ICOT Technical Report: TR-710

TR-710

Scalability Analysis of Static Load Balancing under Unpredictable Subproblem Sizes

by

K. Kimura & N. Ichiyoshi

Nobember, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Scalability Analysis of Static Load Balancing under Unpredictable Subproblem Sizes

KOUICHI KIMURA AND NOBUYUKI ICHIYOSHI

Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan

Abstract

We investigate the balance of load between processors in parallel execution, in which a given problem consists of many subproblems of unpredictable different sizes. If we solve each subproblem at a different processor with a polynomial-time algorithm of degree $d \geq 1$, unevenness in the subproblem size is translated into larger unevenness (according to d) in the load between processors. However, we have found that we can almost balance the load between processors by assigning only a modest number of subproblems to each processor. Namely, an $\omega(\log^d p)$ number of subproblems per processor is sufficient, where p denotes the number of processors. Thus, a large parallel machine will be efficiently utilized in solving a modestly large problem of this kind with such a naïve load balancing strategy.

Formally, we are engaged in average-case analysis of the balance of load, assuming that the size of each subproblem is random and independently identically distributed according to a Poisson distribution. Our results are extensions of the results by Vitter and Flajolet on the average maximum bucket occupancy in hash tables.

1 Introduction

Recently, there has been growing interest in massively parallel computers. These will become indispensable in the future for solving larger and larger problems. In order to efficiently utilize parallel computers, we must confront matters that do not appear in sequential computation. Among others, *load balancing* is very important to all kinds of these parallel computers, and deserves to be studied in a general formulation because of its rather machine-independent nature. (In contrast, the overheads associated with interprocessor communications, another important concern in parallel processing, are likely to be affected by the characteristics of an individual machine such as the topology of interconnections and others [1]). In this paper, we investigate the balance of load between processors in parallel execution, where a given problem consists of many subproblems of unpredictable different sizes, each of which is independently solved with the same algorithm. For example, a problem concerning large geometric data (points, segments, polygons, etc.) in a planar region may consists of subproblems concerning those in the partitioned subregions. Or, a problem concerning a large tree may consists of subproblems concerning its subtrees. We assume any such decomposition of a problem is given.

One of the most naïve approaches to solving this kind of problem in parallel is to solve each subproblem at a different processor. However, such an approach is likely to incur large load imbalances due to non-uniform subproblem sizes. For example, if we are using a linear-time algorithm to solve each subproblem, load imbalance will occur just in proportion as the subproblems vary in size. Moreover, if we are using a polynomial-time algorithm of degree $d > 1$ to solve each subproblem, unevenness in the subproblem size is translated into larger unevenness (according to d) of the load between processors, as we will see in Section 3.

A simple remedy for such load imbalance is to assign a sufficient number of subproblems to each processor. Intuitively speaking, if we assign many subproblems to each processor, they will consist of from smaller to larger subproblems and will, hence, yield a comparable load between processors on average; thus we may expect a better result. In Section 5, we investigate this approach, and will see that in fact we can almost balance the load by assigning only a modest number of subproblems to each processor. Namely, an $\omega(\log^d p)$ number of subproblems per processor are shown to be sufficient for p processors, when we are solving each subproblem with a polynomial-time algorithm of degree d.

Similar intuition is behind the scattered decomposition, which has been successfully applied to balance the load for irregular matrix problems [2]. It divides a matrix regularly into a lot of fine-grained pieces and distributes them to the processors cyclically.

These approaches to load balancing are *static*, since the assignment of the subproblems to the processors is determined prior to execution. A more sophisticated approach is *dynamic* load-balancing, in which the subproblems are adaptively distributed to the processors that are currently lightly loaded [2, 3]. In general, the latter approach seems to outperform the former although it may incur some overheads, which is discussed elsewhere (*e.g.*, [7, 10]). The purpose of this paper is to investigate the static approach.

In this paper, we are concerned with the asymptotic behavior of the balance of load between processors as the number of processors p increases to infinity. In particular, we discuss how large problem size is required in order to exploit a large number of processors efficiently, namely, how the problem size should scale with the number of processors. By concentrating on asymptotic analysis with $p \to \infty$, we can obtain rather general results without referring to the details of machines. These results will be valuable, since the performance of an extra large parallel computation is properly predicted by its *scalability* [9]. This is just like the case in which discussions on the order of the computation time of a sequential algorithm provide us with a rough image with respect to its performance for large data.

Because of the randomness of the subproblem size which we usually observe in many combinatorial problems, we assume a probabilistic model and engage ourselves in average-case analysis. Formally, our results are extensions of the results by Vitter and Flajolet [11] on the average maximum bucket occupancy in hash tables, which owes much to Kolchin *et al.* [8].

The organization of this paper is as follows. In Section 2, we define a formal model of the balance of load between processors. Section 3 is devoted to the analysis of the balance of load when each processor solves a single subproblem. In Section 4, we prepare some mathematical prerequisites for the analysis in Section 5, which is concerned with the case when each processor solves a sufficient number of subproblems. Section 6 summarizes the results.

2 Model of the Balance of Load between Processors

In this section, we define a simple model of the balance of load between processors in a general formulation without referring to details of machines or problems being solved.

Let p be the number of processors, and $X_1, X_2, \ldots, X_m$ be the size (description length) of each subproblem, where m is the number of subproblems.

For the varying size of subproblems, we assume a Poisson model [11]. Namely, we assume that $X_1, \ldots, X_m$ are independent identically distributed (i.i.d.) random variables according to the Poisson distribution with parameter $\alpha > 0$. This is a natural assumption due to the classical law of small numbers, when each subproblem consists of many small parts that are distributed randomly and unbiasedly over all the subproblems. For example, if a lot of points are distributed randomly and unbiasedly over a planar region that is partitioned into disjoint subregions having the same area, the number of points in each subregion satisfies the above condition. Note that α represents the average size of a subproblem.

In solving each subproblem, we assume a polynomial-time algorithm. Namely, we assume that the computation time for solving a subproblem of description length n is given by a polynomial: $c(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_0$. Hence the computation time for solving the i-th subproblem is $c(X_i)$. Note that the average computation time for solving a subproblem is given by $E(c(X_i)) = c^*(\alpha)$, where $c^*(X)$ is defined by $c^*(X) = a_d^* X^d + \cdots + a_1^* X + a_0^*$ and $c(X) = a_d^* X^{(d)} + \cdots + a_1^* X^{(1)} + a_0^*$. Here $X^{(k)}$ denotes $X(X-1)\cdots(X-k+1)$.

