6

Non-extensive Thermodynamics of Algorithmic Processing – the Case of Insertion Sort Algorithm

Dominik Strzałka and Franciszek Grabowski
Rzeszów University of Technology
Poland

1. Introduction

In this chapter it will be shown that there can exist possible connections of Tsallis non-extensive definition of entropy (Tsallis, 1988) with the statistical analysis of simple insertion sort algorithm behaviour. This will be done basing on the connections between the idea of Turing machines (Turing, 1936) as a basis of considerations in computer science and especially in algorithmic processing and the proposal of non-equilibrium thermodynamics given by Constatino Tsallis (Tsallis, 1988; Tsallis, 2004) for indication of the possible existence of non-equilibrium states in the case of one sorting algorithm behaviour. Moreover, it will be also underlined that a some kind of paradigm change (Kuhn, 1962) is needed in the case of computer systems analysis because if one considers the computers as physical implementations of Turing machines should take into account that such implementations always need energy for their work (Strzałka, 2010) – Turing machine as a mathematical model of processing does not need energy. Because there is no (computer) machine that have the efficiency η = 100%, thus the problem of entropy production appears during their work. If we note that the process of sorting is also the introduction of order (obviously, according to a given appropriate relation) into the processed set (sometimes sorting is considered as an ordering (Knuth, 1997)), thus if one orders it must decrease the entropy in sorted set and increase it somewhere else (outside the Turing machine – in physical world outside its implementation). The connections mentioned above will be given basing on the analysis of insertion sorting, which behaviour for some cases can lead to the levels of entropy production that can be considered in terms of non-extensivity. The presented deliberations can be also related to the try of finding a new thermodymical basis for important part of authors' interest, i.e., the physics of computer processing.

2. Importance of physical approach

The understanding of concept of entropy is intimately linked with the concept of energy that is omnipresent in our lives. The principle of conservation of energy says that the difference of internal energy in the system must be equal to the amount of energy delivered to the system during the conversion, minus the energy dissipated during the transformation. The principle allows to write an appropriate equation but does not impose any restrictions on the quantities used in this equation. What's more, it does not give any indications of how the energy should be supplied or drained from the system, or what laws (if any exist) should govern the transformations of energy from one form to another. Only the differences of transformed energy are important. However, there are the rules governing the energy transformations. A concept of entropy and other related notions create a space of those rules.

Let's note that Turing machine is a basis of many considerations in computer science. It was introduced by Alan Mathison Turing in the years 1935–1936 as a response to the problem posed in 1900 by David Hilbert known as the *Entscheidungsproblem* (Penrose, 1989). The conception of Turing machine is powerful enough to model the algorithmic processing and so far it haven't been invented its any real improvements, which would increase the area of decidable languages or which will improve more than polynomial its time of action (Papadimitriou, 1994). For this reason, it is a model which can be used to implement any algorithm. This can be followed directly from Alonso Church's thesis, which states that (Penrose, 1989; Wegner & Goldin, 2003):

"Any reasonable attempt to create a mathematical model of algorithmic computation and to define its time of action must lead to the model of calculations and the associated measure of time cost, which are polynomial equivalent to the Turing machines."

Note also that the Turing machine is, in fact, the concept of mathematics, not a physical device. The traditional and widely acceptable definition of machine is connected with physics. It assumes that it is a physical system operating in a deterministic way in a well-defined cycles, built by a man, whose main goal is focusing energy dispersion for the execution of a some physical work (Horákowá et al., 2003). Such a machine works almost in accordance with the concept of the mechanism specified by Deutsch – as a perfect machinery for moving in a cyclical manner according to the well-known and described laws of physics, acting as a simple (maybe sometimes complicated) system (Deutsch, 1951; Grabowski & Strzałka, 2009; Amral & Ottino, 2004).

On the other hand the technological advances have led to a situation in which there is a huge number of different types of implementations of Turing machines and each such an implementation is a physical system. Analysis of the elementary properties of Turing machines as a mathematical concept tells us, that this is a model based on unlimited resources: for example, in the Turing machine tape length is unlimited and the consumption of energy for processing is 0 (Stepney et al., 2006). This means that between the mathematical model and its physical implementation there are at least two quite subtle but crucial differences: first, a mathematical model that could work, does not need any Joule of energy, while its physical implementation so, and secondly, the resources of (surrounding) environment are always limited: in reality the length of Turing machine tape is limited (Stepney et al., 2006).

