In order to further explore this possibility we proceed as follows. We draw the prediction line in a graph in which the density information is displayed instead of using simply a spatial one. Thus, bi-dimensional histograms have been drawn, as can be seen in figure 3.

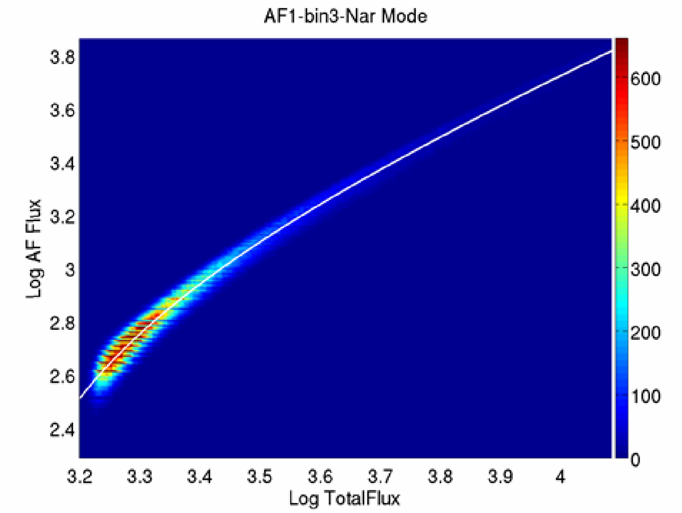


Figure 3: Histogram representation of the third bin of the NWM

This visualization is also more intuitive due to the fact that it allows us to see how the prediction line passes over the densest places. Lines are not exactly centred on these areas in all the cases because of the high dispersion of their distribution and the necessity of minimizing the error in the prediction. In the end we find one prediction line for each of the samples of the AF1 row, either it is Narrow Window Mode (NWM) or Wide Window Mode (WWM).

2.4.2. Generalization to the rest of the AF rows

The previous strategy has led to a better pre-compression ratio and, consequently, a better overall compression ratio (except in the case of ROTs / ROCs previously explained). However, the relation between *TotalFlux* and AF samples cannot be used for the rest of AFs. The prediction of AF1 samples has been largely improved with the additional information given by the magnitude of the star. However, in order to code the next AF rows, the best values to be considered are those of their previous AFs. They will be more accurate than models because they are small variations from a single star observation. And from all the previous AF samples, it seems logical to take the last one as the reference.

While analyzing some of the AF dependencies some patterns were revealed. First of all, the peak (or brightest sample) is always present in the 3rd bin (narrow window mode) or in the 6th bin (wide window mode). Their value is also constant although it presents slight variations (both in wide and narrow mode). The dispersion is really small and the total addition of the three central bins is more or less constant. However, what is not always constant is the value of the side maximum bins (2nd and 4th bins in NWM, and 5th and 7th in WWM). They have noticeable differences between a bin in an AF and the same bin in the previous AF.

Therefore, with this property it will be possible to approximate the value of the (central+1)th bin having the two previous ones.

Also similar properties can be applied to some other bins to reduce the error obtained in the differential coding. The constant bin addition works also for multiple bins, so we can work with them as follows. The following equations have been deduced from a careful analysis of the simulated data provided by GASS. Also some of the values have been tuned up by means of several essays.

- ***NWM***
  - $Bin_1(n) = Bin_1(n-1)$
  - $Bin_2(n) = \frac{Bin_1(n)}{Bin_1(n-1)} \cdot Bin_2(n-1)$
  - $Bin_3(n) = Bin_3(n-1)$
  - $Bin_4(n) = [Bin_2(n-1) + Bin_3(n-1) + Bin_4(n-1)] - [Bin_2(n) + Bin_3(n)]$
  - $Bin_5(n) = \frac{Bin_4(n)}{Bin_4(n-1)} \cdot Bin_5(n-1) \cdot reductFactr$
  - $Bin_6(n) = Bin_6(n-1)$

The reduction factor depends on the result of the Bin_4 division. The aim is to reduce its significance and, in consequence, the factor will be 0.8 if the division is more than 1 and 1.2 is in less than 1.

