The Differencing Method of Set Partitioning

*Narendra Karmarkar and Richard M. Karp**

University of California
Berkeley, CA 94720

1. Introduction

We consider the following set partitioning problem:

Given a finite set $S \subseteq [0, 1]$ of real numbers, partition S into k subsets $A_1, A_2, \cdots A_k$ so as to minimize

$$D(A_1, A_2, \cdots A_k) = \max_i \left\{ \sum_{x \in A_i} x \right\} - \min_i \left\{ \sum_{x \in A_i} x \right\}$$

This is related to the well-known problem of scheduling tasks on k identical processors to minimize the completion time of the last task completed. Since this is an NP-complete problem, we do not expect to find a polynomial time algorithm unless $P = NP$. We present an $O(n)$-time algorithm which produces a partition with $D = O(n^{-c \log n})$ $c > 0$, except in pathological cases. In order to show that such pathological cases are extremely rare, we use the following probabilistic model: The input to the problem is a set of n independent and identically distributed random variables, with density function $f(x)$. We allow any arbitrary function $f(x)$, subject to the Lipschitz condition:

$$\exists B > 0, s.t. \ \forall x, y \in [0, 1], |f(x) - f(y)| \leq B |x - y|$$

Under these assumptions, we prove that the algorithm performs as claimed with probability $\to 1$, as $n \to \infty$.

The algorithm is based on a new method of combining partial solutions to the partitioning problem based on "Differencing Operations." These operations seem to outperform any previously known heuristic methods such as LPT (Largest Processing Time first) [1] or MULTIFIT [2]. Under similar probabilistic assumptions, LPT or MULTIFIT cannot be expected to give a partition better than $D = O(\frac{1}{n})$.

Finally, we remark that many heuristic modifications to our algorithms are possible, which do not change the asymptotic behaviour of the algorithm, but can make it more practical for small values of n. Also, the differencing operations reported in this paper can be used as subroutines in other more complicated heuristic algorithms. Thus, the main contribution of this paper is two-fold: A set of differencing operations for the set partitioning problem, and a provably good algorithm for partitioning based on these operations.

*Research supported by NSF Grant MCS81-05217

2. Partitioning Algorithm for 2-Processor Case

We now restrict attention to the case $k = 2$. i.e., we are interested in partitioning a set $S \subseteq [0, 1]$ into two subsets A, B, so as to minimize $D(A, B)$ approximately.

In Section 2.1 we discuss the main concepts used in the partitioning algorithm in an informal way. Section 2.2 contains the description of the algorithm and statements of the inductive assertions about the algorithm. Proofs of these assertions are given in Section 2.3.

2.1. An informal view of the main concepts

The basic operation used in this algorithm consists of selecting a pair $a, b \in S$, such that $|a - b|$ is small, and restricting the solution to those partitions in which a and b appear on opposite sides. The new (restricted) problem is equivalent to partitioning

$$S' = S - \{a, b\} \cup \{|a - b|\}.$$

From a partition (A', B') of S', it is easy to construct a partition (A, B) of S, so that $D(A, B) = D(A', B')$.

e.g., suppose $a > b$ and $|a - b| \in A'$

then $A = A' - \{|a - b|\} + \{a\}$

$B = B' + \{b\}$

gives the desired partition.

The following example illustrates an algorithm based on repeated application of this differencing operation:

<u>Algorithm A</u>

While $|S| > 1$ do begin
  pick the largest two numbers $a, b \in S$
  $S \leftarrow S - \{a, b\} + \{|a - b|\}$.
end

This algorithm terminates when $|S| = 1$. The last number is $D(A, B)$. It is trivial to construct the actual partition (A, B) by backtracing through the sequence of differencing operations.

<u>Example:</u>

Let $S = \{3, 6, 13, 20, 30, 40, 73\}$

Set	a	b	b-a	Partition	
{3, 6, 13, 20, 40, 73}	40	73	33	{6, 13, 73}	{3, 20, 30, 40}
{3, 6, 13, 20, 30, 33}	30	33	3	{6, 13, 33}	{3, 20, 30}
{3, 3, 6, 13, 20}	13	20	7	{3, 6, 13}	{3, 20}
{3, 3, 6, 7}	6	7	1	{3, 6}	{3, 7}
{1, 3, 3}	3	3	0	{3}	{1, 3}
{0, 1}	0	1	1	{0}	{1}
{1}				φ	{1}

In the first pass, the algorithm works down the table filling in the columns headed 'set', a, b, and $b - a$. Then it proceeds bottom-up to fill in the column headed 'partition'. The final result is

$$A = \{6, 13, 73\}, B = \{3, 20, 30, 40\}, D(A, B) = 1$$

<u>Algorithm B</u>

Here is a second example of an algorithm based on the differencing operation:

(Assume $n = 2^k$)

Sort the set S . Pair the largest two numbers, the next largest two, and so on. Differencing each pair, we get a set of size $\frac{|S|}{2} = 2^{k-1}$, and it appears that the order of magnitude of the numbers is reduced approximately by a factor of n . Repeating this operation k times, we might hope to create a partition with

$$D(A, B) = O(n^{-k}) = O(n^{-\log n}).$$

There are a few problems with this scheme:

1. n may not be a power of two.
2. Although most differences produced after one phase are of the order of $\frac{1}{n}$, a few of them could be much larger.
3. The distribution of the numbers after one phase is not the same as the one we started with. In particular the numbers in the new distribution are not independent random variables. In order to be able to prove any inductive assertions about the behaviour of the algorithm, the same distribution should be reproduced after one phase, or some relevant property of the distribution should be reproduced.

Hence we introduce the following modifications:

1. Random Pairing: Instead of pairing the numbers according to the scheme above, we divide the interval [0, 1] into $N = \frac{|S|}{K}$ subintervals of equal length. (K is a constant to be defined later.) Then we pick pairs of numbers at random from each subinterval, until each subinterval has at most one number left.

Each of the differences created this way is smaller than $\frac{K}{n}$. Assuming that the initial distribution $f(x)$ is reasonably smooth, the distribution of numbers in each subinterval is approximately uniform. (It can be treated as if it were exactly uniform by a resampling technique described later.) Therefore, the difference of a pair randomly chosen from each subinterval is triangularly distributed over $[0, \frac{K}{n}]$. Also, the differences are independent random variables. The triangular distribution is again smooth enough that during the next phase, subdivision of $[0, \frac{K}{n}]$ gives an approximately uniform distribution in each subinterval. Thus the two essential properties of the distribution, viz smoothness and independence are reproduced by this scheme.

2. Compaction

After random pairing, all subintervals with an odd number of points are left with one point each. To get rid of these odd points, we introduce a compaction operation. First, we execute Algorithm A on the set C' of odd points. Let c be the last single number left. With high probability, c is less than $\beta \ln n \cdot \frac{K}{n}$ for some β. This is proved as follows: If we divide [0, 1] into a sequence of intervals of with endpoints $L, \sqrt{2}L, 2L, 2\sqrt{2}L \cdots$, and each of these intervals has at least one point, then the result of applying Algorithm A is less than L. An interval of length $L = \frac{K}{n} \cdot \beta \ln n$ contains $\beta \ln n$ subintervals of length $\frac{K}{n}$ each. The probability that each of these intervals contains an even number of points is very small. Hence with high probability, each interval contains at least one odd point.

In order to bring c into the range $(0, \frac{K}{n})$ we difference it against points from $(0, \frac{K}{n})$ which are otherwise meant to be used as input to the next phase. Since the mean of the triangular distribution over $(0, \frac{K}{n})$ is $\frac{1}{3}\frac{K}{n}$, the expected number of such points required to compensate for c is $\frac{3n}{K} \cdot c < 3\beta \ln n$. The actual number of such points used up can be shown to be less than $\beta' \ln n$ with high probability, using the Chernoff Bound [3], and choosing β' suitably.

3. Resampling:

This is a conceptual operation used in the analysis and does not occur in the actual algorithm.

Since the density function $f(x), x \in [0, \alpha_m]$, of numbers input to the m^{th} phase is smooth (i.e., satisfies a Lipschitz condition), it can be treated as if it were uniform in each subinterval of length $\frac{K}{n}$. The resampling process is a technique to rigorously analyze the effect of this approximation. It takes a set of points S in an interval I, distributed independently according to a density function $f(x)$, and labels a large subset of S as "good" and the remaining as "bad." The density function of the subset G of good points is uniform in each subinterval of interest. The size of the bad set of points B is extremely small with high probability. The labels created by resampling are used only in the analysis and are not required in the actual algorithm. This is similar to radioactive tagging of chemical molecules of interest so that they can be observed in a chemical reaction, without interfering with the reaction itself.

2.2. Description of the algorithm

The algorithm is divided into a sequence of phases. The input to the m^{th} phase is a set of numbers $S_m \subseteq [0, \alpha_m]$. The output of each phase constitutes the input to the next phase. For the purpose of analysis, each number in S_m is labelled as "good" or "bad," forming a partition $S_m = G_m \cup B_m$. Initially,

$$\alpha_1 = 1, \; S_1 = S, \; |S_1| = n, \; B_1 = \varphi.$$

Each phase consists of four operations.

1. Partitioning
2. Conceptual Resampling
3. Random Differencing
4. Compaction

In the following description, of the m^{th} phase, the suffix m will often be dropped, when there is no danger of ambiguity. Thus S_m, α_m will sometimes be denoted by S, α.

1. Partitioning

Let $n_m = |S_m|$, $N_m = \frac{n_m}{K}$, $\alpha_{m+1} = \frac{\alpha_m}{N_m}$

(K is a constant which is required to satisfy certain bounds described later). Divide the interval $[0, \alpha_m]$ into N_m subintervals of length α_{m+1} each.

i.e.,

$$[0, \alpha_m] = \bigcup_{i=1}^{N_m} I_i \quad I_i = [(i-1)\,\alpha_{m+1},\ i\cdot\alpha_{m+1}]$$

Construct the corresponding partition of S into $S_1, S_2 \cdots S_{N_m}$.