Because in the mathematical model of algorithmic computations there is no consumption of energy, i.e., the problem of physical efficiency of the model (understood as the ratio of energy supplied to it for work, which the machine will perform) does not exist. Moreover, it seems that since the machine does not consume energy the possible connections between thermodynamics and problems of entropy production aren't interesting and don't exist. However, this problem is not so obvious, not only due to the fact that the implementations of Turing machines are physical systems, but also because the use of a Turing machine for the solution of algorithmic problems can be also associated with the conception such as the order, which is (roughly speaking) *anti-entropic*. A classic example of this type of problem is sorting. It is usually one of the first problems discussed at the courses of algorithms to show what is the algorithmic processing and to explain the idea of computational complexity (see for example the first chapter in famous book (Cormen et al., 2001)).

Generally, the main objective of sorting is in fact find such a permutation (ordering change) $< a_1', a_2', ..., a_N' >$ of the input consisting of *N* numbers (or in general *N* keys) $< a_1, a_2, ..., a_N >$ to ensure that $a_1' \le a_2' \le ... \le a_N'$. As one can see the search for an appropriate permutation is carried out using the ordering relation < defined on the values (keys) and the following conditions for three values *a*, *b*, *c* are satisfied:

- there is true exactly one of the possibilities $a < b$, $a = b$, $b < a$;
- if $a < b$ and $b < c$, then $a < c$.

In this chapter, basing on the context of so far presented considerations, it will be discussed a simple algorithm for sorting based on the idea of insertion sort. This is one of the easiest and most intuitive sorting algorithms (based on the behaviour of bridge player who sorts his cards before the game) and its detailed description can be found in the literature (Cormen et al., 2001). It is not too fast algorithm (for the worst-case it belongs to a class of algorithms of complexity $O(n^2)$, however for the optimistic case it has the complexity $\Omega(n)$), but it is very simple, because it only consists of two loops: the outer guarantees sorting of all elements and the internal one, which finds the right place for each key in the sorted set. This loop is a key-point of our analysis because it will represent a very interesting behaviour in the context of analysis of algorithm dynamics for all possible input set instances. This follows from the fact that the number of this inner loop executions, which can also be identified with the duration of this loop, depends on (Strzałka & Grabowski, 2008):

- the number of sorted keys (the size *n* of the task). If, for example, the pessimistic case is sorted for long input sets and elements of small key values, the duration of this loop can be very long especially for the data contained at the end of the input sorted set;
- currently sorted value of the key. If the sorting is done in accordance with the relation "<", then for large values of data keys finding the right place in output set should last a very short period of time, while for small values of keys it should take a lot of inside loop executions. Thus, all parts of the input set close to the optimistic case, i.e., the parts with preliminary, rough sort of data (e.g., as a result of the local growing trend in input), will result in fewer executions of inner loop, while the parts of input set closer to the worst-case (that is, for example, those with falling local trends) will mean the need of many executions of inside loop.

The third condition is visible when the algorithm will be viewed as a some kind of *black box* (system), in which the input set is the system INPUT and the sorted data is the system OUTPUT (this approach is consistent with the considerations, which are given by Knuth in (Knuth, 1997) where in his definition of algorithm there are 5 key features among which are the *input* and *output* or with the approach presented by Cormen in (Cormen et al., 2001)). Then it can be seen that there is a third additional condition for the number of inner loop executions: the so far sorted values of processed set contained in this part of the output, where the sorting was already done, influence on the number of this loop executions. Thus we have an elementary feedback. The position of each new sorted element depends not only on its numerical value (understood here as the input IN), but also on the values of the items already sorted (that is, *de facto* output OUT). If it were not so, each new element in the sorted input would be put on pre-defined place in already sorted sequence (for example, it would

be always included at the beginning, end or elsewhere within the output – such a situation is for example in the case of sorting by selection).
The above presented observations will influence the dynamics of analysed algorithm and its analysis will be conducted in the context of thermodynamic conditions. Let's note once again that the sorting is an operation that introduces the order into the processed set and in other words it is an operation that reduces the level of entropy considered as the measure of disorder. In the case of the classical approach, which is based on a mathematical model of Turing machines the processing will cause the entropy reduction in the input set but will not cause its growth in the surroundings of the machine (it doesn't consume the energy). But in the case of the physical implementation of Turing machine, the processing of input set must result in an increase of entropy in the surroundings of the machine. This follows from the fact that even if the sorting operation is done by the machine that has the efficiency $\eta = 100\%$ it still will require the energy consumption – this energy should be produced at the source and this lead to the increase of the entropy "*somewhere*" near the source.