How to assign each subproblem to the processors is specified by a mapping $\varpi : \{1, 2, \ldots, m\} \longrightarrow \{1, 2, \ldots, p\}$. In this paper, we consider only a simple assignment: assigning q subproblems to each processor, specified by $m = pq$ and $\varpi(j) = \lfloor (j + q - 1)/q \rfloor$ for some $q = 1, 2, \ldots$.

The amount of load at the i-th processor is given by: $L_i = \sum_{\varpi(j)=i} c(X_j) = \sum_{(i-1)q < j \leq iq} c(X_j)$ for $i = 1, 2, \ldots, p$. In an ideal case in which we can neglect other overheads such as communication latency, $\max_{1 \leq i \leq p} L_i$ represents the *parallel computation time* with p processors, and $L_1 + \cdots + L_p$ represents the *sequential computation time* with a single processor. We define the *load balance factor* by:

$$\eta = \frac{E(L_1 + \cdots + L_p)}{p \cdot E(\max\limits_{1 \leq i \leq p} L_i)} = \frac{q \cdot c^*(\alpha)}{E\left(\max\limits_{1 \leq i \leq p} \sum\limits_{j=1}^{q} c(X_{(i-1)q+j})\right)} \tag{2.1}$$

where $E(\max_{1 \leq i \leq p} L_i)$ represents the average parallel computation time, and $E(L_1 + \cdots + L_p)$ represents the average sequential computation time. Hence $E(L_1 + \cdots + L_p)/E(\max_{1 \leq i \leq p} L_i)$ indicates the expected *speedup*, and η indicates the *efficiency* (the speedup divided by the number of processors p), which also indicates the *processor utilization*. Clearly, we have $0 \leq \eta \leq 1$, and $\eta = 1$ if and only if $L_1 = \cdots = L_p$, which means a perfect balance of load. We will be interested in whether $\eta \to 1$ or not as the number of processors p increases to infinity either with $q = 1$ and increasing α (Section 3) or with increasing q and fixed α (Section 5).

Throughout this paper, we implicitly assume a standard probability space $(\Omega, \mathcal{B}, P)$, where Ω is the base space, $\mathcal{B} \subset 2^\Omega$ is the σ-algebra of events, and P is the underlying probability distribution over Ω. Basically we adopt frequently-used mathematical notations such as those in [5]. $\boldsymbol{R}$ denotes the set of real numbers, $\boldsymbol{R}_+$ the set of non-negative real numbers, and $\boldsymbol{N}$ the set of natural numbers. $X \vee Y$ denotes $\max(X, Y)$, and $X^{(k)}$ denotes $X(X-1)\cdots(X-k+1)$. For a random variable $X : \Omega \longrightarrow \boldsymbol{R}$, P^X denotes the distribution of X over $\boldsymbol{R}$. $P(A)$ denotes the probability of an event A, $E(X)$ does the expectation of a random variable X; while $P(A|C)$ and $E(X|C)$ denote those under condition C. We also employ the standard Landau's symbols:

$$a_p = O(b_p) \Leftrightarrow \limsup_{p\to\infty} |a_p|/b_p < \infty, \qquad a_p = o(b_p) \Leftrightarrow \lim_{p\to\infty} |a_p|/b_p = 0,$$
$$a_p = \Omega(b_p) \Leftrightarrow \liminf_{p\to\infty} a_p/b_p > 0, \qquad a_p = \omega(b_p) \Leftrightarrow \lim_{p\to\infty} a_p/b_p = \infty,$$

$$a_p = \Theta(b_p) \Leftrightarrow a_p = O(b_p) \text{ and } a_p = \Omega(b_p).$$

We usually use these symbols only for positive values except for $O(\cdot)$ and $o(\cdot)$. And besides,

$$a_p \sim b_p \Leftrightarrow \lim_{p\to\infty} a_p/b_p = 1, \qquad a_p \lesssim b_p \Leftrightarrow \limsup_{p\to\infty} a_p/b_p \le 1.$$

3 Assigning a Single Subproblem to Each Processor

In this section, we investigate the balance of load between processors when we solve each subproblem at a different processor (*i.e.*, $q = 1$ in the above model). Namely, we study the case in which the load at the i-th processor is given by $c(X_i)$, where X_i denotes the size of the i-th subproblem and $c(\cdot)$ the time-complexity function of the algorithm being used. So, the parallel computation time is given by $\max_{1\le i\le p} c(X_i)$. The next theorem gives an estimation of its expectation. This is a direct extension of the result by Vitter and Flajolet [11] on the average maximum bucket occupancy in hash tables, which owes much to Kolchin *et al.* [8].

Theorem 3.1 *Let $c : \boldsymbol{R}_+ \longrightarrow \boldsymbol{R}_+$ be a non-decreasing (possibly except for small values) polynomial function of degree $d \ge 1$. And let $X_1, \ldots, X_p$ be i.i.d. according to the Poisson distribution with parameter $\alpha > 0$. Then we have*

$$E\{\max_{1\le i\le p} c(X_i)\} \sim \begin{cases} c(b(\alpha, p^{-1})) & \text{as } p \to \infty \text{ with } \alpha = o(\log p) \\ c(\alpha) & \text{as } p \to \infty \text{ with } \alpha = \omega(\log p) \end{cases}$$

and

$$\max_{1\le i\le p} c(X_i) - c(\alpha) \sim c'(\alpha)\sqrt{2\alpha \log p} \quad (\text{in probability}) \quad \text{as } p \to \infty \text{ with } \alpha = \omega(\log p)$$

Here we define $b = b(\alpha, \varepsilon) \in \boldsymbol{N}$ by $b > \alpha$ and

$$\frac{e^{-\alpha}\alpha^{b+1}}{(b+1)!} \le \varepsilon < \frac{e^{-\alpha}\alpha^b}{b!} \tag{3.1}$$

for $\alpha > 0$ and $0 < \varepsilon < e^{-\alpha}$. In particular, when $\alpha = \Theta(1)$, we have $b(\alpha, p^{-1}) \sim \log p/\log\log p$.

