Theoretical Analysis of the Heterogeneous Dynamic Load Balancing Problem Using a Hydro-Dynamic Approach

Chi-Chung Hui* and Samuel T. Chanson*

Technical Report HKUST-CS96-01
March 5, 1996

*Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

†This research was supported in part by the Hong Kong Research Grant Council under grant number HKUST 606/94E.
Abstract

This paper presents a hydro-dynamic framework to solving the dynamic load balancing problem on a network of heterogeneous computers. In this approach, each processor is viewed as a liquid cylinder where the cross-sectional area corresponds to the capacity of the processor, the communication links are modeled as liquid channels between the cylinders, the workload is represented as liquid, and the load balancing algorithm describes the flow of the liquid. It is proved that all algorithms under this framework converge geometrically to the state of equilibrium, in which the heights of the liquid columns are the same in all the cylinders. In this way, each processor obtains an amount of workload proportional to its capacity. The parameters that affect the convergence rate of the algorithms are also identified and discussed.
1 Introduction

It is useful to explore remote computing power in local area networks (LANs) as processors get more and more powerful and the availability of high speed networks such as fast Ethernet, FDDI and ATM has reduced the cost of interprocessor communication. In the LAN environment, a significant portion of the workstations is left idle even during busy periods [1, 2]. The aggregate free CPU power in a large scale LAN may be comparable to that of a supercomputer, and it is possible to improve the overall system throughput by balancing the workload among the workstations.

Because of this potential gain, load balancing has been investigated intensively in recent years [3, 4, 5, 6, 7]. Load balancing techniques can be classified as either static or dynamic. Static load balancing requires complete global information on the computing system and workload characteristics. On the other hand, dynamic load balancing makes little assumption about the system or workload, and the scheduling decision is based on measured loading information at the time scheduling is performed. In the LAN environment, workload characteristics and workstation utilizations are generally difficult to predict. Furthermore, the workstations are usually not homogeneous (i.e., run at different speeds). It is therefore more suitable to employ heterogeneous dynamic load balancing strategies in practice.

The nearest-neighbor approach is a dynamic load balancing technique that allows the workstations to communicate and migrate tasks with their immediate neighbors only [7]. Each workstation balances the workload with its neighbors in the hope that after a number of iterations the whole system will approach the balanced state. Since it is not necessary to
have a global coordinator, nearest-neighbor algorithms are inherently local, fault tolerant and scalable. Hence this approach is a natural choice for load balancing in a highly dynamic environment.

We consider the following heterogeneous dynamic load balancing problem: The computing system is modeled as an undirected graph $G = (N, E)$ where $N$ represents the set of workstations, and $E$ represents the topology of the communication network. Each workstation is associated with two real variables, capacity and load, which reflect its processing speed and the workload currently running on it respectively. A general hydrodynamic framework is proposed to redistribute the workload among the workstations such that each workstation obtains its share of the workload proportional to its capacity. In particular, we model $G$ as a hydrodynamic system. A potential energy function is defined for $G$ in which the minimum value corresponds to the state of equilibrium. The load balancing algorithms define the flow of liquid to achieve equilibrium. In this paper, we show how the hydrodynamic approach can be analyzed mathematically, prove that all algorithms under this framework converge geometrically to the optimal state, and discuss the parameters that affect the convergence rate.

The rest of the paper is organized as follow: The related work is presented in section 2. Section 3 defines the load balancing problem formally. Section 4 describes the hydrodynamic approach and explains how it is used to solve the problem. The framework assumptions and convergence properties are presented in sections 5 and 6 respectively. The parameters that affect the convergence rate are studied in section 7. Finally, section 8 concludes the paper.
2 Related Work

Early work on the nearest-neighbor approach investigated the stability of the algorithms [8, 9, 10], whereas more recent work has concentrated on proving the convergence property and the convergence rate of the algorithms. Some of the major approaches include the diffusion method [11], the dimension exchange method [11, 12] and the gradient based method [13, 14]. Comparisons of the different algorithms have also been reported [15, 4, 6, 16].

In the diffusion method, each processor simultaneously sends workload to its neighbors with lighter workload and receives workload from its neighbors with heavier workload. Under the synchronous assumption, the diffusion method has been proved to converge in polynomial time for any initial workload distribution given the quiescent assumption that no new workload is generated and no existing workload is completed during execution of the algorithm [17, 11]. Without the quiescent assumption, it is still possible to prove that the variance of the unbalanced workload is bounded [11, 18]. For regular network topologies such as mesh, torus and n-D hypercube, optimal parameters that maximize the convergence rate have been derived [19]. The convergence of the asynchronous version of the diffusion method has also been proved [20, 21].