3. Levels of entropy production in insertion-sort algorithm

The presented analysis will be based on the following approach (Strzałka & Grabowski, 2008). If the sorted data set is of size *n*, then it can occur *n*! of possible key arrangements (input instances). One of them will relate to the case of the proper arrangement of elements in the set (i.e., the set is already sorted – the case is the optimistic one), while the second one will relate to the worst-case (in the set there will be arrangement, but different from that required). For both of these situations it can be given the exact number of dominant operations that should be done by the algorithm, while for the most of other $n! - 2$ cases this is not necessary so simple. However, the analysis of insertion sorting can be performed basing on the conception of inversions (Knuth, 1997). The number of inversions can be used to calculate how many times the dominant operation in insertion sort algorithm should be done, but it is also an indication of the level of entropy in the processed set, since the number of inversions is information about how many elements of the set are not ordered. Of course, the arrangement will reduce the entropy in the set, but it will increase the entropy in the environment.
Therefore, we can consider the levels of entropy production during insertion sorting. If we denote by M the total number of executions of inside and outside loops needed for successive n_i elements processed from the input set of size n, then for each key $M = n_i$. Let M_1 will be the number of outer loop requests for each sorted key – always it will be $M_1 = 1$. If by M_2 we will denote the number of inner loop calls, then it may vary from 0 to $n_i - 1$, and if by M_3 we determine the number of such inside loop executions that may have occurred but not occurred due to the some properties of sorted set, we will have $M = M_1 + M_2 + M_3$. For the numbers M_1, M_2 and M_3 one can specify the number of possible configurations of inner and outer loop executions in the following cases: optimistic, pessimistic and others. By the analogy, this approach can be interpreted as a try to determine the number of allowed microstates (configurations), which will be used to the analysis of entropy levels production in the context of the number of necessary internal loop executions.
This number will be equal to the number of possible combinations $C_M^{M_1}$ multiplied by $C_{M-M_1}^{M_2}$ (this number is multiplicative):

$$W = C_M^{M_1} \cdot C_{M-M_1}^{M_2} = \frac{M!}{M_1!(M-M_1)!} \cdot \frac{(M-M_1)!}{M_2!(M-M_1-M_2)} = \frac{M!}{M_1!M_2!M_3!}, \quad (1)$$

i.e., the number C of M_1 combinations of necessary outer loop calls from M executions multiplied by C combinations of M_2 necessary executions of inner loop from the rest possible $M - M_1$ calls.

An optimistic case is characterized by the need of a single execution of outer loop ($M_1 = 1$) for each sorted key, the lack of inside loop calls ($M_2 = 0$) and $n_i - 1$ no executions of this loop ($M_3 = n_i - 1$), which means that the number of possible W_O configurations of these two loops will be equal

$$W_O = \frac{n_i!}{1!0!(n_i - 1)!} = n_i . \quad (2)$$

For the pessimistic case it will be: $M_1 = 1$ and $M_2 = n_i - 1$ – one need to use this loop a maximal available times – $M_3 = 0$, thus W_P (P – pessimistic) will be equal

$$W_P = \frac{n_i!}{1!(n_i - 1)!0!} = n_i . \quad (3)$$

Thus, the number of microstate configurations in both cases is the same ($W_O = W_P$). It might seem a little surprising, but it is worth to note that although in the worst case the elements are arranged in reverse order than the assumed in sorting process, it is still the order. From the perspective of thermodynamics the optimistic and pessimistic cases are the same because they are characterised by the entropy production at the lowest possible level; in any other cases W will be greater. For example let's consider the case when one needs only one excess dominant operation for key n_i, i.e., : $M_1 = 1$, $M_2 = 1$, $M_3 = n_i - 2$, so W_D (D – dynamical) will be equal

$$W_D = \frac{n_i!}{1!1!(n_i - 2)!} = \frac{(n_i - 2)!(n_i - 1)n_i}{(n_i - 2)!} = n_i(n_i - 1) . \quad (4)$$

The lowest possible levels of entropy production for the optimistic or pessimistic cases correspond to the relationship given by Onsager (Prigogine & Stengers, 1984). They show that if a system is in a state close to thermodynamic equilibrium, the entropy production is at the lowest possible level. Thus, while sorting by the insertion-sort algorithm the optimistic or pessimistic cases, Turing machine is in (quasi)equilibrium state.