Proof: For conciseness, we write $c_n = c(n)$ for $n \in \boldsymbol{N}$, and use the following notations within this proof.

$$e_n(\alpha) = \sum_{j=0}^{n} \frac{\alpha^j}{j!} \ : \ \text{truncated exponential function up to the } n\text{-th term} \quad (n = 0, 1, 2, \ldots)$$

$$\rho_n = \rho(\alpha, p, n) = P(\max_{1\le i\le p} X_i \le n) = \{e^{-\alpha} e_n(\alpha)\}^p \qquad (n = 0, 1, 2, \ldots), \qquad \rho_{-1} = 0$$

Then, we get an expression:

$$E\{\max_{1\leq i\leq p} c(X_i)\} = \sum_{n=0}^{\infty} c_n(\rho_n - \rho_{n-1}) = c_0 + \sum_{n=0}^{\infty}(c_{n+1} - c_n)(1 - \rho_n) \tag{3.2}$$

First we shall consider the case with $\alpha = o(\log p)$. For a sufficiently large p, $b = b(\alpha, p^{-1}) \in \boldsymbol{N}$ satisfying $b > \alpha$ and condition (3.1) exists, since $e^{-\alpha}\alpha^n/n!$ is decreasing in $n \geq \alpha$ and $e^{-\alpha}\alpha^\alpha/\Gamma(\alpha+1) \sim 1/\sqrt{2\pi\alpha} = \omega(p^{-1})$. Let us define $\lambda = \lambda(\alpha, p)$ by $e^{-\alpha}\alpha^{b+1}/(b+1)! = \lambda/p$. Clearly, we have $\alpha/(b+1) < \lambda \leq 1$. We also define: $r = r(\alpha, p) = \alpha/b(\alpha, p^{-1})$. Then we can easily see that $r \to 0$ as $p \to \infty$ with $\alpha = o(\log p)$, using Stirling's formula (*cf*. Lemma 2 in §6, Chapter III in [8]).

Now we have for each $k = 1, 2, \ldots,$

$$\rho_{b+k} = \left(1 - e^{-\alpha}\sum_{n=b+k+1}^{\infty}\frac{\alpha^n}{n!}\right)^p \geq \left(1 - \frac{e^{-\alpha}\alpha^{b+1}}{(b+1)!}\sum_{n=k}^{\infty} r^n\right)^p \geq \left(1 - \frac{1}{p}\cdot\frac{r^k}{1-r}\right)^p \geq 1 - \frac{r^k}{1-r}$$

Hence we obtain

$$\sum_{n>b}(c_{n+1} - c_n)(1 - \rho_n) \leq \sum_{k=1}^{\infty} c_{b+k+1}\cdot\frac{r^k}{1-r} \sim c_b r = o(c_b) \quad \text{as } p \to \infty \tag{3.3}$$

since c_n is a polynomial function of n.

Similarly, we have

$$\rho_{b-2} = \left(1 - e^{-\alpha}\sum_{n=b-1}^{\infty}\frac{\alpha^n}{n!}\right)^p \sim \left(1 - \frac{e^{-\alpha}\alpha^{b+1}}{(b+1)!}\sum_{n=-2}^{\infty} r^n\right)^p \sim \left(1 - \frac{\lambda}{pr^2}\right)^p \leq e^{-\lambda/r^2} \to 0 \quad \text{as } p \to \infty$$

Hence we obtain

$$\begin{aligned} &\sum_{n=0}^{b}(c_{n+1} - c_n)(1 - \rho_n) \\ = \;& c_{b+1}(1 - \rho_b) + c_b(\rho_b - \rho_{b-1}) + c_{b-1}(\rho_{b-1} - \rho_{b-2}) + \sum_{n=1}^{b-2} c_n(\rho_n - \rho_{n-1}) + c_0(\rho_0 - 1) \\ \sim \;& c_b \quad \text{as } p \to \infty \end{aligned} \tag{3.4}$$

since the sum of the first three terms is $\sim c_b$ and the fourth term is $\leq c_{b-2}\rho_{b-2} = o(c_b)$. From expressions (3.2), (3.3), and (3.4), we obtain

$$E\left(\max_{1\leq i\leq p} c(X_i)\right) \sim c(b(\alpha, p^{-1})) \quad \text{as } p \to \infty \text{ with } \alpha = o(\log p)$$

And it is easily verified that $b(\alpha, p^{-1}) \sim \log p/\log\log p$ for $\alpha = \Theta(1)$, using Stirling's formula (*cf*. p.501 in [11]). These complete the proof for the case with $\alpha = o(\log p)$.

Next let us consider the case with $\alpha = \omega(\log p)$. By integration by parts, we have for $n > \alpha$,

$$1 - e^{-\alpha}e_n(\alpha) = \frac{e^{-\alpha}\alpha^{n+1}}{n!}\int_0^1(1-\theta)^n e^{\alpha\theta}d\theta \tag{3.5}$$

Here the integral in the right side is less than $1/(n-\alpha)$, since $(1-\theta)^n \leq e^{-n\theta}$ for $0 \leq \theta < 1$. Hence we have

$$1-\rho_n = 1-\{e^{-\alpha}e_n(\alpha)\}^p \leq p\cdot\{1-e^{-\alpha}e_n(\alpha)\} < \frac{pe^{-\alpha}\alpha^{n+1}}{n!\cdot(n-\alpha)}$$

In particular, for $n = \lfloor 2\alpha \rfloor$,

$$1-\rho_{\lfloor 2\alpha \rfloor} \lesssim \frac{pe^{-\alpha}\alpha^{2\alpha+1}}{\alpha\Gamma(2\alpha)} \sim p\left(\frac{e}{4}\right)^{\alpha}\sqrt{\frac{\alpha}{\pi}}$$

and $1-\rho_{\lfloor 2\alpha\rfloor+k} \lesssim p(e/4)^{\alpha}\sqrt{\alpha/\pi}\cdot 2^{-k}$ for $k \in \boldsymbol{N}$. Hence, we obtain

$$(3.6) \qquad \sum_{n>\lfloor 2\alpha \rfloor}(c_{n+1}-c_n)(1-\rho_n) \lesssim p\left(\frac{e}{4}\right)^{\alpha}\sqrt{\frac{\alpha}{\pi}}\sum_{k=1}^{\infty}c_{\lfloor 2\alpha\rfloor+k+1}2^{-k} \to 0 \quad \text{as } p\to\infty$$

Now, let κ be a positive constant, $\nu = \lfloor \alpha+\sqrt{2\alpha\kappa\log p}\rfloor$, and consider ρ_ν.