A processor in the dimension exchange method balances the workload with its neighbors one at a time. It has been proved that on a hypercube, the entire system is balanced when every processor has exchanged workload with all its neighbors once [11]. The performance of the dimension exchange method on hypercube, the shuffle-exchange, the cube-connected cycles and the butterfly has been compared [22]. By applying the edge
coloring technique to map the edges into dimensions, the dimension exchange method can also be applied to arbitrary graphs [12]. More recently, this method has been generalized and optimal parameters derived to maximize the convergence rate on $n$-D mesh, torus and $k$-ary $n$-cubes [23, 24].

The processors in the gradient based method maintain gradient maps which describe the workload variations in the system. Tasks are moved toward the processors with the steepest gradient. In the gradient model (GM) method, a pressure surface that represents the propagated pressure of the workload is defined [13]. In the contracting within a neighborhood (CWN) method, the workload index is used directly and the tasks are sent to the processor with the smallest index [14]. Various techniques such as randomized methods and simulated annealing have been used to provide solutions to the load balancing problem [7].

Most of the work mentioned above assumes the processors to be homogeneous. By comparison, the hydro-dynamic framework defines the notion of capacity for the processors. In this way, it is more convenient to control the amount of workload allocated to heterogeneous processors. The proposed framework is general in that it is applicable to systems with arbitrary topologies, and can work with asynchronous networks as well as synchronous networks. It may also be viewed as a natural extension to some homogeneous techniques such as the diffusion method and the generalized dimension exchange method. In this paper, we prove that all algorithms under this framework converge geometrically for all workload configurations. Moreover, the parameters that affect the rate of convergence have been identified and their effect studied. This provides valuable insight into the properties and applicability of the proposed algorithms.
3 Problem Formulation

We consider an environment consisting of a set of autonomous processors connected by a communication network. The system is modeled as an undirected graph \( G = (N, E) \), where the node set \( N \) represents the set of asynchronous heterogeneous processors, and the edge set \( E \) describes the connection pattern among the processors. Each processor may execute multiple processes in a multitasking manner, and is equipped with software/hardware facilities such that non-blocking message delivery is possible. Each node \( n_i \) is associated with a capacity \( c_i > 0 \) which specifies the relative capacities of the processors. A second attribute load \( l_i \geq 0 \) reflects the amount of workload currently running on \( n_i \). Both \( c_i \) and \( l_i \) are real numbers, and it is assumed the workload is infinitely divisible. Since the values of the variables vary with time, the variables are usually expressed as \( c_i[t] \) and \( l_i[t] \). For simplicity, if a variable is not qualified explicitly, time \( t \) is assumed. If \( c_i = x \cdot c_j \), then \( n_i \) and \( n_j \) are said to have achieved fairness if \( l_i = x \cdot l_j \) (i.e., the loads acquired by the processors are proportional to their capacities). If there exists a communication link \((n_i, n_j) \in E\), the nodes \( n_i \) and \( n_j \) can exchanging workload information and move workload between them. The links are assumed to be FIFO channels with bounded delay times. \( G \) is disturbed if at least one of the following conditions is satisfied: (i) \( N \) is changed, (ii) \( c_i \) is changed for some \( n_i \in N \), or (iii) the total workload in the system (i.e., \( \sum_j l_j \)) is changed. In practice, \( G \) will be disturbed from time to time. However, the load balancing algorithm should quickly adapt to perturbations and reach equilibrium if \( G \) is not disturbed for a sufficiently long period. Modifying the edge set \( E \) should not affect

\(^*\)Although the speed of processor is fixed, \( c_i[t] \) may vary with time. For example, if the processor \( n_i \) is reclaimed by the workstation owner, \( c_i[t] \) may be adjusted to a smaller value.
the convergence of the algorithm as long as \( G \) is connected, though it may affect the rate of convergence. The problem to be solved is summarized in the following:

**Definition 1**: [Heterogeneous dynamic load balancing problem] Given a network of computers \( G = (N, E) \) and any workload, an algorithm is to be found to redistribute the workload among the processors such that if \( G \) is not disturbed in some finite time \( A \), the workload allocated to each node \( n_i \) is fair, that is,

\[
l_i[t + A] = \sum_j c_i \cdot \sum_j l_j
\]

for all \( n_i \in N \). When this happens, the system \( G \) is said to have achieved *global fairness*.

In this paper, we do not consider task migration during execution. The term *migration* used in this paper means *load index balancing* (i.e., notification of load transfer) which does not necessarily result in immediate transfer of workload. The actual workload movement may take place later and combine several load transfer notifications to reduce the frequency of data transfer.