It can be seen that in the optimistic and pessimistic cases the process of sorting (or entropy production) is extensive, but it is not known if these considerations are entitled to the other instances. However, one can see this by doing a description of the micro scale, examining the behavior of the algorithm for input data sets with certain properties (let's note that this is in contradiction to the commonly accepted approach in computer science where one of the most important assumptions in computational complexity assumes that this measure should be independent on specific instances properties, thus usually the worst case is considered (Mertens, 2002)). Moreover, to avoid problems associated with determining the number of inversions, one can analyze the behavior of this algorithm by recording for each sorted key the number of executed dominant operations (it will be labeled as $Y(n)$) and then examine the process of increments of number of dominant operations (i.e., $Y'(n)$); in other words – to consider how the process

$$Y'(n) = Y(n+1) - Y(n) \tag{5}$$

can behave.

The equation (1) shows the entropy production for each sorted key. In the classical analysis of algorithms computational complexity two similar ways can be taken: one can consider a total number of dominant operations executions or a number of dominant operations for each processed element of input set (the increments of the first number). The second approach will be more interesting one and it will show the properties of distribution of all possible increments of number of dominant operations: for each sorted key n_i the number of dominant operations is a random variable and its values can appear with changing probabilities for each n_i. We would like to know how the distribution of increments looks like when $n_i \to n$ and of course when $n \to \infty$. It is not hard to see that in the optimistic case, the expression (5) will always be zero, and for the worst-case is always equal to one. If there will be sorted the instances "similar" to the cases that are optimistic or pessimistic, the deviations from the above number of increments will be small and their probability distributions should be characterized by quickly vanishing tails, therefore, it will belong to the Gaussian basin of attraction. As we know this is a distribution, which is a natural consequence of the assumptions underlying the classical definition of Boltzmann-Gibbs entropy. However, it may also turn-out that certain properties of sorted input sets (e.g., long raising and falling trends) will cause that the probability distributions of the number of dominant operations increments will have a different character, and then the concept of non-extensive entropy will be useful.

4. Non-equilibrium states of insertion sort algorithm behaviour

In order to visualize the so far presented deliberations a simple experiment involving the sorting of input data sets by insertion-sort algorithm was done. Sorted sets were the trajectories of one-dimensional Brownian motion (random walk) – denoted by $X(t)$; see Fig. 1. Each sorted set has 10^6 elements. One of the most characteristic feature of these sets is the presence of local increasing or decreasing trends, which for the sorting algorithm can be regarded as a local optimistic or pessimistic cases. These trends and their changes should affect the dynamics of algorithm behavior (considered as the changing number of executed dominant operations) – see for example Fig. 1. If sorting is done according to the non-decreasing order (i.e., by the relation $\leq$), any raising trend would be the case of initially correct order of keys in input data (in other words it can be very roughly treated as a case similar to the optimistic one) – in mathematical analysis, this situation would be described by a small number of inversions. However, any falling trend will be the case of improper order of data (i.e., very roughly – similar to the worst-case) – in mathematical analysis, this situation involves a large number of inversion. Any raising trend in input set will cause the decline of the number of dominant operations, while the falling trend its rapid growth (Fig. 1).