$S_i \leftarrow S \cap I_i$

2. Resampling

Recall that the input set S is partitioned into good and bad points.

$$S = G \cup B$$

Let $f(x)$ be the density function of points in G, (During the first phase, this is the same as input density function, in subsequent phases, it is triangular, i.e., $f(x) = (\frac{2(\alpha_m - x)}{\alpha_m^2})$.

Let

$$f_i = \min_{x \in I_i} f(x)$$

Define a function $g(x)$, which approximates $f(x)$ from below and remains constant in each of the subintervals I_i.

$$g(x) = f_i \text{ when } x \in I_i$$

A point $x \in G$ is relabelled as "good" with probability $\frac{g(x)}{f(x)}$ (when $f(x) > 0$) and "bad" with probability $1 - \frac{g(x)}{f(x)}$, independently of other random variables. Hence the density function for points labelled as "good" is uniform in each subinterval.

3. Random Differencing

From each subinterval we repeatedly select a pair of points at random and take their absolute difference.

```
For all i ∈ {1, 2,·· N} do begin
      S'_i ← φ
      while |S_i| ≥ 2 do begin
            pick x, y ∈ S_i at random
            S_i ← S_i − {x, y}
            S'_i ← S'_i + {|x − y|}
            < if x, y ∈ G, then label |x − y| as "good" else label it as "bad" >
      end
end
```

$$S' \leftarrow \bigcup_{i=1}^{N} S'_i$$

$$C' \leftarrow \bigcup_{i=1}^{N} S_i$$

S' is the set of differences created by this process and C' is the set of "odd" points left-over. Note that good points in S' are independent random variables with a triangular density function.

4. Compaction

The input to the compaction process is a set $C' \subseteq [0, \alpha_m]$ of "odd" points and a set $S' \subseteq [0, \alpha_{m+1}]$ of differences created during random differencing.

The first step of compaction is to apply algorithm A to the set C' and reduce it to a single point, say c. (Assume $C \neq \varphi$).

The second step is to difference c against randomly chosen points from S' until $c \le \alpha_{m+1}$.

```
While c > α_{m+1} do begin
    pick x ∈ S' at random
    S' ← {x}
    c ← x
end
```

The set $S' \cup \{c\}$ forms input to the next phase.

$$S_{m+1} = S' \cup \{c\}$$

This completes the description of the m^{th} phase. The algorithm stops when there is only one number left, which is $D(A, B)$. The actual partition can be constructed by backtracing through the sequence of differencing operations.

2.3. Inductive assertions about the algorithm

Theorem 1:

Recall that $D(A, B)$ is the difference between the two sides of the partition.

$\exists$ a constant $\alpha > 0$ such that with probability $\to 1$ as $n \to \infty$

$D(A, B)$ satisfies the following bound:

$$D(A, B) \le n^{-\alpha \log n}$$

Theorem 2:

Let M be the total number of phases.

Recall that the input to the m^{th} phase is a set of n_m points in the interval $[0, \alpha_m]$.

let ϑ be such that $0 < \vartheta < \frac{1}{2} - \frac{1}{2k}$ following statements are jointly true with probability $\to 1$ as $n \to \infty$

1. There are at least $M' = \left\lceil \frac{\frac{1}{4}\ln n}{\ln \frac{1}{\vartheta}} \right\rceil$ phases. i.e., $M \ge M'$

2. $n_m \ge \vartheta^{m-1} \cdot n$, $\alpha_{m+1} \le e^{-\frac{m}{2}\ln n}$ for $m = 1, 2, \cdots M'$.
i.e., the number of points decreases less rapidly than a certain geometric series.

Theorem 3:

The set $S_m \subseteq [0, \alpha_m]$, the input to the m^{th} phase, is partitioned into "good" and "bad" numbers.

$$S = G_m \cup B_m$$

Let $g = |G_m|$ $b_m = |B_m|$

For any $\alpha > 0$, $\exists \beta_1 > 0$ such that, for sufficiently large n, and for $1 \le m \le M'$,

$$Pr\left\{ \begin{array}{l} g_{m+1} \ge \vartheta^m \cdot n \\ b_{m+1} \le (m+1)\beta_1 \ln n \end{array} \middle| \begin{array}{l} g_m \ge \vartheta^{m-1} \cdot n \\ b_m \le m \cdot \beta_1 \ln n \end{array} \right\} \ge 1 - n^{-\alpha}.$$

<u>Theorem 4:</u>

Define l_1 = the number of good points relabelled as bad during the resampling process.

With very high probability $l_1 = O(\ln n)$. More precisely, for any $\alpha > 0$, $\exists\ \beta > 0$ such that, for sufficiently large n,

$$Pr\{ l_1 \geq \beta \ln n \} \leq n^{-\alpha}$$

<u>Theorem 5:</u>

Define c to be the single bad number at the end of the first step of compaction.

Then $\frac{c}{\alpha_{m+1}} = O(\ln n)$ with high probability. More precisely,

1. for any $\alpha > 0$, $\exists\ \beta > 0$ such that

$$Pr\{ c > \beta \ln n \cdot \alpha_{m+1} \} \leq n^{-\alpha}$$

Define l_3 = the number of good points from S' used up during the second step of compaction to compensate for the bad point c.

Then $l_3 = O(\ln n)$ with high probability. More precisely,

2. for any $\alpha > 0$, $\exists\ \beta > 0$ such that

$$Pr\{ l_3 > \beta \ln n \} \leq n^{-\alpha}$$

2.4. Analysis of the algorithm

In this section we give proofs of Theorem 5 through 1, in that order.

Proof of Theorem 5: In preparation for the analysis of the compaction operation, we require a lemma about algorithm A.

Lemma 5.1. Let $S \subseteq [0,1]$ be a set of points input to Algorithm A, and let d be the final outcome, i. e. the last single number left. Subdivide the interval $[0,1]$ into a sequence of intervals

$$I_0 = [0 = \alpha_0, \alpha_1],\ I_2 = [\alpha_1, \alpha_2], \ldots\ldots I_m = [\alpha_m, \alpha_{m+1} = 1]$$

so that $\alpha_{j+1} \leq \alpha_j + \alpha_{j-1}$, $j = 2, \ldots m$.

Suppose $S \cap I_j \neq \varphi$ for $j = 1,2,\ldots m$ i. e. each of the interval I_j contains at least one point from S.

Then $d \in I_0 \cup I_1$.

Proof: Observe that

1. $x, y \in I_{j+1} \Rightarrow |x-y| \in I_0 \cup I_1 \ldots \cup I_j$, $j > 1$
2. $x \in I_{j+1}, y \in I_j \Rightarrow |x-y| \in I_0 \cup I_1 \ldots I_j$, $j > 1$

At any time during the execution of algorithm A, the following property holds:

All intervals up to the last non-empty interval contain at least one point each. We can prove this inductively working backwards:

If the last non-empty interval I_m contains an even number of points, say $2k$, then after k differencing operations it becomes empty and the intervals $I_0, I_1, \ldots I_{m-1}$ continue to have at least one point each.

If I_m contains an odd number of points, say $2k+1$, then after k differencing operations it will contain a single point which will be differenced with a point from I_{m-1}, since I_{m-1} is non-empty. Then I_m becomes empty and intervals $I_0, I_1, \ldots I_{m-2}$ continue to have at least one point each. I_{m-1} may or may not be empty. In either case, all intervals up to the last non-empty interval contain at least one point.

Lemma 5.2. Suppose we throw n balls into $l+1$ boxes $B_1, B_2, \ldots B_l$ and B. Let the probability that a given ball lands in box B_i be $\frac{p_i}{n}$, independently of other random variables. (The probability that a given ball goes in box B is $1 - \sum \frac{p_i}{n}$).

Further suppose that the p_i's are bounded from below by a positive constant p.

and that

$$\exists p > 0 \text{ such that } p_i \geq p \quad \forall i = 1, \ldots l.$$

Let q_φ be the probability that every box except B contains an even number of balls. (The significance of the suffix φ will become clear in the next lemma).

Then $q_\varphi \leq 2 \left[\frac{1+e^{-2p}}{2}\right]^l$ for sufficiently large n.

Proof: Let m_i = number of balls in B_i, $i = 1, \ldots l$.

Let $m = \sum_{i=1}^{l} m_i$

Hence B contains $n-m$ balls.

Then q is given by

$$q_\varphi = \sum_{m=0}^{n} \binom{n}{m} \left(1 - \sum \frac{p_i}{n}\right)^{n-m} \cdot Q_m$$

where Q_m is given by

$$Q_m = \sum_{\substack{m_i = 0 \\ \sum m_i = m}}^{m} \frac{m!}{m_1! m_2! \ldots m_l!} \left(\frac{p_1}{n}\right)^{m_1} \cdots \left(\frac{p_l}{n}\right)^{m_l} \cdots \left(\frac{1+(-1)^{m_1}}{2}\right) \cdots \left(\frac{1+(-1)^{m_l}}{2}\right)$$

$$= \sum_{m=0}^{n} \binom{n}{m} \left(1 - \frac{\sum p_i}{n}\right)^{n-m} \sum_{\substack{\text{all sign} \\ \text{combinations}}} \frac{1}{2^l} \left(\frac{\pm p_1 \pm p_2 \ldots \pm p_l}{n}\right)^m$$

$$= \frac{1}{2^l} \sum_{\substack{\text{all sign} \\ \text{combinations}}} \left[1 - \frac{\sum p_i}{n} + \left(\frac{\pm p_1 \pm p_2 \ldots \pm p_l}{n}\right)\right]^n$$

Let $S \subseteq \{1,2,...l\}$ be the subset of indices corresponding to negative terms. Then

$$q = \frac{1}{2^l} \sum_{all\ S \subseteq \{1,2,...l\}} \left[1 - \frac{2}{n} \sum_{i \in S} p_i\right]^n$$