Lemma 3.1 *For any fixed $\kappa > 0$, we have $\log\log(1/\rho_\nu) = \{1-\kappa+o(1)\}\log p$ as $p\to\infty$. In particular, $\rho_\nu \to 0$ for $0<\kappa<1$, and $\rho_\nu\to 1$ for $\kappa > 1$.*

PROOF: By Equation (3.5), we have

$$\log\rho_\nu = p\log\left\{1-\frac{e^{-\alpha}\alpha^{\nu+1}}{\nu!}\int_0^1(1-\theta)^\nu e^{\alpha\theta}d\theta\right\}$$

For $\gamma = \alpha^{-1/2}(\log p)^{-1/3}$, since $(\nu-\alpha)\gamma\to\infty$ and $\nu\gamma^2\to 0$ as $p\to\infty$, we have

$$\int_0^\gamma(1-\theta)^\nu e^{\alpha\theta}d\theta \sim \int_0^\gamma e^{-(\nu-\alpha)\theta}d\theta \sim \frac{1}{\nu-\alpha} \sim \frac{1}{\sqrt{2\alpha\kappa\log p}}$$

$$\int_\gamma^1(1-\theta)^\nu e^{\alpha\theta}d\theta \leq \int_\gamma^1 e^{-(\nu-\alpha)\theta}d\theta = \frac{e^{-(\nu-\alpha)\gamma}-e^{-(\nu-\alpha)}}{\nu-\alpha} = o\left(\frac{1}{\nu-\alpha}\right)$$

Hence

$$\frac{e^{-\alpha}\alpha^{\nu+1}}{\nu!}\int_0^1(1-\theta)^\nu e^{\alpha\theta}d\theta \sim \frac{e^{\nu-\alpha}}{2\sqrt{\pi\kappa\log p}}\left(\frac{\alpha}{\nu}\right)^\nu$$

and the logarithm of its dominant factor is

$$\nu-\alpha+\nu\log\frac{\alpha}{\nu} = -\frac{(\nu-\alpha)^2}{2\alpha}+O\left(\frac{(\nu-\alpha)^3}{\alpha^2}\right) = -\kappa\log p+O\left(\sqrt{\frac{\log p}{\alpha}}\right)\cdot\log p$$

From these we obtain the claim. ∎

PROOF OF THEOREM 3.1 (CONTINUED): Taking $\kappa = 2$ and $\nu = \lfloor\alpha+2\sqrt{\alpha\log p}\rfloor$, we have $\rho_\nu\to 1$ and $\rho_{\lfloor\alpha\rfloor}\to 0$ as $p\to\infty$ due to the above Lemma. Hence

$$(3.7) \qquad \sum_{n=\nu}^{\lfloor 2\alpha\rfloor}(c_{n+1}-c_n)(1-\rho_n) \leq c_{\lfloor 2\alpha\rfloor+1}(1-\rho_\nu) = o(c(\alpha)) \quad \text{as } p\to\infty$$

$$(3.8) \qquad \sum_{n=\lfloor\alpha\rfloor}^{\nu-1} (c_{n+1} - c_n)(1-\rho_n) \le c_\nu - c_{\lfloor\alpha\rfloor} = O((\nu-\alpha)c'(\nu)) = o(c(\alpha)) \quad \text{as } p \to \infty$$

$$(3.9) \qquad \sum_{n=0}^{\lfloor\alpha\rfloor-1} (c_{n+1} - c_n)(1-\rho_n) \sim (c_{\lfloor\alpha\rfloor} - c_0)(1-\rho_{\lfloor\alpha\rfloor}) \sim c(\alpha) \quad \text{as } p \to \infty$$

Therefore, from expressions (3.2), (3.6), (3.7), (3.8) and (3.9), we obtain

$$E\{\max_{1\le i\le p} c(X_i)\} \sim c(\alpha) \quad \text{as } p \to \infty \text{ with } \alpha = \omega(\log p)$$

Moreover, for arbitrary $0 < \kappa_0 < 1 < \kappa_1$, we have $P(\alpha + \sqrt{2\alpha\kappa_0 \log p} \le \max_{1\le i\le p} X_i \le \alpha + \sqrt{2\alpha\kappa_1 \log p}) \to 1$ as $p \to \infty$, by Lemma 3.1. Hence

$$P\left(\sqrt{\kappa_0} < \frac{\max_{1\le i\le p} c(X_i) - c(\alpha)}{c'(\alpha)\sqrt{2\alpha\log p}} < \sqrt{\kappa_1}\right) \to 1 \quad \text{as } p \to \infty \quad \text{for } 0 < \forall\kappa_0 < 1 < \forall\kappa_1$$

This implies $\max_{1\le i\le p} c(X_i) - c(\alpha) \sim c'(\alpha)\sqrt{2\alpha\log p}$ in probability. These complete the proof for the case with $\alpha = \omega(\log p)$. ∎

As an immediate corollary, we obtain the following result. It tells us how large the average subproblem size (α) should be in order to attain good load balance by assigning only a single subproblem to each processor, when the number of processors p increases.

Corollary 3.1 *Under the assumptions of Theorem 3.1, we have*

(i) $\eta \to 0$ *as* $p \to \infty$ *with* $\alpha = o(\log p)$

(ii) $\eta \to 1$ *as* $p \to \infty$ *with* $\alpha = \omega(\log p)$

where η *is the load balance factor defined by Equation* (2.1)

4 Partial Order for Probability Distributions on R_+

In this section we prepare some mathematical prerequisites for the analysis in the next section. Namely, we introduce a partial order on the class of probability distributions on $\boldsymbol{R}_+$ and give its basic properties. Within this section, $\mathcal{M}$ denotes the class of non-negative random variables with finite expectation. ($X \vee Y$ denotes $\max(X,Y)$ throughout this paper.)

Definition 4.1 For $X, Y \in \mathcal{M}$, we define $P^X \prec P^Y$ (or simply, $X \prec Y$), if and only if $E(X \vee c) \le E(Y \vee c)$ for all $c \ge 0$.

Strictly speaking, the binary relation $\prec$ is defined on the class of probability distributions rather than the class of random variables, as one can see in the next proposition. However, for conciseness, we will usually write $X \prec Y$ instead of $P^X \prec P^Y$.

This inequality $\prec$ is weaker than the stochastic inequality [6], as we will show in the next proposition. Let μ and ν be probability distributions over $\boldsymbol{R}_+$. μ is said to be *stochastically smaller* than ν, and denoted by $\mu <_{st} \nu$, if and only if their distribution functions, $\phi_\mu(t) = \mu([0,t])$ and $\phi_\nu(t) = \nu([0,t])$, satisfy $\phi_\mu(t) \geq \phi_\nu(t)$ for all $t \geq 0$.