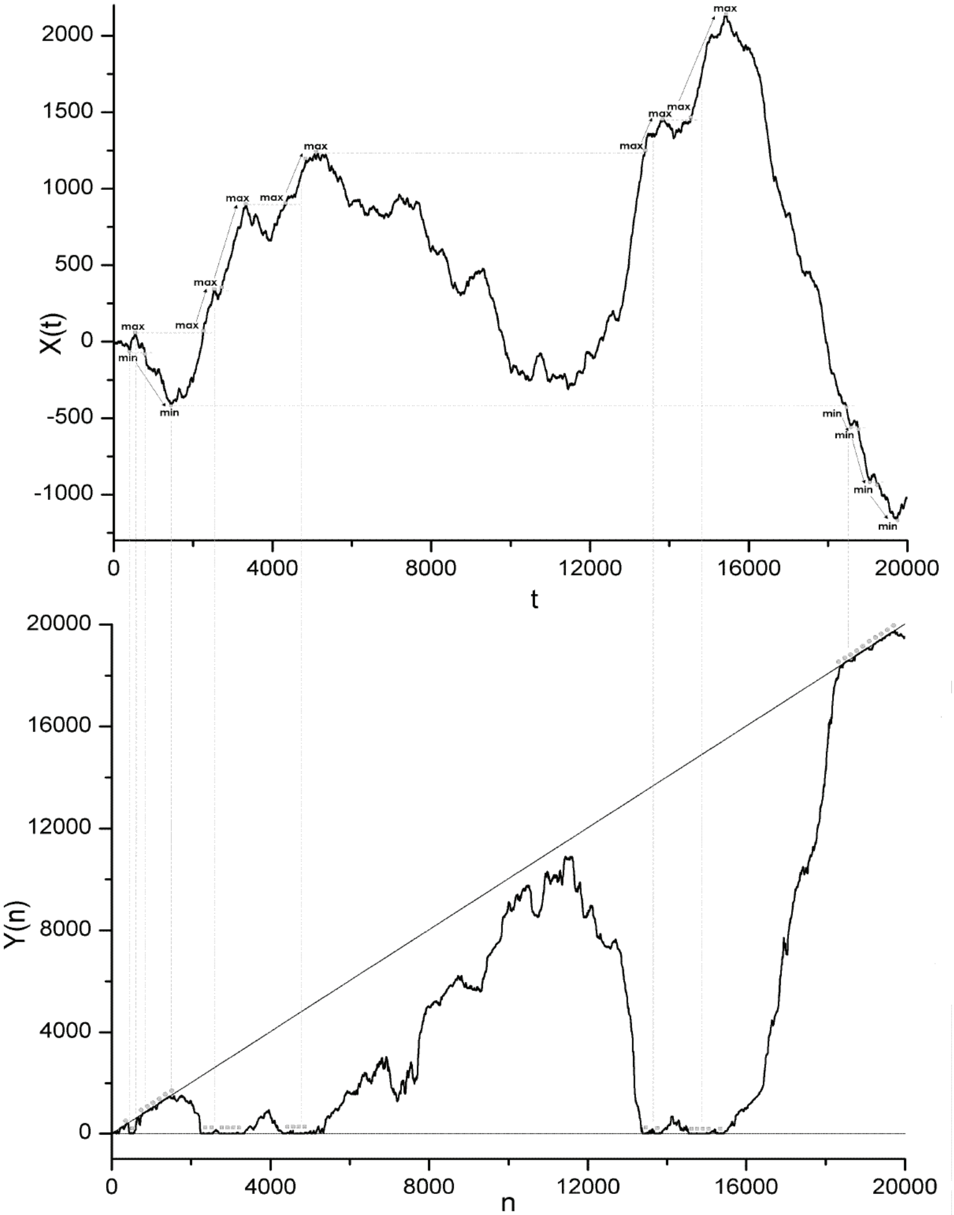


Fig. 1. The example of one input set (top) with the behaviour of algorithm (down), i.e., the recorded set of dominant operations needed for sorting n = 20000 keys. As it can be seen when in input set there is a local minimum, there is a need to execute the maximal number of dominant operations, and when in input set there is a local maximum there is a need to execute only one dominant operation

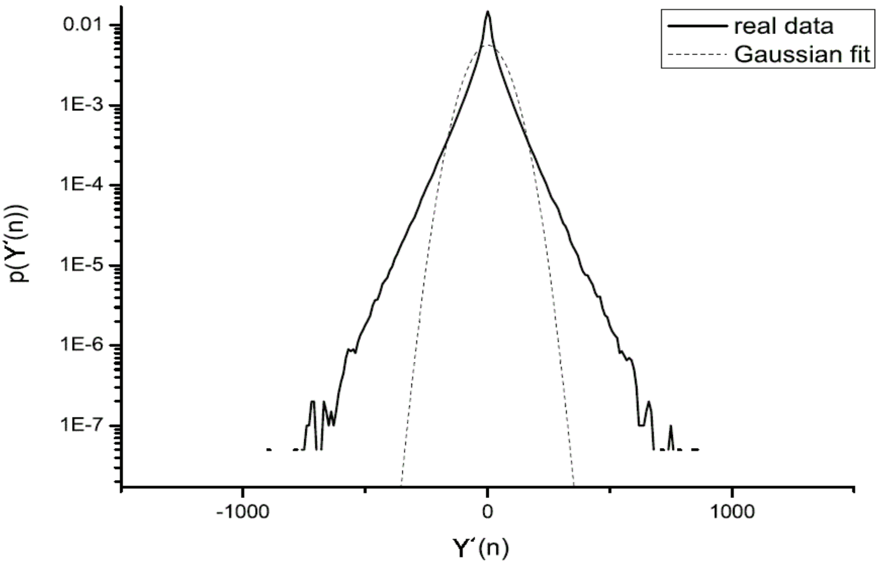


Fig. 2. The example of one empirical probability distribution of increments $Y'(n)$ of dominant operations used by insertion-sort algorithm. Line with dots stands for a normal distribution fitted by calculated process mean and variance; continuous line represents the empirical distribution obtained by a kernel estimator