Using the fact that for

$$\alpha > 0, \ => \ \left(1 - \frac{\alpha}{n}\right)^n \le 2e^{-alpha} \quad \text{for} \quad n \ge \alpha^2,$$

$$\left[1 - \frac{2}{n} \sum_{i \in S} p_i\right]^n \le 2. \exp\{-2 \sum_{i \in S} p_i\}$$

since

$$p_i \ge p, \quad \exp\{-2 \sum_{i \in S} p_i\} \le \quad \exp\{-2 \, . \, |S| \, . \, p\}$$

$$\therefore q_\varphi \le \frac{1}{2^l} . 2 \sum_{i=0}^{l} \binom{n}{i} \exp(-2 i p) = \frac{2}{2^l} \left[1 + e^{-2p}\right]^l$$

Lemma 5.3. As in Lemma 5.2, we throw balls into boxes with given probabilities.
Let $T \subseteq \{1,2,...l\}$.
Let q_T = probability that all the boxes B_i, $i \in T$ contain an odd number of balls, and all B_i, $i \in \{1,2,...l\} - T$ contain an even number of balls.
Then $q_T \le q_\varphi$

Proof: Define

$$f_e(m) = \frac{1 + (-1)^m}{2} = \begin{cases} 1 & \text{if } m \text{ is even} \\ 0 & \text{if } m \text{ is odd} \end{cases}$$

and

$$f_0(m) = \frac{1 - (-1)^m}{2} = \begin{cases} 1 & \text{if } m \text{ is odd} \\ 0 & \text{if } m \text{ is even} \end{cases}$$

Then

$$q_T = \sum_{m=0}^{n} \binom{n}{m} \left(1 - \frac{\sum p_i}{n}\right)^{n-m} . \ Q_m .$$

where Q_m is given by

$$Q_m = \sum_{\substack{m_i = 0 \\ \sum m_i = m}}^{m} \frac{m!}{m_1! \, m_2! \ldots m_l!} \left(\frac{p_1}{n}\right)^{m_1} \left(\frac{p_2}{n}\right)^{m_2} \ldots \left(\frac{p_l}{n}\right)^{m_l} . \left[\prod_{i \in T} f_0(m_i)\right] \left[\prod_{i \notin T} f_e(m_i)\right].$$

Simplifying as before,

$$q_T = \frac{1}{2^l}\left\{\sum_{\substack{\text{all } S\subseteq\{1,2,\ldots l\}\\ |S\cap T|\ \text{even}}}\left[1-\frac{2}{n}\sum_{i\in S}p_i\right]^n - \sum_{\substack{\text{all } S\subseteq\{1,2,\ldots l\}\\ |S\cap T|\ \text{odd}}}\left[1-\frac{2}{n}\sum_{i\in S}p_i\right]^n\right\}$$

and

$$q_\varphi = \frac{1}{2^l}\sum_{\text{all } S\subseteq\{1,2,\ldots l\}}\left[1-\frac{2}{n}\sum_{i\in S}p_i\right]^n$$

Since each of the terms $\left[1-\frac{2}{n}\sum p_i\right]^n$ is positive, $q_T \le q_\varphi$.

Lemma 5.4. Let $S = G\cup B \subseteq [0,1]$ be a set of points, where $|G| = n$, and the points in G are independent random variables with the triangular distribution over $[0,1]$. Let the interval $[0,1]$ be subdivided into intervals of length $\frac{k}{n}$.

Let $I \subseteq [0, \frac{1}{\sqrt{2}}]$ be a collection of l such subintervals, $I_1, I_2, \ldots I_l$ i. e. $I = I_1 \cup I_2, \ldots I_l$.

Let q be the probability that each subinterval I_i contains an odd number of points from S.

Then $q \le 2\left[\frac{1+e^{-2p}}{2}\right]^l$ where $p = kf_\Delta\left(\frac{1}{\sqrt{2}}\right) = 2k\,(1-\frac{1}{\sqrt{2}})$ and $f_\Delta(x)$ is the triangular distribution over $[0,1]$.

Proof: $|S\cap I_i| = |G\cap I_i| + |B\cap I_i|$

$|S\cap I_i|\ mod\ 2 = |G\cap I_i|\ mod\ 2 + |B\cap I_i|\ mod\ 2$

Define $T \subseteq \{1,2,\ldots l\}$ so that $T = \{i \mid |B\cap T_i|\ is\ even\}$

$$\therefore |S\cap T_i|\ \textit{is odd for all } i \Leftrightarrow \begin{cases}|G\cap I_i|\ \textit{is odd for } i\in T\\ |G\cap I_i|\ \textit{is even for } i\in\{1,2,\ldots l\}-T\end{cases}$$

$q = p_r\{|S\cap I_i|\ \textit{is odd for all } i\} = p_r\{|G\cap I_i|\ \textit{is odd iff } i\in T\}$

for any interval $I_i \subseteq [0, \frac{1}{\sqrt{2}}]$, the probability $\frac{p_i}{n}$ that a given point in set G belongs to I_i, is given by

$$\frac{p_i}{n} = \int_{I_i} f_\Delta(x)\,dx \ge \min_{x\in I_i} f_\Delta(x)\cdot \textit{length}\,(I_i) \ge f_\Delta\left(\frac{1}{\sqrt{2}}\right)\cdot\frac{k}{n}$$

$$\therefore p_i \ge k\,.\,f_\Delta\left(\frac{1}{\sqrt{2}}\right) = 2k\left(1-\frac{1}{\sqrt{2}}\right) = p\ (say)$$

Applying Lemma 5.3 and Lemma 5.2, the probability that

$$\{|G \cap I_i| \text{ is odd iff } i \in T\} = q_T \le q_\varphi \le 2\left[\frac{1+e^{-2p}}{2}\right]^l$$

$$\therefore q \le 2\left[\frac{1+e^{-2p}}{2}\right]^l \quad \text{where } p = 2k\left(1-\frac{1}{\sqrt{2}}\right)$$

Lemma 5.5 (Theorem 5.1). Let C' be the input to the compaction process and let c be the result of applying algorithm A to C'. Then, for any $\alpha > 0$ $\exists$ $\beta > 0$ s.t.

$$p_r\{c > \beta \ln n \cdot \frac{k}{n}\} \le n^{-\alpha}$$

Proof: Let the interval $[0, 1]$ be divided into subintervals of length $\frac{k}{n}$ as before. We group these subintervals into a sequence of intervals $I_0, I_1 \ldots I_m$ with the following properties. Let l_i = number of subintervals contained in I_i. Then

1. $l_0 \ge \frac{\beta \ln n}{(1+\sqrt{2})}$ the constant β will be specified later
2. $l_j = \lfloor \sqrt{2}\, l_{i-1} \rfloor$

$\therefore$ $1.4 \le \frac{l_j}{l_{j-1}} \le \sqrt{2}$ when n is sufficiently large.

Now we apply Lemma 5.4 to each interval I_i, $i < m$. Note that

1. $I_i \subseteq [0, \frac{1}{\sqrt{2}}]$ for $i = 0, 1, \ldots m-1$
2. $C' \cap I_i = \varphi \Leftrightarrow$ each subinterval of I_i contains an even number of points from $S = G \cup B$, the input to the phase.

$$p_r\{C' \cap I_i = \varphi\} \le 2\left[\frac{1+e^{-2p}}{2}\right]^{l_i} \quad p = 2k\left(1-\frac{1}{\sqrt{2}}\right)$$

$$\therefore \quad p_r\{\exists i \in \{0, 1, \ldots m-1\} \text{ s. t. } C' \cap I_i = \varphi\} \le \sum_{i=0}^{m-1} 2\left[\frac{1+e^{-2p}}{2}\right]^{l_i}$$

If $C' \cap I_i \ne \varphi$ $\forall i = 0, 1, \ldots m-1$, then applying Lemma 5.1, $c \le \beta \ln n \cdot \frac{k}{n}$.

$$\therefore \quad p_r\{c > \beta \ln n \cdot \frac{k}{n}\} \le \sum_{i=0}^{m-1} 2\left[\frac{1+e^{-2p}}{2}\right]^{l_i}$$

$$\because \quad \frac{l_j}{l_{j-1}} \ge 1.4 \rightarrow l_j \ge (1.4)^j\, l_0$$

$$\text{let } T = \left[\frac{1+e^{-2p}}{2}\right]^{l_0}$$

$$\therefore \sum_{i=0}^{m-1} 2\left[\frac{1+e^{-2p}}{2}\right]^{l_j} \le \sum_{i=0}^{m-1} 2\left[\left[\frac{1+e^{-2p}}{2}\right]^{l_0}\right]^{1.4^i}$$

$$\leq 2[T + T^{1.4} + T^{(1.4)^2} \ldots]$$
$$\leq 2[T + T + T^2 + T^3 \ldots]$$
$$\leq 2T\left[1 + \frac{1}{1-T}\right]$$
$$\leq 6T \left[since \lim_{n \to \infty} T = 0,\ T < \frac{1}{2} \text{ for } sufficient\ large\ n\right]$$
$$\ln(6T) = \ln 6 + l_c \ln\left[\frac{1+e^{-2p}}{2}\right] = \ln 6 - \left[\frac{2}{1+e^{-2p}}\right] \cdot \frac{\beta \ln n}{(1+\sqrt{2})}$$

for any $\alpha > 0$, the right hand side can be made smaller than $-\alpha \ln n$, by suitable choice of β, and sufficiently large n.

$\therefore \exists\ \beta > 0$, s.t.

$$p_r\{c > \beta \ln n \,.\, \frac{k}{n}\} \leq n^{-\alpha}$$

This proves Theorem 5.1 by scaling by a factor of αm, since $\frac{k}{n} = \frac{\alpha_{m+1}}{\alpha_m}$.

Lemma 5.6 (Theorem 5.2). for any $\alpha > 0\ \exists\ \beta > 0$ s.t. $p_r\{l_3 > \beta \ln n\} \leq n^{-\alpha}$.