Proposition 4.1 *Let $X, Y, Z, X_n, Y_n \in \mathcal{M}$, where $n = 1, 2, \ldots$.*

(i) *If $X \leq Y$ a.s. (almost surely), then $X \prec Y$.*

(ii) *If $X \prec Y$ and $Y \prec Z$, then $X \prec Z$.*

(iii) *If $X \prec Y$ and $Y \prec X$, then $P^X = P^Y$.*

(iv) *If $X \prec Y$, then $E(X) \leq E(Y)$.*

(v) *If $P^X <_{st} P^Y$, then $X \prec Y$.*

(vi) *Assume that P^{X_n} and P^{Y_n} converge to P^X and P^Y respectively as $n \to \infty$. If $X_n \prec Y_n$ for $\forall n$, then $X \prec Y$.*

Proof: Since (i), (ii), (iv) and (vi) are trivial by definition, we shall only prove (iii) and (v). Let $\phi_X(t) = P(X \leq t)$ and $\phi_Y(t) = P(Y \leq t)$ be the distribution functions of X and Y respectively. And define $\psi_X(t)$ and $\psi_Y(t)$ by

$$\psi_X(t) = \int_t^\infty (x-t)d\phi_X(x), \qquad \psi_Y(t) = \int_t^\infty (y-t)d\phi_Y(y)$$

Then we have

$$\begin{array}{llll} \psi_X(+\infty) = 0, & \psi'_X(t) = \phi_X(t) - 1, & E(X \vee t) = t + \psi_X(t) & (\forall t \geq 0) \\ \psi_Y(+\infty) = 0, & \psi'_Y(t) = \phi_Y(t) - 1, & E(Y \vee t) = t + \psi_Y(t) & (\forall t \geq 0) \end{array}$$

(iii) If $X \prec Y$ and $Y \prec X$, then $\psi_X \equiv \psi_Y$, $\phi_X \equiv \phi_Y$ and $P^X = P^Y$ follow.

(v) $P^X <_{st} P^Y$ implies $\phi_X \geq \phi_Y$ ([6]). Hence $\psi_X \leq \psi_Y$ and $X \prec Y$ follow. ∎

Proposition 4.2 *Let $X, Y, Z \in \mathcal{M}$ and Z is independent of X and Y.*

(i) *If $X \prec Y$, then $X \vee Z \prec Y \vee Z$.*

(ii) *If $X \prec Y$, then $X + Z \prec Y + Z$.*

(iii) *If $X \prec Y$ and $YZ \in \mathcal{M}$, then $XZ \prec YZ$.*

(iv) *Let $f : \boldsymbol{R}_+^2 \to \boldsymbol{R}_+$ be a (Lebesgue) measurable function and $f(X,z), f(Y,z) \in \mathcal{M}$ for $\forall z \geq 0$. If $f(X,z) \prec f(Y,z)$ for $\forall z \geq 0$, then $f(X,Z) \prec f(Y,Z)$.*

Proof: Since (i), (ii) and (iii) follow from (iv), we shall only prove (iv). For arbitrary $c \geq 0$, since $f(X,z), f(Y,z) \in \mathcal{M}$ for $\forall z \geq 0$,

$$\begin{aligned} E\{f(X,Z) \vee c\} &= \int_0^\infty E\{f(X,z) \vee c \mid Z = z\} P^Z(dz) = \int_0^\infty E\{f(X,z) \vee c\} P^Z(dz) \\ &\leq \int_0^\infty E\{f(Y,z) \vee c\} P^Z(dz) = \int_0^\infty E\{f(Y,z) \vee c \mid Z = z\} P^Z(dz) = E\{f(Y,Z) \vee c\} \end{aligned}$$

Thus we obtain $f(X,Z) \prec f(Y,Z)$. ∎

Corollary 4.1 *Let $X_1, \ldots, X_n \in \mathcal{M}$ and $Y_1, \ldots, Y_n \in \mathcal{M}$ be i.i.d. according to μ and ν respectively. If $\mu \prec \nu$, then*

$$E(\max_{1 \leq i \leq n} X_i) \leq E(\max_{1 \leq i \leq n} Y_i)$$

Proposition 4.3 *Let $f : (\boldsymbol{R}_+{}^n)^r \longrightarrow \boldsymbol{R}_+$ be a measurable function such that*

$$\begin{aligned} & f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) + f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) \\ & \qquad \leq f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) + f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) \end{aligned}$$

$$\begin{aligned} & \max\{f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r),\ f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r)\} \\ & \qquad \leq \max\{f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r),\ f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r)\} \end{aligned}$$

for $2 \leq \forall k \leq r$ and $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_r \in \boldsymbol{R}_+{}^n$. And let $g : (\boldsymbol{R}_+{}^n)^r \longrightarrow \boldsymbol{R}_+$ be a measurable function such that

$$g(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = g(\boldsymbol{x}_{\sigma_1}, \ldots, \boldsymbol{x}_{\sigma_r}) \quad \text{for } \forall \sigma \in \mathcal{S}_r,\ \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \boldsymbol{R}_+{}^n$$

where $\mathcal{S}_r$ denotes the symmetric group of degree r. And let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_r$ be i.i.d. random vectors of dimension n, and $E[f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_r)], E[g(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_r)] < +\infty$. Then

$$(4.1) \quad f(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r) + g(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r) \prec f(\boldsymbol{X}_1, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r)$$

Proof: By induction on k, it is sufficient to show that

$$\begin{aligned} & f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_1, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) + g(\boldsymbol{X}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) \\ & \prec f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_1, \boldsymbol{X}_1, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) + g(\boldsymbol{X}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) \end{aligned}$$

for each $k = 2, 3, \ldots, r$ and arbitrary $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r \in \boldsymbol{R}_+^n$. Since this reduces to (4.1) with $r = 2$, we may assume $r = 2$ from the beginning. Note that the assumption on f implies

$$f(\boldsymbol{x}, \boldsymbol{y}) \vee c + f(\boldsymbol{y}, \boldsymbol{x}) \vee c \leq f(\boldsymbol{x}, \boldsymbol{x}) \vee c + f(\boldsymbol{y}, \boldsymbol{y}) \vee c \quad \text{for } \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}_+^n,\ \forall c > 0$$

Hence, for arbitrary $c \geq 0$,

$$\begin{aligned}
& 2E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c] \\
= \;& E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c + \{f(\boldsymbol{X}_2, \boldsymbol{X}_1) + g(\boldsymbol{X}_2, \boldsymbol{X}_1)\} \vee c] \\
\leq \;& E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c + \{f(\boldsymbol{X}_2, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c] \\
= \;& 2E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c]
\end{aligned}$$

Thus the claim is established. ∎

As a special case of this proposition, we obtain the following.