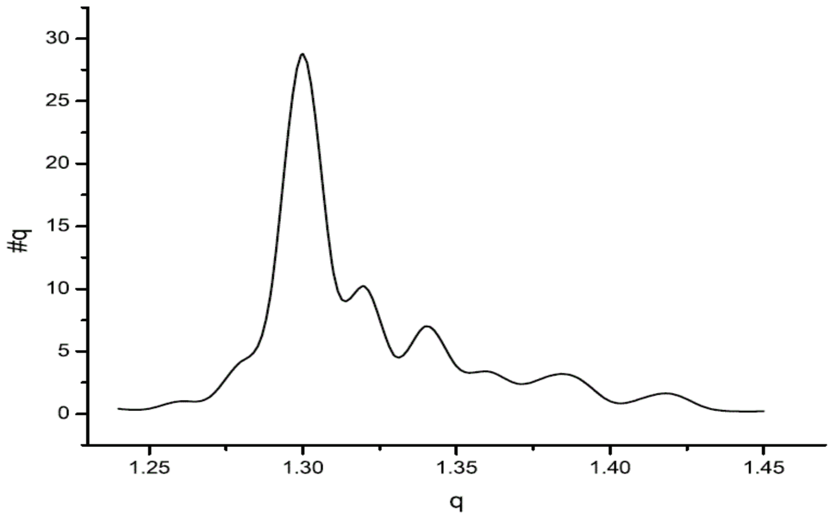


Fig. 3. The distribution of q values for 500 sorted sets; as it can bee seen in most cases there is $q \approx 1.3$, but its range lies between 1.25 and 1.45

During the experiment, the numbers of dominant (needed for each sorted key) operations $Y(n)$ were recorded. 500 sets of input data have been sorted and as a result we received the set of sorting processes realisations. Next, primarily the empirical probability density distributions of increments $Y'(n)$ of the number of dominant operations were examined. As one can see (Fig. 2), the empirical distribution of dominant operations increments has slowly vanishing tails than fitted by process mean and variance normal distribution. This is a first example, which shows that the process of sorting by insertion input sets that are the trajectories of random walk can be related to the idea of Tsallis approach with his proposal of q-Gaussian distributions (Tsallis et al., 1995; Alemany & Zanette, 1994). In this case the estimated value of q parameter equals ≈ 1.3.