Proof: for given $\alpha > 0$, choose β_1 (using Lemma 5.5) so that $p_r\{c \,.\, \beta_1 \ln n > \alpha_{m+1}\} \leq n^{-\alpha}$. Suppose $c \leq \beta_1 \ln n \,.\, \alpha_{m+1}$

Let $A \subseteq S'$ be the set of points used during the second step of compaction. Let $l_3 = |A \cap G'|$. Points in $A \cap G'$ are independently and triangularly distributed over $[0, \alpha_{m+1}]$.

Let S_l = sum of l independent random variables triangularly distributed over $[0, 1]$.

$$p_r\{\sum_{x \in A} x \leq c\} = p_r\{\sum_{x \in A \cap G'} x + \sum_{x \in A \cap B'} x \leq c\}$$
$$\leq p_r\{\sum_{x \in A \cap G'} x \leq c\}$$
$$= p_r\{s_{l_3} \cdot \alpha_{m+1} \leq c\}$$
$$\leq p_r\{s_{l_3} \cdot \alpha_{m+1} \leq \beta_1 \ln n.\ \alpha_{m+1}\}$$
$$= p_r\{s_{l_3} \leq \beta_1 \ln n\}$$

$$p_r\{l_3 \geq \beta'' \ln n \,|\, c < \alpha_{m+1}\, \beta_1 \ln n\} \leq p_r\{S_{\beta''} \ln n \leq \beta_1 \ln n\}$$

Since the mean of a random variable with triangular density function is $\frac{1}{3}$,

$$E\{S_l\} = \frac{1}{3} l$$

By the Chernoff Bound [3], $\exists\ \beta'$ such that

$p_r\{S_{4l} \le l\} \le e^{-beta' \cdot l}$

$\therefore$ choosing $\beta'' > 4\beta_1$,

$$p_r\{S_{\beta'' \ln n} \le \beta_1 \ln n\} \le p_r\{S_{\beta'' \ln n} \le \frac{\beta''}{4} \ln n\} \le e^{-\frac{\beta' \beta''}{4 \ln n}}$$

Taking $\beta'' > 4\frac{\alpha}{\beta'}$,

$$e^{\frac{-\beta' \beta'' \ln n}{4}} \le e^{-\alpha \ln n} = n^{-\alpha}$$

$\therefore$ choosing $\beta'' \ge \max\left\{4\beta_1, \frac{4\alpha}{\beta'}\right\}$ we get

$$p_r\{l_3 \ge \beta'' \ln n \mid c \le \alpha_{m+1} \cdot \beta_1 \ln n\} \le n^{-\alpha}$$

$$\begin{aligned}\therefore p_r\{l_3 \ge \beta'' \ln n\} &\le p_r\{l_3 \ge \beta'' \ln n \text{ and } c \le \alpha_{m+1} \beta_1 \ln n\} \\ &\quad + p_r\{l_3 \ge \beta' \ln n \text{ and } c > \alpha_{m+1} \beta_1 \ln n\} \\ &\le p_r\{l_3 \ge \beta'' \ln n \mid c \le \alpha_{m+1} \beta_1 \ln n\} \\ &\quad + p_r\{c > \alpha_{m+1} \beta_1 \ln n\} \le n^{-\alpha} + n^{-\alpha} = 2n^{-\alpha}.\end{aligned}$$

$\therefore$ for any $\alpha > 0$, $\exists\ \beta > 0$ s.t.

$p_r\{l_3 \ge \beta \ln n\} \le n^{-\alpha}$.

This proves Theorem 5.2.

Theorem 4

For any

$$\alpha > 0 \ \exists\ \beta > s.t. \ \ Pr\ \{l_1 \ge \beta \ln n\} \le n^{-\alpha},$$

for sufficiently large n.

Proof: The density function $f_m(x)$ satisfies

$$|\ f_m(x) - f_m(y)\ | \le \frac{B}{\alpha_m^2} \cdot |\ x - y\ |,\ \forall\ x, y \in [0, \alpha_m]$$

Initially this is satisfied because $f(x)$ satisfies Lipschitz condition and $\alpha_1 = 1$. In subsequent phases, $f_m(x)$ is triangular, hence the condition is trivially satisfied. (B is independent of m.)

When

$$\begin{aligned} x, y \in I_i,\ |\ f_m(x) - f_m(y)\ | &\le \frac{B}{\alpha_m^2} \cdot \text{length}(I_i) \\ &= \frac{B}{\alpha_m^2} \cdot \frac{\alpha_m K}{n} = \frac{B \cdot k}{\alpha_m n} \end{aligned}$$

Let $P =$ probability that a given point is relabelled as "bad" during resampling

$$P = \sum_{i=1}^{N} \int_{I_i} (f(x) - f_i)\, dx$$

$$\leq \frac{B \cdot k}{\alpha_m\ n} \sum_{i=1}^{N} \int_{I_i} dx = \frac{B\ k}{n}$$

$$Pr\ \{l_1 \geq \beta \ln n\} = \sum_{s=\beta \ln n} \binom{n}{s} p^s\ (1-p)^{n-s} \leq n \cdot \binom{n}{\beta \cdot \ln n} p^{\beta \cdot \ln n}$$

$$\leq n \cdot \left[\frac{n\ e\ p}{\beta \ln n}\right]^{\beta \ln n} \leq n \cdot \left[\frac{B\ k\ e}{\beta \ln n}\right]^{\beta \ln n}$$

$$\ln Pr\ \{l_1 \geq \beta \ln n\} \leq \ln n\ + \beta \cdot \ln \left[\frac{B\ k\ e}{\beta}\right] \cdot \ln n - \beta \cdot \ln n \cdot \ln \ln n$$

$\leq -\alpha \ln n$, for $\alpha = \beta$ and sufficiently large n

$Pr\ \{l_1 \geq \beta \ln n\} \leq n^{-\alpha}$, for sufficiently large n

Proof of Theorem 3

Define $l_2 =$ number of "odd" points created during random differencing.

Lemma:

$$g_{m+1} \geq \frac{g_m + b_m - l_2}{2} - (b_m + l_1) - l_3$$

$$b_{m+1} \geq b_m + l_2 + 1$$

Proof: $\frac{g_m + b_m - l_2}{2}$ is the total number of differences created during random pairing. $(b_m + l_1)$ is the number of bad points after the resampling process. Each bad point creates at most one bad difference. Form the good differences created after random differencing. At most l_3 may be used up during compaction. hence the bound on g_{m+1} follows. At the end of compaction, there is at most one bad point created, and $(b_m + l_2)$ bad points existing after resampling may create as many bad differences. Hence the result.

Proof of the main theorem:

From Theorem 4,

$$\exists\ \beta_1 > 0\quad s.t.\ Pr\ \{l_1 \geq 2\beta_1 \ln n_m\} \leq n^{-\alpha}$$

From Theorem 5,

$$\exists\ \beta_2 > 0\quad s.t.\ Pr\ \{l_3 \geq 2\beta_2 \ln n_m\} \leq n^{-\alpha}$$

Also,

$$l_2 \leq \frac{n}{k} = \frac{g_m + b_m}{k}$$

$$\text{Since } m \leq M',\quad n_m \geq g_m \geq \sqrt{n}$$

$$\ln n_m \geq \frac{1}{2} \ln n$$

Hence with probability at least $1 - 2n^{-\frac{\alpha}{2}}$, the following bounds hold:

$l_1 \leq \beta_1 \ln n$

$$l_2 \le \frac{g_m + b_m}{k}$$

$$l_3 \le \beta_2 \ln n$$

$$g_m \ge \vartheta^{m-1} \cdot n$$

$$b_m \le m \; \beta_1 \ln n$$

Combining these with the bounds on g_{m+1} and b_{m+1} from the previous lemma,

$$b_{m+1} \le (m + 1) \; \beta_1 \ln n,$$

$$g_{m+1} \ge \frac{g_m}{2} - \frac{m}{2} \cdot \beta_1 \ln n - \frac{g_m}{2k} - \frac{m \; \beta_1 \ln n}{2k} - \beta_1 \ln n - \beta_2 \ln n$$

$$g_{m+1} \ge g_m \left[\frac{1}{2} - \frac{1}{2k} \right] - \frac{K' \ln n \cdot m}{g_m}$$

but

$$m \le k'' \ln n, \text{ and } \lim_{n \to \infty} \frac{k' \, (ln \; n)^2 \, k''}{\vartheta^{m-1} \cdot n} = 0$$

Until now, ϑ was unspecified. Choose ϑ so that

$$\vartheta < \frac{1}{2} - \frac{1}{2k}$$

Hence, for sufficiently large n ,

$$g_{m+1} \ge \vartheta \cdot g_m \ge \vartheta^m \cdot n$$

Proof of Theorem 2

Define an event

$$A_m = \{ g_m \ge \vartheta^{m-1} \cdot n \text{ and } b_m \le m \cdot \beta_1 \ln n \}$$

$$\therefore \; Pr \; \{A_{m+1} \mid A_m\} \ge 1 - n^{-\alpha}$$

$$\therefore \; Pr \; \{A_{m+1}\} \ge Pr \; \{A_{m+1} \text{ and } A_m\} = Pr \; \{A_{m+1} \mid A_m\} \; Pr \; \{A_m\}$$

$$Pr \; \{A_{m+1}\} \ge (1 - n^{-\alpha}) \cdot Pr \; \{A_m\}$$

$$\therefore \; Pr \; \{A_m\} \ge (1 - n^{-\alpha})^m \ge 1 - m \cdot n^{-\alpha}$$

$$\therefore \; Pr \; \{A_1 \text{ and } A_2 \cdots A_{M'}\} \ge 1 - \left[\sum_{m=1}^{M'} m \right] \cdot n^{-\alpha} \ge 1 - (M')^2 \, n^{-\alpha} \ge 1 - n^{-\alpha'},$$

for suitable α', since $M' = O(ln \; n)$

$$\therefore \; Pr \; \{g_m \ge \vartheta^{m-1} \cdot n \;\; \forall \; m = 1, \cdots M'\} \ge 1 - n^{-\alpha'}$$

