Research Journal of Engineering Sciences ISSN 2278 – 9472
Vol. **1(5),** 12-16, November **(2012)** *Res. J. Engineering Sci.*

Parameterized Complexity of Partition Sort for Negative Binomial Inputs

Singh Kumar Niraj[1], Pal Mita[2] and Chakraborty Soubhik[3]
[1]Department of Computer Science and Engineering, B.I.T. Mesra, Ranchi-835215, INDIA
[2,3]Department of Applied Mathematics, B.I.T. Mesra, Ranchi-835215, INDIA

Available online at: www.isca.in
Received 14th October 2012, revised 24th October 2012, accepted 1st November 2012

Abstract

The present paper makes a study on Partition sort algorithm for negative binomial inputs. Comparing the results with those for binomial inputs in our previous work, we find that this algorithm is sensitive to parameters of both distributions. But the main effects as well as the interaction effects involving these parameters and the input size are more significant for negative binomial case.

Keywords: Partition sort, average case, negative binomial distribution, parameterized complexity, computer experiments, factorial experiments.

Introduction

Partition Sort was introduced in a paper[1] which indicates a higher average case robustness compared to that of the popular quick sort algorithm[2]. This robustness was further reconfirmed in another paper[3] where it was subjected to an unconventional distribution (Cauchy) inputs apart from the parameterized complexity analysis over binomial inputs. Here in this paper we study this algorithm for Negative Binomial, NB(k, p), inputs. Our first study suggests an empirical O(nlogn) complexity for this distribution data for input size n. Next, as a parameterized complexity analysis, for different p, the probability of success, the average run times are observed and found to be a quadratic function of p, $Y_{avg}(n, k, p) = O_{emp}(p^2)$ for fixed n and k values. Further when k (the desired number of successes) is varied the average run times are found to be a cubic function of k, $Y_{avg}(n, k, p) = O_{emp}(k^3)$ for fixed n and p values. Lastly, to investigate the individual effect of number of sorting elements (n), negative binomial parameters k and p and also their joint effects, a 3-cube factorial experiment is conducted with three levels of each of the factors n, k and p. Comparing the result of factorial experiment of binomial distribution inputs in our previous work[3] and the result of factorial experiment of negative distribution inputs presently, it is observed that negative binomial distribution affects the mean time more than binomial distribution does and this is true for all the main effects as well as the interaction effects.

Material and Methods

The Algorithm: Partition Sort: Introduced in paper[1], Partition sort is a divide and conquer based robust and efficient comparison sort algorithm. The key sub routine 'partition' when applied on input A[1…….n] divides this list into two halves of sizes floor (n/2) and ceiling (n/2) respectively. The property of the elements in these halves is such that the value of each element in first half is less than the value of every element in the second half. The recursive call to Partition-sort routine finally yields a sorted sequence of data as desired. The worst case performance of Partition Sort is found to be O ($nlog_2^2n$), whereas the best case count is $\Omega(nlog_2n)$. The average case performance as estimated through the statistical bound estimate is empirical $O(nlog_2n)$ which is obtained by working directly on time. The reason for going with statistical bound may be found in paper[1] and the book[4].

Statistical Analysis: Negative Binomial (NB) distribution is obtained by performing independent Bernoullian trials (a Bernoullian trial is one that results in one of two possible outcomes which we call 'success' and 'failure') till the desired number of successes say 'k' are obtained, with p as the constant probability of successes in a trial. The number of failures preceding the k-th success is the required NB variate with parameters k and p. This section includes empirical results performed over Partition Sort algorithm for negative binomial inputs. The average case analysis is performed using statistical bound estimate (or empirical O). For definitions of statistical bound and 'empirical-O' one may consult the references[1,4]. It suffices to point out here that a statistical bound, unlike a count based mathematical bound which is consequently operation specific, is *weight based* and involves *collective consideration of all operations* into a conceptual bound. Average case analysis was done by directly working on program run time to estimate the weight based statistical bound over a finite range by running computer experiments[5,6]. In other words, the time of an operation is conceived as its weight.

The entry 'T' in the tables 1-3 denotes the mean time (in sec.) data that are averaged over 100 trial readings.

Remark: All the experiments have been carried out using PENTIUM 1600 MHz processor and 512 MB RAM.

Average Case Analysis Using Statistical Bound Estimate or Empirical O: The negative binomial distribution inputs are taken with parameters k and p, where k=1000 and p=0.5 are

fixed. The empirical result is shown in figure 1. Experimental result as shown in figure 1 is suggesting a step function that is trying to get close to O (nlogn) complexity. So we can safely conclude that

$Y_{avg}(n) = O_{emp}(nlogn)$.