- ***WWM***
  - $Bin_1(n) = Bin_1(n-1)$
  - $Bin_2(n) = \frac{Bin_1(n)}{Bin_1(n-1)} \cdot Bin_2(n-1)$
  - $Bin_3(n) = \frac{Bin_2(n)}{Bin_2(n-1)} \cdot Bin_3(n-1)$
  - $Bin_4(n) = \frac{Bin_2(n)}{Bin_2(n-1)} \cdot Bin_4(n-1)$
  - $Bin_5(n) = \frac{Bin_2(n)}{Bin_2(n-1)} \cdot Bin_5(n-1)$
  - $Bin_6(n) = Bin_6(n-1)$
  - $Bin_7(n) = [Bin_5(n-1) + Bin_6(n-1) + Bin_7(n-1)] - [Bin_5(n) + Bin_6(n)]$
  - $Bin_8(n) = \frac{Bin_7(n)}{Bin_7(n-1)} \cdot Bin_8(n-1)$
  - $Bin_9(n) = \frac{Bin_8(n)}{Bin_8(n-1)} \cdot Bin_9(n-1)$
  - $Bin_{10}(n) = \frac{Bin_8(n)}{Bin_8(n-1)} \cdot Bin_{10}(n-1)$
  - $Bin_{11}(n) = \frac{Bin_8(n)}{Bin_8(n-1)} \cdot Bin_{11}(n-1)$
  - $Bin_{12}(n) = Bin_{12}(n-1)$

Obviously, these equations are not applicable to the first AF row, which is predicted from the linear equations.

2.5. New pre-compression rules

With all these changes implemented together in the new *GaiaSPaP* code we have obtained a better compression ratio. In the algorithm each value is usually saved using 16 or 8 bits (2^{16}=65536, 2^8=256) depending on the maximum value of the entire block. For example, if we have all the ASM patch and the central differential value (the maximum is always centred) is larger than 255, a flag bit will be saved as 1, and all the 25 ASM values will be coded using 16 bits. Otherwise, if the maximum value is less than 256, the flag bit is saved as 0, and all the ASM values will be coded using 8 bits.

However, it could be argued that, as the prediction is now improved, we should have now smaller differences than when the previous pre-compression scheme is used. Thus, it may be possible to lower sizes of the files with small codification changes. It is only needed to change the 8-bits codification condition to 127 instead of 255. Now the value fits in 7 bits (2^7=128) and we have one bit left which we will use to code the sign bit. This bit was previously coded in a reference file, and it continues to be coded there in the 16-bits variant. Now an important file size reduction is achieved in an easy way. And what is more important, they have been obtained with no overload of the processing unit.

3. Final results with the new techniques

Applying all the modifications and improvements stated above, significantly better results are obtained. Nonetheless, it is necessary to distinguish between the ratios *before* and *after* compression. As explained before, there are two compression phases: firstly there is the *rearrangement* stage and secondly the *standard compression* stage. It is the first one where the changes made are effective but, however, they affect to the final ratio obtained too. Therefore, a better result in the first phase does not always mean a better final result as it is shown below.

As an example, we show in figure 4 the statistical analysis for the different methods used – which directly leads to the entropy and maximum theoretical ratios. Our new results seem to be worse than those obtained with the previous pre-compression algorithm, but this is not exactly accurate. As already explained, there has been a codification change (7 bits + 1 sign bit) that makes entropy worse than without it. This can be verified by observing the green line: it depicts the situation for the same values but with the original coding scheme of 16 or 8 bits and the sign bit coded in the ref file. In this case it is clear that the new changes done improve the algorithm. So, the final values histogram (in blue) seems to be worse, but we have already decreased the output file size and, consequently, the final compression ratio will be better.

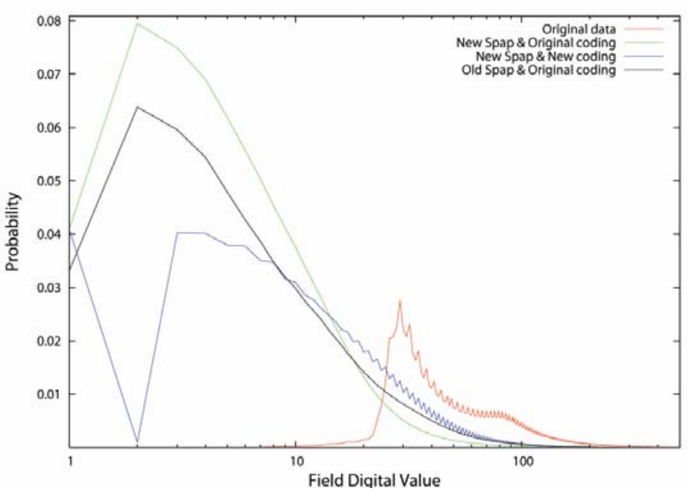


Figure 4: Histogram of AF sample values

3.1. Pre-compression ratio achieved

The final tests are done using an extensive simulation of a great circle of Gaia which means a complete revolution of the satellite. The value obtained from this simulation is considered as highly reliable and it is held as the standard value because it is not constrained to a small piece of the sky but, instead, passes through high (the galactic plane) and low star density areas.