Suppose $g_m \ge \vartheta^{m-1} \cdot n$

$$\therefore \; n_m = g_m + b_m \ge \vartheta^{m-1} \cdot n$$

$$\therefore \; n_{M'} \ge \vartheta^{m-1} \cdot n \ge n \cdot e^{-\ln \frac{1}{\vartheta}(M'-1)} \ge n \cdot e^{-\ln \frac{1}{\vartheta} \cdot \frac{1}{2} \frac{\ln n}{\ln \frac{1}{\vartheta}}} = \sqrt{n}$$

$$\therefore \; n_{M'} \ge 1$$

$\therefore$ There are at least M' phases.

$$\alpha_{m+1} = \frac{\alpha_m}{N_m} = \frac{\alpha_m k'}{n_m} \le \frac{k' \alpha_m}{n \vartheta^{m-1}}$$

$$\therefore \quad \alpha_{m+1} \le \left(\frac{k'}{n}\right)^m \vartheta^{\frac{-m(m-1)}{2}}$$

$$\ln \alpha_{m+1} \le -[m(\ln n - \ln k') - \frac{m(m-1)}{2} \ln \frac{1}{\vartheta}]$$

$$= -\frac{m}{2} \ln n - \frac{m}{2}[\ln n - 2 \ln k' - \frac{m-1}{2} \ln \frac{1}{\vartheta}]$$

$$\le -\frac{m}{2} \ln n - \frac{m}{2}[\ln n - 2 \ln k' - \frac{1}{4} \ln n]$$

$$\le -\frac{m}{2} \ln n \quad \text{for sufficiently large } n$$

$$\therefore \alpha_{m+1} \le e^{-\frac{m}{2} \ln n}$$

Proof of Theorem 1

$$D(A, B) \le \alpha_{M'} \le e^{-\frac{M' \ln n}{2}} \le e^{-\frac{1}{4}\frac{(\ln n)^2}{\ln \frac{1}{\vartheta}}}$$

with $Prob \to 1$ as $n \to \infty$.

Taking $\alpha = \frac{1}{4} \frac{1}{\ln \frac{1}{\vartheta}}$, proves the main claim of this paper, for the case of two processors.

3. Extension to k-Partitioning

3.1. Generalized differencing operations in k dimensions

1. Difference Vector

Let S be the set of numbers to be partitioned into k subsets.

Let $A_1, A_2, \cdots A_k$ be any collection of disjoint subsets of S. (Their union is not necessarily equal to S.) We will refer to such a collection as a "partial" solution to the k-partitioning problem, and it has the following interpretation: We have restricted the solution of the problem so that any two numbers belonging to the same set A_i must appear on the same side of the partition and any two numbers belonging to different A_i's must appear on different sides of the partition. Stated another way, if $A'_1, A'_2, \cdots A'_k$ is the final solution to the problem then there exist s a permutation $\sigma = \{i_1, i_2, \cdots i_k\}$ of the set of indices $\{1, 2, \cdots k\}$ such that $A_i \subseteq A'_{\sigma(i)} \ \forall i$.

let $s_i = \sum_{x \in A_i} x$

The vector $(s_1, s_2, \cdots s_k)$ is defined as the difference vector corresponding to this partial solution and is denoted by $\underline{D}(A_1, A_2, \cdots A_k)$. We consider two difference vectors $\underline{S}, \underline{S}'$ as equivalent if

1. One can be obtained from another by permutation of components.
2. Their corresponding components differ by equal amounts. i.e. $s_i - s'_i$ is the same for all $i = 1, 2, \cdots k$.

Example 1 let $S = \{4, 9, 11, 23, 28, 32, 37\}$

$$\text{let } A_1 = \{4, 23\} \quad A_2 = \{11, 28\} \quad A_3 = \{37\}$$

$$\underline{S} = \{27, 39, 37\}$$

$$\text{let } \underline{S}' = \{20, 32, 30\}$$

then $\underline{S}$, S' are equivalent because

$$(\underline{S} - \underline{S}')_i = 7 \text{ for } i = 1, 2, 7$$

$$\text{let } \underline{S}'' = \{12, 10, 0\}$$

then $\underline{S}''$ is equivalent to $\underline{S}$, since it can be obtained from S by subtracting 27 from all components and then permuting the components.

Reduced form of a difference vector:

Let $\underline{S} = \{s_1, s_2, \ldots s_k\}$ be a difference vector

Let $s_m = \min_i s_i$

Define

$$S' = \{s'_1, s'_2, \cdots s'_k\}$$

as

$$s'_i = s_i - s_m$$

Then all components of $\underline{S}'$ are non-negative, at least one component is zero and $\underline{S}'$ is equivalent to $\underline{S}$.

We define $\underline{S}'$ to be the reduced form of $\underline{S}$ and denote it by Reduce ($\underline{S}$).

Example 2: let $\underline{S} = \{10, 15, 23\}$ then Reduce ($\underline{S}$) = $\{0, 5, 13\}$.

2. Differencing Operations

Now we introduce a method of combining two or more partial solutions to the problem to create a single and more restrictive partial solution.

Let $A_1, A_2, \ldots, A_k$ and $A'_1, A'_2, \ldots A'_k$ (all disjoint) be two partial solutions and let $\underline{S}$, $\underline{S}'$ be the corresponding difference vectors.

Let $\sigma = \{i_1, i_2, \ldots, i_k\}$ be any permutation of $\{1, 2, \ldots, k\}$.

Define

$$\sigma o \underline{S}' = \{s'_{i_1}, s'_{i_2}, \cdots s'_{i_k}\} \ .$$

The combined partial solution is defined as:

$$A_1 \cup A'_{i_1} \ , A_2 \cup A'_{i_2} \ , \ldots\ldots A_k \cup A'_{i_k} \ .$$

The corresponding difference vector is:

$$\underline{S} + \sigma o \underline{S}' = \{s_1 + s'_{i_1}, s_2 + s'_{i_2}, \ldots\ldots s_k + s'_{i_k}\}$$

A similar operation can be defined for combining more than two difference vectors. Also, the permutation σ can be chosen in many different ways.

Example 3: let S = {2, 7, 11, 14, 19, 22, 28, 35, 37}
let $A_1 = \{2, 19\}$, $A_2 = \{11\}$, $A_3 = \{37\}$ be a partial solution
then $\underline{S} = \{21, 11, 37\}$.

let $A_1' = \{35\}$, $A_2' = \varphi$, $A_3' = \{7, 22\}$ be another disjoint partial solution
then $\underline{s}' = \{35, 0, 29\}$.

let $\sigma = \{3, 1, 2\}$
$\therefore \sigma o \underline{S}' = \{29, 35, 0\}$.

The combined partial solution B_1 , B_2 , B_3 is given by $B_1 = \{2, 19, 7, 22\}$, $B_2 = \{11, 35\}$, $B_3 = \{37\}$, and the corresponding difference vector is: $\underline{S} + \sigma o \underline{S}' = \{50, 46, 37\} \equiv \{13, 9, 0\}$.

3. Binary Differencing

This is a particular differencing operation for combining $\underline{S}$ and $\underline{S}'$ in which the permutation σ is chosen so that the largest component of $\underline{S}$ is combined with the smallest component of $\underline{S}'$, the second largest component of $\underline{S}$, with the second smallest component of $\underline{S}'$ and so on.

Arrange the components of $\underline{S}$ in non-decreasing order and components of $\underline{S}'$ in non-increasing order:

$$s_1 \leq s_2 \leq \cdots \leq s_{k-1} \leq s_k$$

$$s'_1 \geq s'_2 \geq \cdots s'_{k-1} \geq s'_k$$

Take the sum

$$\underline{S} + \underline{S}' = \{s_1 + s'_1, s_2 + s'_2, \cdots s_k + s'_k\}$$

Then the reduced form of $\underline{S} + \underline{S}'$ is defined as the binary difference of $\underline{S}$ and $\underline{S}'$.

Example 4:

Let

$$A_1 = \{2, 19\} \quad A_2 = \{11\} \quad A_3 = \{4, 27\}$$

$$\underline{D}\,(A_1, A_2, A_3) = \{21, 11, 31\} \equiv \{10, 0, 20\}$$

Let

$$A'_1 = \{5, 17\} \quad A'_2 = \{7, 14\} \quad A'_3 = \{25\}$$

$\therefore$

$$\underline{D}'\,(A'_1, A'_2, A'_3) = \{22, 21, 25\} \equiv \{1, 0, 4\}$$

The binary difference of $\underline{D}$ and $\underline{D}'$ is

$$\{20, 10, 0\} + \{0, 1, 4\} = \{20, 11, 4\} \equiv \{16, 7, 0\},$$

and the corresponding partition is

$$B'_1 = \{4, 27, 7, 14\} \quad B'_2 = \{2, 19, 5, 17\} \quad B'_3 = \{11, 25\}$$

Now we prove some properties of binary differencing.

Lemma: Suppose $\underline{S}, \underline{S}' \in [0, m]^k$ then $\underline{x} \in [0, m]^k$

Proof: let j be such that $s_j + s'_j = \min_i (s_i + s'_i)$

then

$$x_i = s_i + s'_i - (s_j + s'_j) = (s_i - s_j) + (s'_i - s'_j)$$

one of these terms is non-negative and the other is non-positive and the absolute value of each is less thanb or equal to m. Hence the result follows.

An application of this lemma:

$$\text{let } \{\underline{S}_1, \underline{S}_2, \cdots \underline{S}_n\} \subseteq [0, m]^k$$

be a collection of difference vectors. Take any two vectors $\underline{S}_i$ and $\underline{S}_j$ and replace them by their binary difference. By repeating this operation, the set $\{\underline{S}_1, \underline{S}_2, \cdots \underline{S}_n\}$ can be reduced to a single difference vector $\underline{S}$. By the lemma above, $\underline{S} \in [0, m]^k$. This idea will be used in the compaction operation.