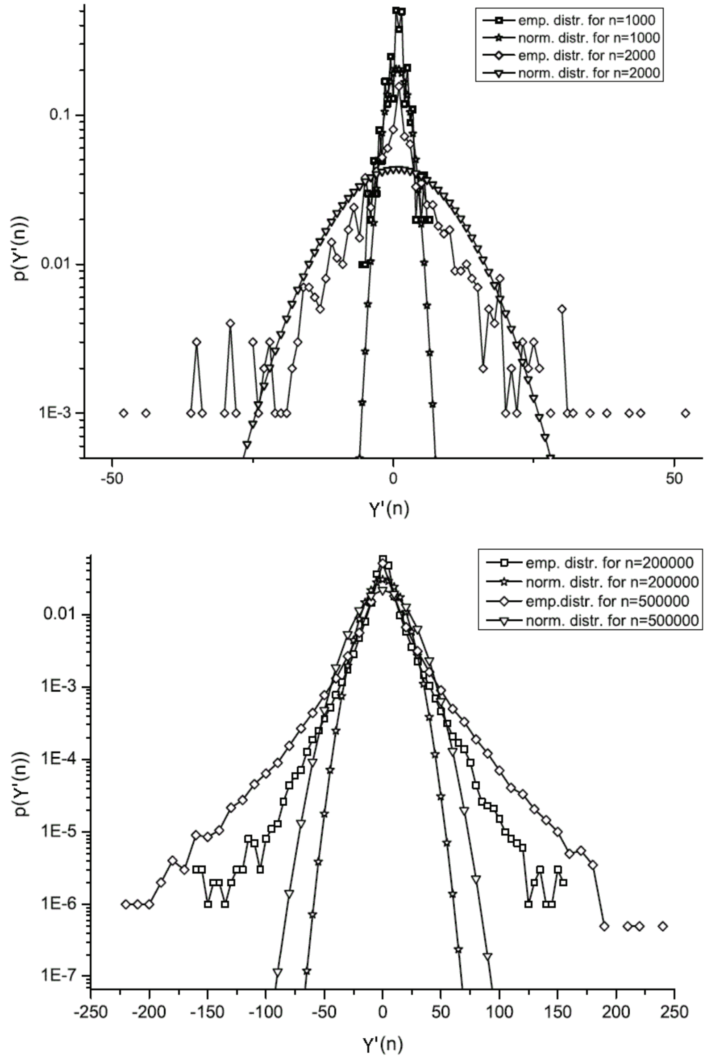


Fig. 5. The comparison of empirical and fitted normal distributions for increment processes $Y'(n)$, when the input size n = 1000 and n = 2000 (top figure) and when n = 200000 and n = 500000 (bottom figure)

The presented case allows us to ask some interesting questions. First one is obvious: is it only the one case when such a situation appears, or this situation is a "normal" one. The answer to the following question can be given immediately because as it was written above in the experiment 500 sets were sorted and for each sorted set we can see similar behaviour of empirical distributions. But this obviously allows to ask another important question: how the values of q parameter change for different input sets. To answer this we perform an analysis for all sorted sets an its results can be seen on Fig. 3 – the values of q parameter are between 1.25 and 1.45.

The second important problem can be the size n of input set: it can be very interesting for what n we can see the first symptoms of slowly vanishing tails for distributions of increments. If the number of keys in the sorted set is small (Fig. 4 shows the empirical and fitted Gaussian distributions for sets with different input sizes n), for example less than 1000 keys the q-Gaussian probability distribution of increments for sorting process is not clearly visible, but if the input set has quit large n the considered effect is definitely more visible. One of the conclusions drawn from these observations can be the suggestion that the classical (mathematical) analysis of this algorithm behavior (for example, shown in (Cormen et al., 2001)), in which the behavior of the algorithm for small data sets also shows the possible behavior for any size of input data, i.e., in a some sense it is *extended* to sets of input data of any size n, not quite well shows the nature of all possible behaviors of the analyzed algorithm. It seems that the problem considered in this chapter (non-extensive behavior of entropy) is rather the emergent one – it can appear when the size n of input is numerous.

Of course, the insertion sort algorithm isn't a very efficient one (we know this basing on its computational complexity) and sorting of large data sets is rather done for example by Quick-sort, but this simple algorithm can open a new field of discussion concerning the analysis of algorithms in the context of not only the determinants of the nature of mathematics, but also the conditions related to the characteristics of a physical nature – this can be viewed as a some kind of paradigm change in the so far presented approaches.

5. Conclusions

The chapter presents considerations concerning the existence of possible connections between non-extensive definition of entropy proposed by C. Tsallis and algorithmic processing basing on the example of sorting by insertion-sort. The figures presenting empirical distributions in log-lin scale show the existence of a slowly vanishing tails of probability distributions indicating that the thermodynamic conditions of analysed algorithm work emerge for a suitably large data sets – in a some sense, it seems that this is a feature of an emergent character; in the classical analysis of algorithms it can't be taken into account. In the chapter it was also shown that some features of input sets can also be transferred on the level of dynamic behaviour of the algorithm and the number of necessary dominant operations that are performed during its work. This differs from the assumption in the classical computational complexity analysis where the most interesting and authoritative case is the pessimistic (worst) one even if one takes into account also the analysis of the average case. Meanwhile, the analysis of algorithm combined with knowledge about the properties of processed data shows that there can appear interesting phenomena that may be of fundamental importance for the analysis of Turing machines that are not treated as a mathematical models, but considered in the context of their physical properties of the implementations.