An example system \( G_1 \) is shown in Figure 1. The 2-tuple \((c_i, l_i)\) associates with each \( n_i \); describes the initial capacity and workload of the node. This example is used throughout the paper to illustrate the proposed algorithm.
4 The Hydro-Dynamic Approach

The *hydro-dynamic* approach forms the basic framework of our solution to the heterogeneous dynamic load balancing problem. The idea is shown in Figure 2, where $G_1$ in Figure 1 is represented as a system of globally connected liquid cylinders. Each node $n_i \in N$ is associated with a liquid cylinder; the size of the cross-sectional area corre-
sponds to \( c_i \), and the volume of the liquid represents the workload currently allocated to \( n_i \) (i.e., \( l_i \)). There is an infinitely thin liquid channel joining the bottoms of two liquid cylinders if there is an edge between the two corresponding nodes in \( G \). Our proposed solution models the flow of liquid among the cylinders. It is intuitive that global fairness is achieved when the heights of the liquid columns in the cylinders are equal. It is also obvious that after global fairness has been achieved there is no liquid flow among the cylinders and therefore the system is stable. In the following subsections we show how this system can be analyzed mathematically.

### 4.1 The Concept of Potential Energy

The core of the analysis is to derive a function of global potential energy \( GPE \) to measure the level of fairness among the nodes in \( G \). We first define the concepts of height and mass:

**Definition 2**: Given a node \( n_i \in N \), the height of \( n_i \) is defined as

\[
h_i = \frac{l_i}{c_i}. \quad \Box
\]

**Definition 3**: The mass of \( n_i \) between the height range \( (h_j, h_k) \) is denoted by \( m_{h_j}^{h_k}(n_i) \) where

\[
m_{h_j}^{h_k}(n_i) = \int_{h_j}^{h_k} c_i \, dh = c_i \cdot (h_k - h_j). \quad \Box
\]
Definition 4: The potential energy of $n_i$ between the height range $[h_j, h_k]$ is defined as

$$PE_{h_j}^{h_k}(n_i) = \int_{h_j}^{h_k} c_i \cdot h \, dh$$

$$= c_i \left[ \frac{h^2}{2} \right]_{h_j}^{h_k}$$

$$= m_{h_j}^{h_k}(n_i) \cdot (h_k - h_j)/2,$$

and the potential energy of the liquid column in $n_i$ with a height of $h_i$ is simply

$$PE(n_i) = PE_{0}^h(n_i) = \frac{c_i h_i^2}{2}. \quad \square$$

Finally, the global potential energy of the system is defined as the sum of potential energies of all the nodes:

Definition 5: The global potential energy of $G = (N, E)$ is defined as

$$GPE(G) = \sum_{n_i \in N} PE(n_i). \quad \square$$

4.2 Global Potential Energy and Global Fairness

In this section we prove that $GPE$ is minimized only at the state of global fairness. The following lemma states that for any two liquid segments $\mathcal{L}_a$ and $\mathcal{L}_b$ with the same volume, $\mathcal{L}_a$ has larger potential energy if the bottom level of $\mathcal{L}_a$ is equal to or higher than the top level of $\mathcal{L}_b$: 
Lemma 1: For any two nodes \( n_i, n_j \in N \), if \( m^{a_2}(n_i) = m^{b_2}(n_j) \), \( a_2 > a_1 \geq b_2 > b_1 \), and \( \Delta PE = PE^{b_2}_{b_i}(n_j) - PE^{a_2}_{a_i}(n_i) \), then

\[
\Delta PE < - \frac{c_i \cdot (a_2 - a_1)^2}{2} < 0 \quad \text{and} \quad \Delta PE < - \frac{c_j \cdot (b_2 - b_1)^2}{2} < 0.
\]

Proof: For the first case,

\[
\Delta PE = PE^{b_2}_{b_i}(n_j) - PE^{a_2}_{a_i}(n_i)
= m^{a_2}(n_i) \cdot [(b_1 + b_2) - (a_1 + a_2)]/2
< m^{a_2}(n_i) \cdot (2a_1 - a_1 - a_2)/2
= - \frac{c_i \cdot (a_2 - a_1)^2}{2} < 0.
\]

The second case can be proved in a similar way. □

Therefore, the flow of a mass (or volume) of liquid from a higher position to a lower position reduces the potential energy of the liquid. Given the above lemma, it is easy to prove that the global potential energy of \( G \) is minimized when the relative workload allocated to each node is the same (i.e., the system is in the state of global fairness).