4. Cyclic Differencing:

let $\underline{S}_1, \underline{S}_2, \cdots \underline{S}_k$ be a collection of k difference-vectors, each with k components.

let σ be the permutation $(k, 1, 2, \cdots k-1)$.

As before,

$$\sigma o \{x_1, x_2, \cdots x_k\} = \{x_k, x_1, x_2, \cdots x_{k-1}\}$$

$$\text{let } S = S_1 + \sigma^2 o\ S_2 + \sigma o\ S_3 \cdots \sigma^{k-1} o\ S_k$$

The reduced form of S is called the cyclic difference of $\underline{S}_1, \underline{S}_2, \cdots \underline{S}_k$.

Lemma:

$$\text{let } \{\underline{S}_1, \underline{S}_2, \cdots \underline{S}_k\} \subseteq \underline{a} + [0, b]^k$$

and let $\underline{S}$ be their cyclic difference. Then

$$\underline{S} \in [0, kb]^k$$

(Notation: $\underline{a} = \{a_1, a_2, \cdots a_k\}$ is any k vector, and

$$(\underline{a} + [0, b]^k = \{\underline{x} \in R^k \mid a_i \le x_i \le a_i + b\})$$

Proof:

$$\underline{S}_i \in \underline{a} + [0, b]^k \rightarrow \underline{S}_i - \underline{a} \in [0, b]^k$$

$$\rightarrow \sigma^{i-1} o\ \underline{S}_i - \sigma^{i-1} o\ \underline{a} \in [0, b]^k$$

$$\therefore \sum_{i=1}^{n} \sigma^{i-1} o\ \underline{S}_i - \sum_{i=1}^{n} \sigma^{i-1} o\ \underline{a} \in [0, kb]^k$$

All components of

$$\sum_{i=1}^{n} \sigma^{i-1} o\ \underline{a}$$

are equal to

$$\sum_{i=1}^{n} a_i \, .$$

Hence

$$\underline{S} = \text{Reduce}\left[\sum_{i=1}^{n} \sigma^{i-1} \circ \underline{S}_i\right] = \text{Reduce}\left[\sum_{i=1}^{n} \sigma^{i-1} \circ \underline{S}_i - \sum_{i=1}^{n} \sigma^{i-1} \circ \underline{a}\right]$$

$$\therefore \quad \underline{S} \in [0, kb]^k$$

Example 5:

let

$S_1 = \{302, 105, 404\}$

$S_2 = \{307, 102, 409\}$

$S_3 = \{306, 108, 401\}$

If we take $\underline{a} = \{300, 100, 400\}$ and $b = 10$, then

$$S_i \in \underline{a} + [0, b]^3 \quad \text{for } i = 1, 2, 3$$

Now we compute the cyclic difference of S_1, S_2 and S_3

$\sigma = \{3, 1, 2\}$

$\sigma^0 \circ S_1 = \{302, 105, 404\}$

$\sigma^1 \circ S_2 = \{409, 307, 102\}$

$\sigma^2 \circ S_3 = \{108, 401, 306\}$

$$\sigma^0 \circ S_1 + \sigma^1 \circ S_2 + \sigma^2 \circ S_3 = \{819, 813, 812\} \equiv \{7, 1, 0\} \in [0, 30]^3$$

3.2. Description of the algorithm

Now we describe a partitioning algorithm using the cyclic differencing operation. Each number in the input set is converted into a k-tuple by setting the first $k-1$ components to zero and the last component equal to the number. The algorithm is divided into a sequence of phases; the input set S to the m^{th} phase is a set of k-tuples contained in $[0, \alpha_m]^k$, (and the first component of each k-tuple is always zero). The cell $[0, \alpha_m]^k$ is divided into $O(n)$ subcells of equal size (where $n = |S|$). Then we pick a set of k points at random from a subcell and take their cyclic difference. We repeat this operation on each subcell as long as possible, i.e., until each subcell contains fewer than k points. The magnitude of differences created in this process is smaller by a factor of about $O(n^{\frac{1}{k-1}})$. If the initial distribution $\underline{f}(\underline{x})$ of k-tuples over $[0, \alpha_m]^k$ is reasonably smooth (i.e., satisfies a Lipschitz condition), then the distribution of k-tuples inside each subcell is approximately uniform. (It is made exactly uniform by resampling.) Cyclic differences of k-tuples drawn randomly from the uniform distribution are distributed independently according to a distribution $f_\Delta(\underline{x})$, which again is smooth. Thus the essential properties of the distribution are reproduced form phase to phase, just as in the 2-processor case.

This process leaves some bad points behind: those created during resampling and at most $k-1$ per subcell during random differencing. We compact them as follows: First, by repeated application of binary differencing, the set of bad points is reduced to a single bad point. This bad point is brought in the range $[0, \alpha_{m+1}]^k$ by differencing it against good points from $[0, \alpha_{m+1}]^k$, otherwise meant to be used as input to the next phase. The number of good points left as input to the next phase is more than some constant fraction of the number of points input to the current phase, with very high probability. hence the total number of phases is about $O(\log n)$ and each phase reduces the order of magnitude of points by about $O(n^{\frac{1}{k-1}})$ leading to a final difference of $O(n^{-c \log n})$.

Now we give a description of the algorithm and the associated notation.

Input to the algorithm is a set $S \subseteq [0, 1]$ of n real numbers distributed independently and according to a common density function $f(x), x \in [0, 1]$. The function $f(x)$ is arbitrary, subject to a Lipschitz condition:

$$\exists B > 0 \;\; s.t. \;\; \forall x, y \in [0, 1], \; | f(x) - f(y) | \leq B \, | x - y |$$

The first step of the algorithm is a preprocessing operation that converts each number in the input into a k-tuple by prefixing it with $k-1$ zeros.

$S_1 = \varphi$

for all $x \in S$ do begin

$x_i = 0 \quad i = 1, 2, \cdots k-1$

$x_k = x$

$\underline{x} = \{x_1, x_2, \cdots x_k\}$

$S_1 = S_1 \cup \{\underline{x}\}$

end

The algorithm is divided into a sequence of phases. The input to the m^{th} phase is a set $S_m, \{\underline{b}\} \subseteq [0, \alpha_m]^k$ of k-tuples. The first component of each k-tuple is always zero, $\underline{b}$ is a single bad point. There are three parameters n_m, α_m, N_m associated with a phase and they satisfy:

$n_m = | S_m |$ = Number of k-tuples in the input

$N_m = \frac{n_m}{k'}$ = Number of subcells forming partition of $[0, \alpha_m]^k$

$$\alpha_{m+1} = \frac{(k-1)\,\alpha_m}{N_m^{\frac{1}{k-1}}}$$

(k' is a constant parameter of the algorithm, which satisfies certain conditions described in the proof of Theorem 5)

Initally, $\alpha_1 = 1, n_1 = n$.

Each phase consists of the following sequence of operations:

1. Partitioning
2. Resampling
3. Random Cyclic differencing
4. Compaction

We describe each of these operations in turn.

1. **Partitioning:**

Let $N_m = \frac{n_m}{k}$, $\alpha_{m+1} = \frac{\alpha_m}{N_m^{\frac{1}{k-1}}}$

Divide the cell $\{0\} \times [0, \alpha_m]^{k-1}$ into N_m equal size cubic cells, each with side α_{m+1}.

Construct the corresponding partition of input set S.

$$S_i = S \cap C_i \quad i = 1, 2, \cdots N_m$$

2. **Resampling:**

Let $f(\underline{x})$, $\underline{x} \in \{0\} \times [0, \alpha_m]^{k-1}$ be the density function of points in S.

Define $f_i = \min_{\underline{x} \in C_i} f(\underline{x})$

Define a function $g(\underline{x})$, $\underline{x} \in \{0\} \times [0, \alpha_m]^{k-1}$ as follows:

$$g(\underline{x}) = f_i \text{ } when \text{ } \underline{x} \in C_i$$

Thus $g(\underline{x})$ approximates $f(\underline{x})$ from below and remains uniform within each cell C_i.

A point $\underline{x} \in S$ is labelled as

"good" with probability $\frac{g(\underline{x})}{f(\underline{x})}$ (when $f(\underline{x}) > 0$)

and "bad" with probability $1 - \frac{g(\underline{x})}{f(\underline{x})}$, independently of other random variables.

let $G = \{\underline{x} \in S \mid \underline{x} \text{ } is \text{ } labelled \text{ } good\}$

$B = \{\underline{b}\} \cup \{\underline{x} \in S \mid \underline{x} \text{ is labelled } bad\}$

$G_i = G \cap C_i$

3. **Random Cyclic Differencing:**

From each subcell C_i we pick sets of k points randomly and find their cyclic difference. The leftover points in each cell are merged with the bad points.

```
S' = φ
For i = 1, 2, ··N_m do begin
    while |g_i| ≥ k do begin
        randomly pick a set x_1, x_2, ··x_k ∈ G_i
        let x be their cyclic difference
        S' = S' ∪ {x}
        G_i = G_i − {x_1, x_2, ··x_k}
    end
    B ← B ∪ G_i
end
```

4. Compaction

First, we reduce the set of bad points to a single bad point by repeated binary differencing.

While $|B| \geq 2$ do begin
  pick $\underline{x}, \underline{y} \in B$ arbitrarily.
  let $\underline{z}$ be their binary difference.
  $B \leftarrow B - \{\underline{x}, \underline{y}\} + \{\underline{z}\}$
end

Suppose $B = \{\underline{b}\}$. We bring $\underline{b}$ into the range $[0, \alpha_{m+1}]^k$ by differencing it against good points from S'.

while $\underline{b} \not\in [0, \alpha_{m+1}]^k$ do begin
  pick $\underline{x} \in S'$ at random
  $\underline{b} \leftarrow$ binary difference of $\underline{b}$ and $\underline{x}$
end

$S' \cup \{\underline{b}\}$ constitutes input to the next phase.