The subscript "emp" implies an empirical and hence subjective bound-estimate[4].

Parameterized Complexity Analysis: The study of parameterized complexity is an essential activity of any statistical analysis for accessing the true potential of an algorithm's performance. Our previous related work[3] suggests that for Partition Sort, the parameters of the input distribution should also be taken into account for explaining its complexity, and not just the parameter characterizing the size of the input. The study in this section is accordingly devoted to parameterized complexity analysis whereby the sorting elements of Partition Sort come independently from a Negative Binomial (k, p) distribution. Here our interest lies in investigating the response behavior (which is CPU time in our case) as a function of input distribution parameter(s). The first systematic work on parameterized complexity was done by Downey and Fellows[7]. Other significant work on this topic may be found in the references[8, 9].

CASE (A): Parameterized Complexity Analysis when n and k are fixed while p is varying.

The first analysis is done for fixed n and k values, while the p value is varied in the range [0.1 to 0.9]. The experimental result is put into table 2 and the corresponding plot is given in figure 2.

Table-1
Mean time (in sec.) for negative binomial distribution inputs, p=0.5, k=1000

n	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
T	0.02022	0.04282	0.06946	0.09474	0.11876	0.14932	0.1753	0.20566	0.23446	0.26462

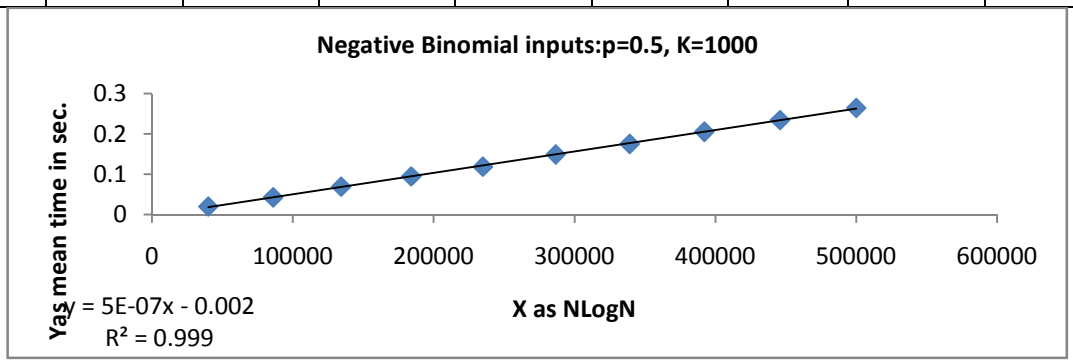

Figure-1
Regression model suggesting empirical O(nlogn) complexity

Table-2
Mean time (in sec.) for negative binomial distribution inputs, N=50000, k=1000

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T	0.14674	0.13838	0.13438	0.12594	0.12066	0.11776	0.11028	0.10432	0.09524

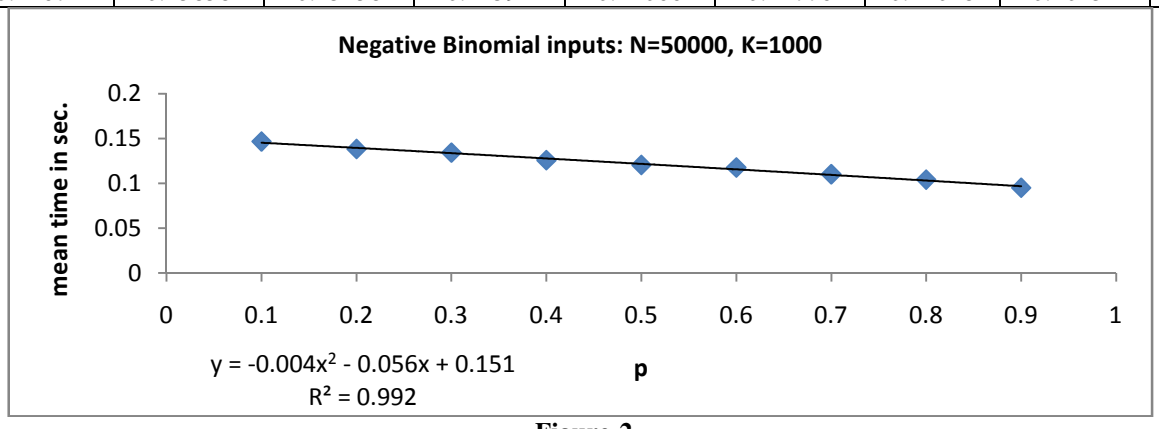

Figure-2
Second degree polynomial fit

The experimental result suggests an average function $Y_{avg}(n, k, p) = O_{emp}(p^2)$ for fixed n and k values.