6. References

Alemany, P. A. & Zanette, D. H. (1994). Fractal random walks from a variational formalism for Tsallis entropies, *Physical Reviev E*, Vol. 49, (1994) p. R956, ISSN 1539-3755

Amaral, L. A. N. & Ottino, J. M. (2004). Complex Systems and Networks: Challenges and Opportunities for Chemical and Biological Engineers, *Chemical Engineering Science*, Vol. 59, No. 8-9, (2004) pp. 1653–1666, ISSN 0009-2509

Cormen, T. H.; Leiserson, Ch. E.; Rivest, R. L. & Stein, C. (2001). *Introduction to Algorithms* (second ed.), MIT Press and McGraw-Hill, ISBN 978-0-262-53196-2, New York

Deutsch, K. (1951). Mechanism, Organism, and Society, *Philosophy of Science*, Vol. 18, (1951) pp. 230–252, ISSN 00318248

Grabowski, F. & Strzałka, D. (2009). Conception of paradigms evolution in science – towards the complex systems approach, *Egitania Scienca*, Vol. 4, (2009) pp. 197–210, ISSN 1646-8848

Horákowá, J.; Kelemen, J, &, Čapek, J. (2003). Turing, von Neumann, and the 20th Century Evolution of the Concept of Machine. *International Conference in Memoriam John von Neumann*, John von Neumann Computer Society, pp. 121–135, Budapešť, 2003.

Knuth, D. E. (1997). *The art of computer programming, vol. 1*, Addison-Wesley, ISBN 0-201-89683-4, Massachusetts

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*, University of Chicago Press, ISBN 0-226-45808-3, Chicago

Mertens, S. (2002). Computational Complexity for Physicists, Computing in Science and Engineering, Vol. 4, No. 3, (May 2002) pp. 31–47, ISSN 1521-9615

Papadimitriou, Ch. H. (1994). *Computational Complexity*, Addison Wesley, ISBN 0201530821, Massachusetts

Penrose, R. (1989). *The Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*, Oxford University Press, ISBN 0-198-51973-7, New York

Prigogine, I. & Stengers, I. (1984). *Order out of Chaos: Man's new dialogue with nature,* Flamingo, ISBN 0006541151,

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, Series 2(42), (1936) pp. 230–265. Errata appeared in Series 2(43), (1937) pp. 544–546

Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics, *Journal Statistical Physics*, Vol. 52, (1988) p. 479, ISSN 0022-4715

Tsallis, C.; Levy, S. V. F.; Souza, A. M . C. and Mayanard, R. (1995). Statistical-Machanical Foundation of the Ubiquity of Lèvy Distributions in Nature, *Physical Review Letters*, Vol. 75, (1995) p. 3589, ISSN 0031-9007

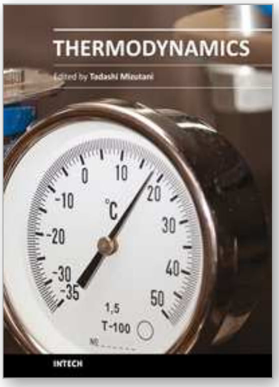
Tsallis, C. (2004). What should a statistical mechanics satisfy to reflect nature?, *Physica D*, 193, (2004) pp. 3-34, ISSN 0167-2789

Stepney, S.; Braunstein, S. L.; Clark, J. A.; Tyrrell, A.; Adamatzky, A.; Smith, R. E.; Addis, T.; Johnson, C.; Timmis, J.; Welch, P.; Milner, R. & Partridge, D. (2006). Journeys in non-classical computation II: Initial journeys and waypoints, *International Journal of Parallel, Emergent and Distributed Systems*, Vol. 21, No. 2, (2006) pp. 97–125, ISSN 1744-5760

Strzałka, D. & Grabowski, F. (2008). Towards possible non-extensive thermodynamics of algorithmic processing – statistical mechanics of insertion sort algorithm, *International Journal of Modern Physics C*, Vol. 19, No. 9, (2008) pp. 1443–1458, ISSN 0129-1831

Strzałka, D. (2010). Paradigms evolution in computer science, *Egitania Scienca*, Vol. 6, (2010) p. 203, ISSN 1646-8848

Wegner, P. & Goldin, D. (2003). Computation Beyond Turing Machines, *Communications of the ACM*, Vol. 46, No. 4, (Apr. 2003), pp. 100–102, ISSN 0001-0782



Thermodynamics
Edited by Prof. Mizutani Tadashi

ISBN 978-953-307-544-0
Hard cover, 440 pages
Publisher InTech
Published online 14, January, 2011
Published in print edition January, 2011

Progress of thermodynamics has been stimulated by the findings of a variety of fields of science and technology. The principles of thermodynamics are so general that the application is widespread to such fields as solid state physics, chemistry, biology, astronomical science, materials science, and chemical engineering. The contents of this book should be of help to many scientists and engineers.