Two different versions are chosen because it was found that the version which produced the best pre-compression ratio did not produce the best final ratio. This phenomenon seems to be related with the way in which standard compressors work. They are able to eliminate some kind of redundancies while others remain. When doing this new codification either the bits are somehow rearranged in a way that does not ease their work or the erased redundancy was already detected and deleted in the old version (we have a gain in pre-compression ratio but not in the final one).

3.2. Compression ratio achieved

Telemetry data will be transmitted after the final compression stage, and the ratio achieved is the key factor to know if a certain combination of compression algorithms will be acceptable for Gaia. There are two parameters of importance to make the choice: processing time and block size.

Block size is important because of the transmission reliability, that is, large block sizes could lead to large data losses if transmission errors occur. On the other hand, CPU load must obviously be checked, since energy and processing power are very valuables resources in space. Hence, once the ratio is achieved it is mandatory to verify that the computer resources are sufficient to make the real-time compression. Consequently, we must also take into account processing time.

Figures 5 and 6 are obtained using data from the great circle simulation. Note that they can be considered as real simulation data, completely valid for calculations. In the figures we show the processing time required for both algorithms to compress the simulation depending on the block size.

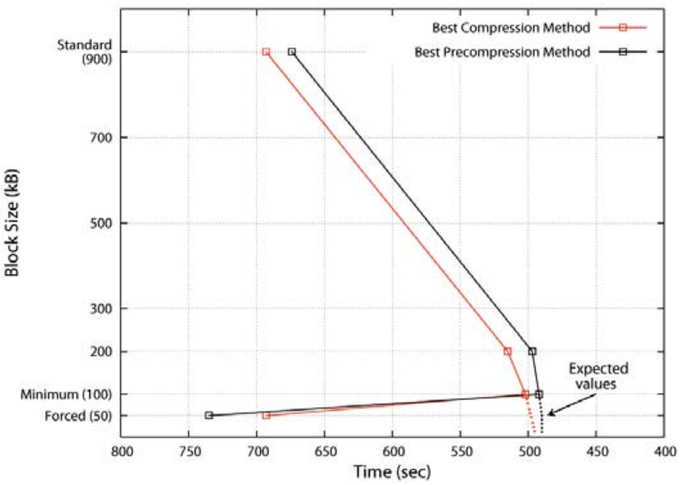


*Figure 5: Time vs Block Size when using **Bzip2** algorithm*

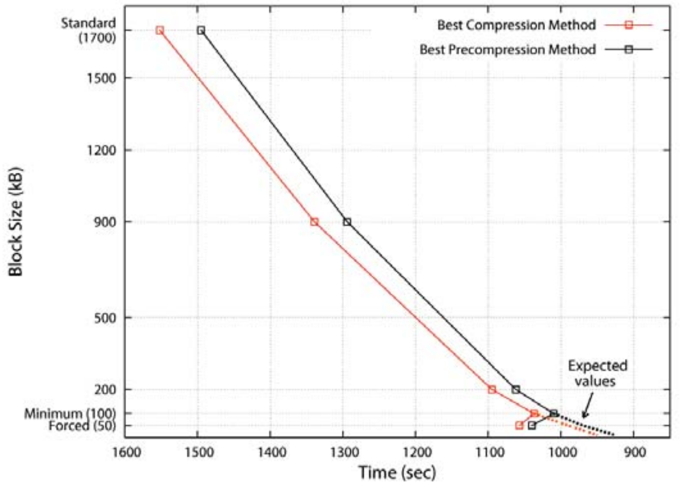


*Figure 6: Time vs. Block Size when using **Szip** algorithm*

Only two standard compression methods have been used for testing the compression stage. In previous studies [(2)] many of them were analysed, but the final conclusion was that only these two methods were interesting for our application. Each of them has been tested in a GNU/Linux environment with open-source implementations (Szip v1.12 and Bzip2 v1.13.25), a Pentium IV processor and a 2.6.5.1 kernel. They were previously used only in their standard configuration, that is, 900 kB block size for Bzip2 and 1700 kB block size for Szip. The compressors take files in little pieces to manage them easily, each one of them of the chosen '*block size*' size. The larger the block sizes, the better the compression ratios – but the higher the processing times. Therefore, this value must be carefully chosen.