3.3. Inductive assertions about the algorithm

In this section, we make a few more definitions and state the inductive assertions about the algorithm. Proofs will be given in the next section.

We define a function $f_\Delta(\underline{x})$, $\underline{x} \in [0, 1]^{k-1}$ as follows:

1. Let $u(x)$ be the uniform distribution over $[0, 1]$.
2. Let $u_k(x)$ be the sum of k independent random variables with common density function $u(x)$.
3. Let $x_1, x_2, \cdots x_k$ be a collection of k independent random variables with density $u_{k-1}(x)$.
4. Let $x_m = \min\limits_i x_i$

$x'_i = x_i - x_m \quad i \neq m$

Let $(x'_1, x'_2, \cdots x'_{k-1})$ be the collection (suitably reindexed) of x'_i's.
Let $g(x'_1, x'_2, \cdots x'_{k-1})$ be their density function.

5. Let $f_\Delta(\underline{x}) = (k-1) \cdot g((k-1)\underline{x})$, $\underline{x} \in [0, 1]^{k-1}$.

Theorem 1

The input S to the m^{th} phase consists of independent random variables with density function $\dfrac{1}{\alpha_m} f_\Delta(\dfrac{x}{\alpha_m})$.

Theorem 2

$f_\Delta(\underline{x})$ satisfies a Lipschitz condition:

$\exists B > 0 \quad s.t. \quad \forall \underline{x}, \underline{y} \in [0, 1]^{k-1}, \ |f_\Delta(\underline{x}) - f_\Delta(\underline{y})| \leq B \, |\underline{x} - \underline{y}|$

Let $l_1 =$ number of points labelled as "bad" during resampling. Recall that N is the number of subcells in a phase.

The following theorem says that l_1 is small with high probability.

Theorem 3

$$\forall \beta > 0 \ \exists N_0 \ s.t. \ \forall N \geq N_0, \ Pr\{l_1 \geq N\} \leq e^{-\beta N}$$

Let l_3 = number of good points used from the set S' to compensate for a single bad point during second step of compaction.

The following theorem says that l_3 is small with high probability.

Theorem 4

$$\exists \text{ a constant } \delta > 0 \ s.t. \ Pr\{l_3 > \delta \cdot N\} \leq e^{-\beta N}$$

The following theorem says that the number of points is reduced by a constant factor (approximately) during each phase.

Theorem 5

$$\exists \text{ a constant } \vartheta > 0 \ s.t.$$

$$\text{if } m \leq M' = \left\lceil \frac{\frac{1}{4}\ln n}{\ln \frac{1}{\vartheta}} \right\rceil$$

Then $Pr\{n_{m+1} \geq \vartheta^m \cdot n \mid n_m \geq \vartheta^{m-1} \cdot n\} \geq 1 - e^{-\beta\sqrt{n}}$

The following theorem establishes the main claim of the paper.

Theorem 6

With probability $\to 1$ as $n \to \infty$.

1. There are at least M' phases.
2. $\exists$ constant α s.t.
   $D(A_1, A_2, \cdots A_k) \leq e^{-\alpha(\ln n)^2}$

3.4. Proofs of the inductive assertions

Proof of Theorem 1: After resampling, the good points are uniformly distributed in each subcell. From the definition of $f_\Delta(x)$ it trivially follows that the cyclic difference of a set of k number randomly chosen from a subcell has the density function $f_\Delta(x)$.

Proof of Theorem 2:

1. for $k \geq 2$ $u_k(x)$ satisfies Lipschitz's condition
   for $k = 2$ $u_2(x) = 1 - |x - 1|$ $x \in [0, 2]$

$$\therefore |u_2(x) - u_2(y)| \leq |x - y|$$

for $k > 2$,

$$u_k(x) = \int_{t=-\infty}^{\infty} u_{k-1}(x - t)\, u(t)\, dt$$

$$|u_k(x) - u_k(y)| \leq \int_{-\infty}^{\infty} |u_{k-1}(x - t) - u_{k-1}(y - t)|\, u(t)\, dt$$

$$\leq \int_{-\infty}^{\infty} |x - y| \, u(t)\, dt = |x - y|.$$

Hence the result follows by induction.

2. $u_k(x)$ is bounded
for $k = 2 \quad |u_2(x)| \leq 1$
for $k > 2 \quad |u_k(x)| \leq \int_{-\infty}^{\infty} |u_{k-1}(x - t)| \, u(t)\, dt \leq \int_{-\infty}^{\infty} u(t) dt = 1$

3. $g(x'_1, x'_2, \ldots x'_{k-1}) = \frac{\partial}{\partial x'_i} \frac{\partial}{\partial x'_2} \cdots \frac{\partial}{\partial x'_{k-1}} G(x'_1, x'_2, \ldots x'_{k-1})$, where

$$\begin{aligned} G(x'_i, x'_2, \ldots x'_{k-1}) &= p_r\{x_m \leq x_i \leq x_m + x'_i\} \\ &= k \cdot p_r\{x_1 \leq x_i \leq x_1 + x'_i \quad i \neq 1\} \\ &= k \int_{-\infty}^{\infty} u_k(x)\, p_r\{x \leq x_i \leq x + x'_i \quad i \neq 1\} dx \\ &= k \int_{-\infty}^{\infty} u_k(x) \prod_{i=1}^{k-1} \left[U_k(x + x_i) - U_k(x)\right] dx \end{aligned}$$

$$\therefore g(x'_1, x'_2, \ldots x'_{k-1}) = k \int_{-\infty}^{\infty} u_k(x) \prod_{i=1}^{k-1} [u_k(x + x'_i)]\, dx$$

$$|g(\underline{x}') - g(\underline{y}')| \leq k \int_{-\infty}^{\infty} u_k(x) \left| \prod_{i=1}^{k-1} u_k(x + x'_i) - \prod_{i=1}^{k-1} u_k(x + y'_i) \right| dx$$

$$\left|\prod_{i=1}^{k-1} u_k(x + x'_i) - \prod_{i=1}^{k-1} u_k(x + y'_i)\right| \leq \sum_{i=1}^{k-1} |x'_i - y'_i| \quad (\textit{using } 1 \text{ and } 2)$$

$$\leq (k - 1)\, |\underline{x}' - \underline{y}'|$$

$$\therefore |g(\underline{x}') - g(\underline{y}')| \leq k(k - 1)\, |\underline{x}' - \underline{y}'|$$

4.

$$\begin{aligned} |f_\Delta(\underline{x}) - f_\Delta(\underline{y})| &= |(k-1)\, g((k-1)\underline{x}) - (k-1)\, g((k-1)\underline{y})| \\ &\leq k(k-1)^3 |\underline{x} - \underline{y}| = B |\underline{x} - \underline{y}| \end{aligned}$$

Proof of Theorem 3: The probability that a number is lost due to resampling (independently of other numbers) is given by

$$p = \sum_{i=1}^{N} \int_{C_i} (f(\underline{x}) - f_i)\, dA$$

1. Observe that p does not change by scaling the distribution so that it is defined over $[0, 1]^{k-1}$ rather than $[0, \alpha]^{k-1}$.

Proof: Let $\underline{y} = \frac{1}{\alpha}\underline{x}$ so that $\underline{x} \in [0, \alpha]^{k-1} \Leftrightarrow \underline{y} \in [0, 1]^{k-1}$. Let $g(\underline{y}) = \alpha^{k-1} f(\alpha \underline{y})$.

If the resampling process is applied to $g(\underline{y})$ then

$$p' = \sum_{\frac{1}{\alpha}C_i} [g(\underline{y}) - g_i]\, dy_1, dy_2, \ldots dy_{k-1}$$

$$= \sum_{i=1}^{N} \int_{C_i} \alpha^{k-1} [f(\underline{x}) - f_i] \frac{dx_1}{\alpha} \frac{dx_2}{\alpha} \cdots \frac{dx_{k-1}}{\alpha}$$

$$= \sum_{i=1}^{N} \int_{C_i} [f(\underline{x}) - f_i]\, dx_1, dx_2 \ldots dx_{k-1}$$

$$\therefore p' = p$$

2. From Theorem 1, it follows that, after scaling, the density function we are dealing with is $f_\Delta(\underline{x})$ which satisfies the Lipschitz condition by Theorem 2.

If $\underline{x}, \underline{y}, \in C_i$, then $|x - y| \le \frac{\sqrt{k-1}}{N^{1/k-1}}$.

$$\therefore |f(\underline{x}) - f_i| \le \frac{B\sqrt{k-1}}{N^{1/k-1}}$$

$$\therefore p = \sum_{i=1}^{N} \int_{C_i} (f(\underline{x}) - f_i)\, dA \le \frac{B\sqrt{k-1}}{N^{1/k-1}} \sum_{i=1}^{N} \int_{C_i} dA = \frac{B\sqrt{k-1}}{N^{1/k-1}}$$

$$\therefore p \le \frac{k''}{N^{1/k}}$$

3.

$$p_r\{l_1 \ge N\} = \sum_{S=N}^{n} \binom{n}{s} p^s (1-p)^{n-s} \le \sum_{S=N}^{n} \left(\frac{ne\,p}{s}\right)^s \le n \cdot \left(\frac{ne\,k''}{N\,N^{1/k}}\right)^N$$

$$\ln [p_r\{l_1 \ge N\}] \le \ln n + \frac{n}{k} \ln \left(\frac{e\,k'' k'^{1/k}}{k'}\right) - \frac{n}{k'k} \ln n \le -\beta' n = -\beta N$$

for any β and sufficiently large N.

$$\therefore p_r\{l_1 \ge N\} \le e^{-\beta . N}$$

Proof of Theorem 4: Let $\underline{b}$ be the single bad point left after first step of compaction. We are interested in finding a bound on l_3, the number of good points from S' required to bring b in the range $[0, \alpha_{m+1}]^k$.