Corollary 4.2 *Let $X_1, \ldots, X_r \in \mathcal{M}$ be i.i.d. and assume that $Y_1, \ldots, Y_r, Z \in \mathcal{M}$ are independent of $X_1, \ldots, X_r$. Then we have*

$$X_1 Y_1 + \cdots + X_r Y_r + Z \prec X_1(Y_1 + \cdots + Y_r) + Z$$

5 Assigning a Sufficient Number of Subproblems to Each processor

In this section, we investigate the balance of load between processors when we solve $q > 1$ subproblems at each processor. We begin with some technical lemmas.

Define $\Delta^d : (\boldsymbol{R}_+^n)^d \longrightarrow \boldsymbol{R}_+$ by

$$\Delta^d(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d) = \sum_{1 \leq j_1 < \cdots < j_d \leq n} x_{1j_1} \cdots x_{d\,j_d} \qquad \text{for} \quad \boldsymbol{x}_i = (x_{i1}, \ldots, x_{in}) \qquad (1 \leq i \leq d)$$

For conciseness, we write $\Delta^d(\boldsymbol{x}) = \Delta^d(\boldsymbol{x}, \ldots, \boldsymbol{x})$ and $\Delta^{d_1,d_2}(\boldsymbol{x}, \boldsymbol{y}) = \Delta^{d_1+d_2}(\underbrace{\boldsymbol{x}, \ldots, \boldsymbol{x}}_{d_1}, \underbrace{\boldsymbol{y}, \ldots, \boldsymbol{y}}_{d_2})$.

Lemma 5.1 *Let X be a Poisson random variable with parameter $\alpha > 0$. For $\forall n \geq d$, let $X_{n,1}, \ldots, X_{n,n}$ be i.i.d. such that $P(X_{n,j} = 1) = \alpha/n$ and $P(X_{n,j} = 0) = 1 - \alpha/n$, and $\boldsymbol{X}_n = (X_{n,1}, \ldots, X_{n,n})$. Then*

$$X^{(d)} \left(\overset{\text{def}}{=} X(X-1)\cdots(X-d+1)\right) = d! \lim_{n\to\infty} \Delta^d(\boldsymbol{X}_n) \qquad \textit{(convergence in law)}$$

Proof: Since $X_{n,i} = 0$ or 1 for each $1 \leq i \leq n$,

$$(X_{n,1} + \cdots + X_{n,n}) \sum_{\substack{i_1,\ldots,i_d \leq n \\ \text{(distinct)}}} X_{n,i_1} \cdots X_{n,i_d} = \sum_{\substack{i_1,\ldots,i_{d+1} \leq n \\ \text{(distinct)}}} X_{n,i_1} \cdots X_{n,i_{d+1}} + d \sum_{\substack{i_1,\ldots,i_d \leq n \\ \text{(distinct)}}} X_{n,i_1} \cdots X_{n,i_d}$$

namely, $(X_{n,1} + \cdots + X_{n,n} - d) \cdot d! \cdot \Delta^d(\boldsymbol{X}_n) = (d+1)! \cdot \Delta^{d+1}(\boldsymbol{X}_n)$ holds. Hence we obtain $(X_{n,1} + \cdots + X_{n,n})^{(d)} = d! \cdot \Delta^d(\boldsymbol{X}_n)$ by induction on d. Here the distribution of $X_{n,1} + \cdots + X_{n,n}$ converges to the Poisson distribution with parameter α due to the classical law of small numbers. Therefore we obtain the desired result. ∎

Lemma 5.2 *Let $\{X_{ij}\}_{1\le i\le r, 1\le j\le n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{in})$ for $1 \le i \le r$. Then, for $\forall d, s = 0, 1, 2, \ldots,$ we have*

$$\sum_{i=1}^{r} \Delta^{d+1,s}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \Delta^{d,s+1}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) \tag{5.1}$$

Proof: Since both sides of (5.1) are identical for $d = 0$, we may assume that $d \ge 1$. For $k = 0, 1, \ldots, n-1$, we define $\Delta_{k+1}^{d+1} : \boldsymbol{R}_+^n \times \boldsymbol{R}_+^n \times \boldsymbol{R}_+^n \longrightarrow \boldsymbol{R}_+$ by

$$\begin{aligned}
\Delta_{k+1}^{d+1,s}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \ &= \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k+1}} x_{j_1}\cdots x_{j_{d+1}} z_{j_{d+2}} \cdots z_{j_{d+s+1}} \\
&\quad + \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\ge k+2}} y_{j_1}\cdots y_{j_d} x_{j_{d+1}} z_{j_{d+2}} \cdots z_{j_{d+s+1}} \\
&\quad \text{for } \ \boldsymbol{x} = (x_1, \ldots, x_n), \quad \boldsymbol{y} = (y_1, \ldots, y_n), \quad \boldsymbol{z} = (z_1, \ldots, z_n)
\end{aligned}$$

Fixing an arbitrary $1 \le k < n$, we write $\hat{\boldsymbol{X}}_i = (X_{i1}, \ldots, X_{ik})$ for $1 \le i \le r$. Taking an arbitrary $(x_{ij})_{1\le i\le r,\, k+1\le j\le n} \in \boldsymbol{R}_+^{r(n-k)}$, we define $a(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r)$ by

$$a(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r) = \sum_{i=1}^{r} \Delta_{k+1}^{d+1,s}(\bar{\boldsymbol{X}}_i, \bar{\boldsymbol{X}}_1, \sum_{h=1}^{r} \bar{\boldsymbol{X}}_h)$$

where $\bar{\boldsymbol{X}}_i = (X_{i1}, \ldots, X_{ik}, x_{i\,k+1}, \ldots, x_{in})$ for $1 \le \forall i \le r$. Since

$$\begin{aligned}
\Delta_{k+1}^{d+1,s}(\bar{\boldsymbol{X}}_i, \bar{\boldsymbol{X}}_1, \sum_{h=1}^{r} \bar{\boldsymbol{X}}_h) \ &= \sum_{\ell=d+2}^{d+s+2} \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k,\ j_{\ell-1}\le k<j_\ell}} X_{ij_1}\cdots X_{ij_{d+1}} Z_{j_{d+2}} \cdots Z_{j_{\ell-1}} z_{j_\ell} \cdots z_{j_{d+s+1}} \\
&\quad + \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}=k+1}} X_{ij_1}\cdots X_{ij_d} x_{i\,k+1} z_{j_{d+2}} \cdots z_{j_{d+s+1}} \\
&\quad + \sum_{\ell=1}^{d+1} \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\ge k+2,\ j_{\ell-1}\le k<j_\ell}} X_{1j_1}\cdots X_{1j_{\ell-1}} x_{1j_\ell} \cdots x_{1j_d} x_{ij_{d+1}} z_{j_{d+2}} \cdots z_{j_{d+s+1}}, \\
&\text{where} \quad Z_j \overset{\text{def}}{=} \sum_{h=1}^{r} X_{hj} \ \ (1 \le j \le k), \quad z_j \overset{\text{def}}{=} \sum_{h=1}^{r} x_{hj} \ \ (k+1 \le j \le n),
\end{aligned}$$