Furthermore, another constraint appears here, because at this moment no block size larger than 100-200 kB is recommended for the mission. The main reason for this decision is that all data from a block size is needed by the compression algorithm to reconstruct original data. As data is transmitted to the ground station and space medium is not highly reliable, block sizes cannot be very large to minimize data loss (the whole block size) in case of an unrecoverable transmission error. Thus, only the lower part of graphs must be considered, although the higher part is used for comparisons with old studies.

Szip and Bzip2 could only be configured down to block sizes of 100 kB, due to the fact that lower sizes will almost not improve the processing time but will worsen compression ratio. The problem is that probably the alternative to be implemented in Gaia may be of 50 kB or even 10 kB. To test their results with these *strange* values one little trick was performed: pre-compressed files were split into files with the size of the block size (50, 10 & 2 kB) and passed through the standard compressors.

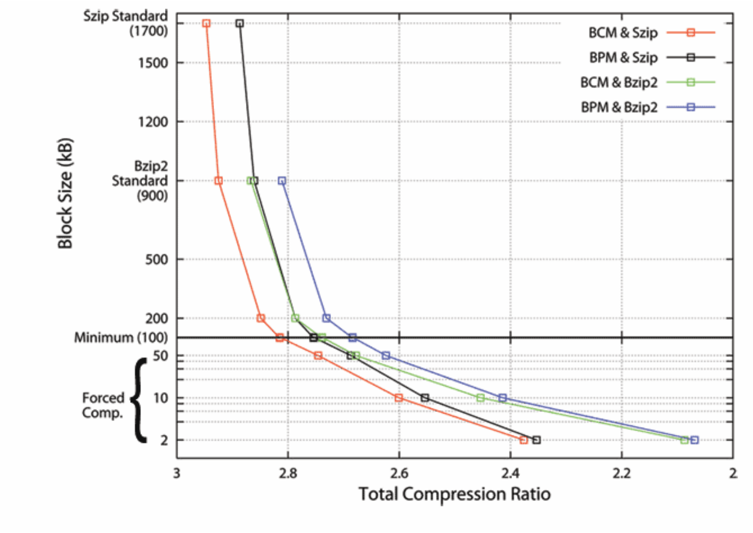


Figure 7: Compression ratio vs Block Size for both algorithms

In figure 7 it is shown the final compression results in the test file. With lower block sizes, differences between Szip and Bzip2 tend to get bigger and it is not clear that this shall respond completely to a real evolution. As the split file sizes get smaller, things that were negligible in standard cases, i.e. compressor headers, start to be dominant. So they are reference values only.

However, for the most realistic case (small allowed block size: 100kB) we see a difference of about 0.1 which corresponds to a 5% of the ratio. Having into account that CPU compressing time for Szip is twice as the Bzip2 case, we currently consider the Bzip2 as the best option for the compression stage.

4. Conclusions and forthcoming work

The final objective of this codec modification has been to improve the compression ratios obtained in a *simple* manner: the processing unit was slightly overloaded wrt the initial algorithm. Nonetheless, the ratios and improvements achieved are already in line with the requirements of the mission as currently stated. It seems that the Bzip2 algorithm is faster and powerful enough to accomplish with the restrictions previously mentioned.

At the moment the development of this new codec version has been tackled conservatively. Now that the new GASS version has appeared, it is mandatory to upgrade to it. Consequently, it is necessary to export all the changes previously done to a new *gaiaspap* version. This new version will include new capabilities as the effect of cosmic rays and NEOS which will make simulations more realistic. Thus, the effects in the pre-compression algorithm of these new models must be analysed in order to verify that they are robust to these changes. Also it would be interesting to make some tests with deviated values of the prediction lines to see how these changes affect the compression ratios.

On the other hand, there are still many improvement possibilities in *gaiaspap*. Most of them must be studied to state the convenience or possibility of adding them to the code, as for example a probability-based code. Another possibility is the direct implementation of one of the standard compression algorithms into the code, to see the real performance with variable block sizes. It would be also possible to improve and debug in order to optimize it for this very specific problem. They should be tested and evaluated to decide whether their improvement/complexity compromise is good enough to be included in the final release. Obviously it will bring a higher load to the processing unit, so there must be taken a final decision on which of the modifications are included in the final version and which not.

4. References

(1) ESA-SCI(2000)4, 2000. GAIA Concept and Technology Study Report (Red Book).

(2) GAIA-BCN-013, 2005. Realistic tests of the data compression system using GASS data.

(3) GAIA-BCN-011, 2004. Definition of a telemetry CODEC.

(4) GAIA-UL-008, 2004. Gaia telemetry rate simulations: A first look at the complete picture.

(5) GAIA-CUO-113, 2002. Sampling for all magnitudes - scheme C.