Arrange the components of $\underline{b}$ in non-decreasing order :

$0 = b_1 \le b_2 \le b_3 \ldots b_{k-1} \le b_k$

We say that component b_i is "small" if $b_i \le \alpha_{m+1}$ and "big" otherwise. This partitions the indices $\{1, 2, \ldots k\}$ into small $(\underline{b})$ and Big $(\underline{b})$:

Small $(\underline{b}) = \{i \mid b_i \text{ is small}\}$
Big $(\underline{b}) = \{i \mid b_i \text{ is big}\}$

Recall that $\underline{b}$ is differenced with some $\underline{x} \in [0, \alpha_{m+1}]^k$. Under these conditions the following property holds:

Lemma 1. After single binary differencing step, a small component remains small, and a big component does not increase in magnitude. (Proof deferred until the end of this section).

This suggests that we divide the sequence of differencing steps into phases: During a phase, the set of small components remains the same and a phase change occurs when a new component becomes small for the first time. Hence there are at most $k-1$ phases. If we can establish a bound L on the number of good points lost during a phase, then we have

$$l_3 \le (k-1)\, L$$

To analyze a single phase, we make a few definitions:

Let

$$|\, Big\, (\underline{b})\, | = \sum_{i \in Big\, (\underline{b})} b_i$$

Suppose $\underline{b}'$ is the binary difference of $\underline{b}$ and $\underline{x} \in \{0\} \times [0, \alpha_{m+1}]^{k-1}$

Define $x_m = \min_i \{x_i \mid i = 2, \ldots k\}$

Lemma 2. $|Big\, (\underline{b}')| \le |Big\, (\underline{b})| - x_m$ (Proof deferred) i.e. $|Big\, (\underline{b})|$ is reduced by an amount at least equal to x_m. Note that x_m is a random variable with density function $\frac{1}{\alpha_{m+1}} g_m \left(\frac{x}{\alpha_{m+1}} \right)$ where $g_m(y)$ is density function of the random variable $y = \min_i \{y_1, y_2, \ldots y_{k-1}\}$ and $y_1, \ldots y_{k-1}$ are distributed according to $f_\Delta(\underline{y})$. The density function $g_m(x)$ has some finite non-zero mean μ. If S_n is the sum of n independent random variables with density $g_m(x)$ then we have by Chernoff bound [3] $p_r\{S_n < n \,.\, \frac{\mu}{2}\} \le e^{-\beta n}$ for some β.

If l is the actual number of good points lost in a phase, then

$$\alpha_{m+1} \cdot S_l \le |\, big\, (\underline{b})\, | \le \alpha_m \cdot (k-1)$$
$$\therefore S_l \le N \,.\, (k-1)$$
$$\therefore p_r\{l \ge L\} \le p_r\{S_L \le N(k-1)\} \le e^{-\beta L},$$

by setting

$$L \,.\, \frac{\mu}{2} = N\,(k-1)$$

$\therefore p_r\{l_3 \geq (k-1)\, L\} \leq (k-1)\, e^{-\beta L}$

Choose $\delta = (k-1)^2 \cdot \frac{2}{\mu}$ so that

$\delta N = (k-1)\, L$

$\therefore p_r\{l_3 \geq \delta N\} \leq (k-1)\, e^{\frac{-\beta . 2N}{\mu (k-1)}} \leq e^{-\beta' N}$

for suitable choice of β' and sufficiently large N. This completes the proof of main theorem. Now we prove the lemmas.

Proof of Lemma 1: Arrange components of $\underline{b}$ and $\underline{x}$ as follows:

$0 = b_1 \leq b_2 \ldots\ldots\ldots\; b_{k-1} \leq b_k \ldots\ldots\ldots$

$x_1 \geq x_2 \ldots\ldots\ldots\; x_{k-1} \geq x_k = 0$

Let $b_m + x_m = \min\limits_i \,(b_i + x_i)$.

Let $\underline{b}'$ be the binary difference of $\underline{b}$ and $\underline{x}$
then $b'_i = (b_i + x_i) - (b_m + x_m)$.

$b_m + x_m \leq b_1 + x_1 = x_1 \leq \alpha_{m+1}$

$\therefore b_m \leq \alpha_{m+1}$

$\therefore m \in small\ (\underline{b})$

This has two consequences

$i \in small\ (\underline{b}) \Rightarrow |b_i - b_m| \leq \alpha_{m+1}$
$i \in Big\ (\underline{b}) \Rightarrow x_i - x_m \leq 0$

To prove that a small component remains small, let $i \in small\ (\underline{b})$ then $b'_i = (b_i + x_i) - (b_m + x_m) = (b_i - b_m) + (x_i - x_m)$
$|b_i - b_m| \leq \alpha_{m+1}$, $|x_i - x_m| \leq \alpha_{m+1}$, one of these terms is non-negative and the other is non-positive. Hence $b'_i \leq \alpha_{m+1}$.

To prove that a big component does not increase, let $i \in Big\ (\underline{b})$ then $b'_i = (b_i + x_i) - (b_m + x_m) \leq b_i + (x_i - x_m) \leq b_i$

Proof of Lemma 2.

$b'_k = (b_k + x_k) - (b_m + x_m) = b_k - (b_m + x_m)$
$b_k - b'_k = (b_m + x_m) \geq \min\limits_i \{x_i\}$
$b_i - b'_i \geq 0 \;\; i \neq k,\; i \in Big\ (\underline{b})$

$|Big\ (\underline{b})| - |Big\ (\underline{b}')| = \sum\limits_{i \in Big\ (\underline{b})} (b_i - b'_i) \geq \min\limits_i \{x_i\}$

$|Big\ (\underline{b})| \leq |Big\ (\underline{b})| - x_m \quad where \quad x_m = \min\limits_i \{x_i\}$

Proof of Theorem 5: Let S and S' be the input and output of a phase.

Let $|S| = n$ and $|S'| = n'$.

Recall that l_1 = Number of good points from S lost due to resampling and l_3 = Number of good points from S' lost in compaction.

Define l_2 = Number of left-over points after random cyclic differencing.

Then $n' = \frac{n - l_1 - l_2}{k} - l_3$

l_1, l_2 and l_3 satisfy the following:

1. $\exists\, \delta > 0$ s.t. $p_r\{l_3 > \delta N\} \leq e^{-\beta N}$
2. $l_2 \leq (k-1) N$, since at most $k-1$ points are left in each subcell.
3. $\forall\, \beta > 0\ \exists\, N_0$ s.t. $N \geq N_0 \Rightarrow p_r\{l_1 \geq N\} \leq e^{-\beta N}$. In particular, choose the same β as in 1.

Therefore with probability at least $1 - 2e^{-\beta N}$ we have $l_1 \leq N$, $l_2 \leq (k-1)N$, $l_3 < \delta N$.

$$n' \geq \frac{n - N - (k-1)N}{k} - \delta N = \frac{n}{k} - (\delta + 1) N = n \left[\frac{1}{k} - \frac{\delta + 1}{k'}\right]$$

Choose ϑ such that $0 < \vartheta < \frac{1}{k} - \frac{\delta + 1}{k'}$ (Note that k' must be so chosen that the above inequality has a solution).

$\therefore n' \geq n \,.\, \vartheta$

Now consider the input n to m^{th} phase as a random variable n_m.

Suppose

$n_m \geq \vartheta^{m-1} \,.\, n$ and $m \leq m' = \left\lfloor \frac{1}{4} \ln \frac{n}{\ln 1/\vartheta} \right\rfloor$

$\therefore n_m \geq \sqrt{n}$

$1 - 2e^{-\beta n_m} \geq 1 - 2e^{-\beta\sqrt{n}} \geq 1 - e^{-\beta'\sqrt{n}}$

for suitable β'.

but $n_{m+1} \geq \vartheta \,.\, n_m$

$\therefore n_{m+1} \geq \vartheta^m \,.\, n$

$\therefore p_r\{n_{m+1} \geq \vartheta^m \,.\, n \mid n_m \geq \vartheta^{m-1} \,.\, n\} \geq 1 - e^{-\beta\sqrt{n}}$

Proof of Theorem 6:

$$\alpha_{m+1} = \frac{\alpha_m\,(k-1)}{N_m^{\,1/k-1}}$$

$$\therefore \frac{(\alpha_{m+1})^{k-1}}{(\alpha_m)^{k-1}} = \frac{(k-1)^{k-1}}{N_m} = \frac{k'(k-1)^{k-1}}{n_m} = \frac{k''}{n_m}\,.$$

In the proof of theorem 2 for 2-processor case, we have

$$\frac{\alpha_{m+1}}{\alpha_m} = \frac{k'}{n_m}$$

With the exception of this change, the proof of this theorem is identical to that of theorem 2 for 2-processor case.

Hence we get

$$(\alpha_{M'})^{k-1} \le e^{-\frac{1}{4}\frac{(\ln n)^2}{\ln 1/\vartheta}}$$

$$\therefore \alpha_{M'} \le e^{-\frac{1}{4}\frac{(\ln n)^2}{(k-1)\ln 1/\vartheta}}$$

and $D(A_1, A_2, \ldots A_k) \le \alpha'_M$

taking

$$\alpha = \frac{1}{4}\frac{1}{(k-1)\ln 1/\vartheta},$$

we get

$$D(A_1, A_2, \ldots A_k) \le e^{-\alpha(\ln n)^2}, \ \textit{with Prob} \to 1 \textit{ as } n \to \infty$$

References

1. R.L. Graham, Bounds on multiprocessing timing anomalies, SIAM J. Appl. Math, 17 (1969), pp. 416-429.
2. E.G. Coffman, Jr., M.R. Garey and D.S. Johnson, An application of bin-packing to multiprocessor scheduling, SIAM J. Comput., Vol. 7, No. 1, Feb 1978.
3. Wozencraft and Jacobs, Principles of communication engineering, John Wiley and Sons (1965).