CASE (B): Parameterized Complexity Analysis when n and p are fixed while k is varying

The next analysis is done for fixed n and p values, while the k value is varying in the range [100 to 5000]. The experimental result is put into table 3 and the corresponding plot is given in figure 3. The experimental result suggests an average function $Y_{avg}(n, k, p) = O_{emp}(k^3)$ for fixed n and k values.

We further performed the parameterized complexity analysis by conducting a 3-cube factorial experiment with three levels of each of the three factors n, m and p. All the three factors are found to be significant both individually and interactively. Table 4 contains the data for 3^3 factorial experiments for Partition Sort when input distribution is negative binomial.

Table-3
Mean time (in sec.) for negative binomial distribution inputs, N=50000, p=0.5

K	100	500	1000	2000	3000	4000	5000
T	0.1001	0.11398	0.11876	0.12594	0.12842	0.13098	0.13344

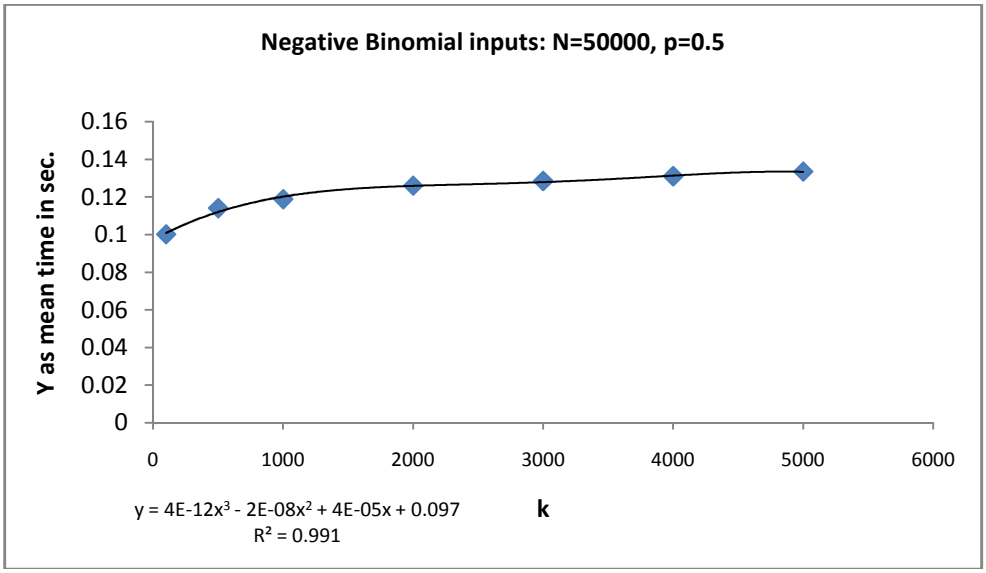

Figure-3
Third degree polynomial fit for varying k values

Table-4
Partition sort times in second Binomial (k, p) distribution input for various n (10000, 30000, 50000), k (1000, 3000, 5000) and p (0.2, 0.5, 0.8)

P=0.2			
N	k=1000	k=3000	k=5000
10000	0.01966	0.02324	0.02376
30000	0.0778	0.07878	0.08214
50000	0.13844	0.1415	0.14746
P=0.5			
N	k=1000	k=3000	k=5000
10000	0.02022	0.02164	0.02094
30000	0.06946	0.0723	0.075
50000	0.11876	0.12842	0.13344
P=0.8			
N	k=1000	k=3000	k=5000
10000	0.0172	0.01932	0.01832
30000	0.05954	0.06334	0.06676
50000	0.1031	0.11278	0.11784

Results and Discussion

Result for 3^3 factorial experiment: Factorial experiment is performed using MINITAB statistical package version 15. The analysis data obtained is put in the following result (table 5, 6, 7).

Experimental results reveal that Partition sort is highly affected by the main effects n, p and k. It is interesting to note that all interactions are found significant in Partition Sort. Moreover, it is observed that negative binomial distribution affects the algorithm's performance (mean time) more than binomial distribution does[3] not only for the main effects but also for the interaction effects. In particular, the main effect p in negative binomial is remarkably sensitive than that in binomial inputs.