we have $a(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r) = f(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r) + g(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r)$, where

$$\begin{aligned}
f(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r) \ &= \sum_{i=1}^{r} \left(u_i \sum_{1\le j_1<\cdots<j_d\le k} X_{ij_1}\cdots X_{ij_d} \right) + \sum_{\ell=1}^{d+1} \left(v_\ell \sum_{1\le j_1<\cdots<j_{\ell-1}\le k} X_{1j_1}\cdots X_{1j_{\ell-1}} \right) \\
g(\hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_r) \ &= \sum_{\ell=d+2}^{d+s+2} \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k,\ j_{\ell-1}\le k<j_\ell}} \left\{ \left(\sum_{i=1}^{r} X_{ij_1}\cdots X_{ij_{d+1}} \right) Z_{j_{d+2}} \cdots Z_{j_{\ell-1}} z_{j_\ell} \cdots z_{j_{d+s+1}} \right\}
\end{aligned}$$

and u_i, v_ℓ are non-negative constants determined by k and $\{x_{ij}\}_{1\le i\le r,\, k+1\le j\le n}$ for $1 \le \forall i \le r$, $1 \le \forall \ell \le d+1$. Note that

$$\sum_{1\le j_1<\cdots<j_\ell\le k} X_{ij_1}\cdots X_{ij_\ell} = \binom{X_{i1} + \cdots + X_{ik}}{\ell} \qquad (1 \le \forall i \le r,\ 0 \le \forall \ell \le d+1)$$

since we have $X_{ij} = 0$ or 1 by assumption. Thus we can apply Proposition 4.3 to obtain $a(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_r) \prec f(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_1) + g(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_r)$. Hence by Proposition 4.2 (iv), we obtain

$$\sum_{i=1}^{r} \Delta_{k+1}^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \sum_{i=1}^{r} \Delta_{k}^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h)$$

Since this holds for arbitrary $1 \leq k < n$, we obtain

$$\sum_{i=1}^{r} \Delta_{n}^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \sum_{i=1}^{r} \Delta_{1}^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h)$$

This is the desired result. ▮

Lemma 5.3 *Let $\{X_{ij}\}_{1 \leq i \leq r^2, 1 \leq j \leq n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{in})$ for $1 \leq i \leq r^2$. Then, for $\forall d, s = 0, 1, 2, \ldots$, we have*

$$\sum_{i=1}^{r} \Delta^{d,s}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_{i+(h-1)r}) \prec \sum_{i=1}^{r} \Delta^{d,s}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_h)$$

PROOF: This claim is immediately established by repeatedly applying Corollary 4.2. Namely, in dominating the left side, we replace $X_{i+(h-1)r\,j}$ by $X_{h'j}$ for each $(i, h, j) \in \{1, \ldots, r\} \times \{1, \ldots, r\} \times \{1, \ldots, n\}$, where $1 \leq h' \leq r$ and $h' \equiv i + h - 1 \bmod r$. ▮

Lemma 5.4 *Let $\{X_{ij}\}_{1 \leq i \leq r^d, 1 \leq j \leq n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{in})$ for $1 \leq i \leq r^d$. Then we have*

$$\sum_{1 \leq i \leq r^d} \Delta^d(\boldsymbol{X}_i) \prec \Delta^d(\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_r)$$

PROOF: By Lemma 5.2, we have

$$\sum_{i=1}^{r} \Delta^d(\boldsymbol{X}_i) \prec \Delta^{d-1,1}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h)$$

Consider the sum of r such expressions as this right side with distinct indices. Successively applying Proposition 4.2 (i), Lemma 5.3 and Lemma 5.2 to it, we obtain

$$\sum_{i=1}^{r^2} \Delta^d(\boldsymbol{X}_i) \prec \sum_{i=1}^{r} \Delta^{d-1,1}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_{i+(h-1)r}) \prec \sum_{i=1}^{r} \Delta^{d-1,1}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \Delta^{d-2,2}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h)$$

Applying again Proposition 4.2 (i), Lemma 5.3 and Lemma 5.2 to the sum of r such expressions as this rightmost side with distinct indices and so on, we finally obtain

$$\sum_{i=1}^{r^d} \Delta^d(\boldsymbol{X}_i) \prec \Delta^{0,d}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) = \Delta^d(\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_r)$$

This completes the proof. ▮

Theorem 5.1 *Let $\{X_i\}_{1\le i\le r^d}$ be i.i.d. Poisson random variables with parameter $\alpha > 0$, and $\tilde{X}$ be a Poisson random variable with parameter $r\alpha$, where $r \in \boldsymbol{N}$. Then we have*

$$\sum_{1\le i\le r^d} X_i^{(d)} \prec \tilde{X}^{(d)}$$

PROOF: Let n be an arbitrary integer greater than d, and $(X_{inj})_{1\le i\le r^d,\,1\le j\le n}$ be i.i.d. such that $P(X_{inj}=1)=\alpha/n$ and $P(X_{inj}=0)=1-\alpha/n$. We write $\boldsymbol{X}_{in}=(X_{in1},\dots,X_{inn})$ for $1\le \forall i\le r^d$. By Lemma 5.4, we have $\sum_{1\le i\le r^d}\Delta^d(\boldsymbol{X}_{in}) \prec \Delta^d(\boldsymbol{X}_{1n}+\cdots+\boldsymbol{X}_{rn})$. Letting $n\to\infty$, we obtain the desired result due to Lemma 5.1 and Proposition 4.1 (vi). ■

Theorem 5.2 *Let $\{X_{ij}\}_{1\le i\le p,1\le j\le q}$ be i.i.d. Poisson random variables with parameter $\alpha > 0$.*

(i) *For $p\to\infty$ with $q=\omega(\log^d p)$, we have:*

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} X_{ij}^{(d)}\right) \sim q\alpha^d$$

(ii) *For $p\to\infty$ with $q=o(\log^d p)$, we have:*

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} X_{ij}^{(d)}\right) \lesssim b(\alpha q^{1/d},1/p)^d$$

Here $b=b(\alpha,\varepsilon)\in\boldsymbol{N}$ is defined by $b>\alpha$ and condition (3.1) for $\alpha>0$ and $0<\varepsilon<e^{-\alpha}$.