Theoretical support for our arguments: Although the statistical approach is the ideal one for verifying the significance of the treatments, here the parameters n, p and k and their interactions, it is not very difficult to theoretically justify why these parameters are important. The time of the code will involve not only the time of the comparisons but also the time for the interchanges. The probability for an interchange is $P[a(i)>a(j)]=\sum P[a(i)>r, a(j)=r]$ where the summation is over r=0, 1, 2…..Since both a(i) and a(j) are independent negative binomial (k, p) variates, the probability inside the summation will be the product of the individual probabilities and will automatically involve k and p as they appear in the probability mass function of a negative binomial (k, p) variate, that is to say, when we consider P[a(i)>r]P[a(j)=r], the first probability is $\sum {}^{x+k-1}C_{k-1}\, p^k\, (1-p)^x$ where the summation is over x=r+1, r+2…. The second probability is simply ${}^{r+k-1}C_{k-1}\, p^k\, (1-p)^r$. Given that the expected number of interchanges will be the product of the expected number of comparisons multiplied by the probability of an interchange in a comparison, and given further that the former will definitely involve n, the input size, we have that all the parameters n, k and p are important at least singularly in explaining the time complexity. The question is: are they important interactively as well? The answer, through factorial experiments, is that they are! Similar arguments can be given for the Binomial case where we shall have the Binomial probability mass function but the relative influence of the underlying parameters can be compared best through statistical approach. This concludes our discussion.

Table-5
Multilevel Factorial Design

Factors	3
Replicates	3
Base runs	27
Total runs	81
Base blocks	1
Total blocks	1
Number of levels	3, 3, 3

Table-6
General Linear Model: y versus n, p, k:

Factor	Type	Levels	Values
n	fixed	3	1, 2, 3
p	fixed	3	1, 2, 3
k	fixed	3	1, 2, 3

Table-7
Analysis of Variance for y, using Adjusted SS for Tests

Source	DF	Seq SS	Adj SS	Adj MS	F	P
n	2	0.1528421	0.1528421	0.0764211	1.77367E+08	0.000
p	2	0.0039854	0.0039854	0.0019927	4624925.02	0.000
k	2	0.0006386	0.0006386	0.0003193	741021.05	0.000
n*p	4	0.0016832	0.0016832	0.0004208	976642.57	0.000
n*k	4	0.0002823	0.0002823	0.0000706	163825.64	0.000
p*k	4	0.0000188	0.0000188	0.0000047	10901.72	0.000
n*p*k	8	0.0000517	0.0000517	0.0000065	14988.37	0.000
Error	54	0.0000000	0.0000000	0.0000000		
Total	80	0.1595021				

S = 0.0000207573 R-Sq = 100.00% R-Sq(adj) = 100.00%

Conclusion

Our experimental results and its subsequent analysis reveals that the Partition Sort exhibits robustness in the average case for negative binomial inputs and hence can serve as a better alternative than the popular quick sort algorithm. Apart from the robustness issue we also found it to be sensitive to input distribution parameters and hence a potential candidate to the study of parameterized complexity analysis. For n independent Negative Binomial (k, p) inputs, all the three factors are significant both independently and interactively. All the two factor interactions n*k, n*p and k*p and even the three factor n*k*p is significant. Moreover, it is observed that the algorithm is more sensitive to negative binomial distribution inputs than to binomial distribution inputs[3] for both main effects and interaction effects. Our finding regarding the measure of parametric influence is an experimental approach. Although such measures can be accomplished through some suitable theoretical analysis, we are still dependent on statistical tools so as to confirm the significance of their interactions. Besides, theoretical analysis is count based rather than weight based and does not provide the statistical bounds which are ideal for the average case[3].

References

1. Singh N.K., and Chakraborty S., *Partition Sort and its Empirical Analysis*, V.V. Das and N. Thankachan (Eds.): CIIT 2011, CCIS 250, 340–346, 2011, © Springer-Verlag Berlin Heidelberg **(2011)**

2. Sourabh S.K. and Chakraborty S., How robust is quicksort average complexity? arXiv:0811.4376v1 [cs.DS], *Advances in Mathematical Sciences Jour*. (to appear)

3. Singh N.K., Pal M. and Chakraborty S., Partition sort revisited: Reconfirming the robustness in average case and much more, *International Journal on Computer Science, Engineering and Applications*, **2(1)**, 23-30 **(2012)**

4. Chakraborty S. and Sourabh S.K., A Computer Experiment Oriented Approach to Algorithmic Complexity, Lambert Academic Publishing **(2010)**

5. Fang K.T., Li R. and Sudjianto A., Design and Modeling of Computer Experiments, Chapman and Hall **(2006)**

6. Sacks J., Weltch W., Mitchel T. and Wynn H., Design and Analysis of Computer Experiments, *Statistical Science,* **4(4)**, **(1989)**

7. Downey R.G. and Fellows M.R., *Parameterized Complexity*, Springer **(1999)**

8. Mahmoud H., Sorting: A Distribution Theory, John Wiley and Son **(2000)**

9. Chakraborty S., Sourabh S.K., Bose M. and Sushant K., Replacement sort revisited: The "gold standard" unearthed!, *Applied Mathematics and Computation*, **189(2)**, 384-394 **(2007)**