PROOF: By Theorem 5.1, Theorem 3.1 and Corollary 4.1, we can immediately establish (ii), and obtain $E\left(\max_{1\le i\le p}\sum_{1\le j\le q}X_{ij}^{(d)}\right)\lesssim q\alpha^d$ in the case of (i). Since $E\left(\max_{1\le i\le p}\sum_{1\le j\le q}X_{ij}^{(d)}\right)\ge E\left(\sum_{1\le j\le q}X_{1j}^{(d)}\right)\sim q\alpha^d$ in the case of (i), the proof is completed. ■

Now, let us turn back to the discussion of the balance of load between processors when we assign q subproblems to each processor. The load at the i-th processor is given by $\sum_{1\le j\le q}c(X_{ij})$, where X_{ij} denotes the size of each subproblem ($1\le i\le p$, $1\le j\le q$) and $c(\cdot)$ the time-complexity function of the algorithm being used. So, the parallel computation time is given by $\max_{1\le i\le p}\sum_{1\le j\le q}c(X_{ij})$. Its expectation is immediately estimated according to the above theorem, as shown below:

Corollary 5.1 *Let $c(X)=a_dX^d+\cdots+a_1X+a_0$ be a polynomial of degree $d>0$ with non-negative coefficients and $\{X_{ij}\}_{1\le i\le p,1\le j\le q}$ be i.i.d. Poisson random variables with parameter $\alpha>0$.*

(i) *For $p\to\infty$ with $q=\omega(\log^d p)$, we have:*

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} c(X_{ij})\right) \sim qc^*(\alpha)$$

where $c^*(X)$ *is defined by* $c^*(X) = a_d^* X^d + \cdots + a_1^* X + a_0^*$ *and* $c(X) = a_d^* X^{(d)} + \cdots + a_1^* X^{(1)} + a_0^*$.

(ii) *For* $p \to \infty$ *with* $q = o(\log^d p)$, *we have:*

$$E\left(\max_{1 \le i \le p} \sum_{1 \le j \le q} c(X_{ij})\right) \lesssim a_d b(\alpha q^{1/d}, 1/p)^d$$

As an immediate corollary, we obtain the following result. It tells us that we can balance the load between processors by assigning an $\omega(\log^d p)$ number of subproblems to each processor, when the number of processors p increases while the average subproblem size is kept constant.

Corollary 5.2 *Under the assumptions of Corollary 5.1, we have* $\eta \to 1$ *as* $p \to \infty$ *with* $q = \omega(\log^d p)$, *where* η *is the load balance factor defined by Equation* (2.1).

In contrast, when $q = o((\log p / \log\log p)^d)$, we have $\eta \to 0$ as $p \to \infty$, since

$$E\left(\max_{1 \le i \le p} \sum_{1 \le j \le q} c(X_{ij})\right) \ge E\left(\max_{\substack{1 \le i \le p \\ 1 \le j \le q}} c(X_{ij})\right) \sim c(b(\alpha, (pq)^{-1})) = \Theta\left(\left(\frac{\log p}{\log\log p}\right)^d\right) = \omega(q)$$

due to Theorem 3.1. Thus the sufficient condition for $\eta \to 1$ (for good load balance) in the corollary, $q = \omega(\log^d p)$, is not too strong. (We conjecture that this condition is also necessary.)

6 Conclusions

We have investigated the balance of load between processors, assuming that:

(i) A given problem consists of many subproblems varying in size (description length).

(ii) The size of each subproblem is random and independently identically distributed according to a Poisson distribution.

(iii) Each subproblem is independently solved by the same polynomial-time algorithm.

(iv) Each processor is assigned the same number of subproblems.

(v) A larger problem consists of a larger number of subproblems, while the average size of a subproblem is constant regardless of the size of the entire problem.

When the number of processors p increases, if we assign an $\omega(\log^d p)$ number of subproblems to each processor, we can almost balance the load between processors, where d is the order of the time complexity of the algorithm being used for solving each subproblem. Hence, a problem that consists of an $\omega(p \log^d p)$ number of subproblems can be efficiently solved in parallel with p processors.

Thus we conclude that a static load balancing scheme, assigning the same number of subproblems to each processor, is fairly scalable.

References

[1] W. Dally, "Network and Processor Architecture for Message-Driven Computers," in: R. Suaya and G. Birtwictle, ed., *VLSI and Parallel Computation*, Morgan Kaufmann, (1990).

[2] G.C. Fox *et al.*, "Solving Problems on Concurrent Processors," vol.I, Prentice-Hall, (1988).

[3] M. Furuichi, K. Taki, and N. Ichiyoshi, "A Multi-Level Load Balancing Scheme for OR-Parallel Exhaustive Search Programs on the Multi-PSI," *Proc. of the 2nd ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming*, pp.50–59 (1990).

[4] N. Ichiyoshi and K. Kimura, "Asymptotic Load Balance of Distributed Hash Tables," *ICOT Technical Report*, to appear, (1991).

[5] K. Ito (ed.), *Encyclopedic Dictionary of Mathematics*, MIT Press, 2nd ed., (1987).

[6] T. Kamac, U. Krengel and G.L.O'Brien, "Stochastic Inequalities on Partially Ordered Spaces," *The Annals of Probability*, Vol.5, No.6, pp.899-912, (1977).

[7] K. Kimura and N. Ichiyoshi, "Probabilistic Analysis of the Optimal Efficiency of the Multi-Level Dynamic Load Balancing Scheme," *Proc. of the 6th Distributed Memory Computing Conference*, (1991).

[8] V.F. Kolchin, B.A. Sevastyanov and V.P. Chistyakov, *Random Allocations*, Winston & Sons, Washington, (1978).

[9] V. Kumar and A. Gupta, "Analysis of Scalability of Parallel Algorithms and Architectures: A Survey," *Proc. of International Conference on Supercomputing*, (1991).

[10] V. Kumar and V. N. Rao, "Load Balancing on the Hypercube Architecture," *Proc. of the 1989 Conference on Hypercubes, Concurrent Computers and Applications*, (1989).

[11] J.S. Vitter and P. Flajolet, "Average-Case Analysis of Algorithms and Data Structures," in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol.A*, North-Holland, Amsterdam, (1990).