Theorem 1: Given \( G = (N, E) \), the function \( GPE(G) \) is minimized if and only if the state of global fairness is achieved, i.e.,

\[
h_i = h_{\text{opt}} \quad \forall i
\]

where \( h_{\text{opt}} = \frac{\sum_i b_i}{\sum_j c_j} \).
Proof: To prove the theorem, we compare the GPE of the globally fair system with those of all the other configurations where the heights of the nodes are not equal. For the unfair case, we partition the node set $N$ into three subsets $N_{\text{lower}} = \{n_i \mid h_i < h_{opt}\}$, $N_{\text{equal}} = \{n_i \mid h_i = h_{opt}\}$ and $N_{\text{higher}} = \{n_i \mid h_i > h_{opt}\}$. It is easy to see that $N = N_{\text{lower}} \cup N_{\text{equal}} \cup N_{\text{higher}}$. Since the total workload $\sum_i l_i$ is the same in the two cases, the equality

$$\sum_{n_i \in N_{\text{lower}}} (h_{opt} - h_i) = \sum_{n_i \in N_{\text{higher}}} (h_i - h_{opt})$$

holds. This situation is shown graphically in Figure 3. Consider moving the workload above $h_{opt}$ in $N_{\text{higher}}$ to the “holes” in $N_{\text{lower}}$. By Lemma 1, the change in potential energy for every movement must be negative since the workload flows from a position above $h_{opt}$ to a position below $h_{opt}$. Hence, the GPE of $G$ at the state of global fairness must be strictly smaller than those of all other configurations. $\square$

![Figure 3: Proof of minimum GPE.](image)
5 Algorithmic Assumptions

The convergence property of the proposed framework relies on three assumptions. In this section the assumptions as well as their effects are discussed. The first assumption is concerned with the direction of liquid flow.

Assumption 1: Liquid must flow from a higher position to a lower position. □

Under the hydro-dynamic framework, this assumption corresponds to the physical law of liquid movement and reduces fluctuation of liquid flow among the cylinders. From the mathematical point of view, it ensures the \( GPE \) to be a monotonic decreasing function (the proof is given in Lemma 1).

The second assumption defines the allowable workload movement. It states that for each decision on workload migration, the change in global potential energy must be negative and the reduction must be greater than some nontrivial amount.

Assumption 2: When a node \( n_i \) balances its workload with some of its neighbors in \( N' \subseteq \text{adj}(n_i) \), where \( \text{adj}(n_i) \) is the set of neighbors of \( n_i \), the resultant reduction in \( GPE \) must be greater than or equal to the baseline case in which \( n_i \) transfers an amount of workload equal to \( \frac{c_{ij}}{c_{ii}+c_{jj}}(h_i - h_j) \) to node \( n_j \), where \( n_j \) is the element in \( N' \) with the lowest height (i.e., \( h_j = \min_{k \in N'} \{ h_k \} \)). The parameter \( \gamma \) is called the balancing factor and is in the range \((0,1]\). □

The balancing factor \( \gamma \) controls the amount of workload flow among the nodes. It must be greater than 0 to ensure there is workload movement. If \( \gamma = 1 \) then for the baseline case the heights of \( n_i \) and \( n_j \) are equal after they have balanced their workload, and this
represents the upper bound on $GPE$ reduction when $n_i$ has only one neighbor ($|N'| = 1$).

However, the reduction of $GPE$ may exceed this value when $|N'| > 1$. Assumption 2 is only concerned with the amount of $GPE$ reduction, and not on the liquid flow pattern. $n_i$ may transfer its excessive workload to its neighbors in an arbitrary manner, provided that the $GPE$ reduction is not smaller than that in the baseline case. Moreover, there is no restriction on the number of neighbors with which $n_i$ may balance its workload. As a result, the proposed framework is much more general than the existing work.

The last assumption is concerned with the boundedness of communication delay.

**Assumption 3**: Every adjacent node pair in $E$ is involved in load balancing activity which is completed within a finite interval $B$ during the execution of the load balancing algorithm. □

Assumption 3 implies that the communication channels in $G$ have bounded delay times for both asynchronous and synchronous networks. It is required that during the period $B$ every node pair in $E$ must have communicated at least once in order to guarantee geometric convergence.

### 6 Convergence Properties

In this section we prove that all the computing systems under the hydro-dynamic framework converge geometrically to the state of global fairness in finite time. The major observation that led to the result is that after the nodes in $G$ have balanced the workload in period $B$ and there is no disturbance during this period (i.e., $[t, t + B]$), there is a significant reduction in the $GPE$ of the system. The proof is divided into three parts:
First we prove in section 6.1 that there exists a lower bound on the reduction of $GPE$ in period $B$. In section 6.2 we identify the maximum distance between $GPE[t]$ and the optimal (i.e., smallest) value. By comparing the two values, it is proved in section 6.3 that the $GPE$ converges to the optimal value geometrically for all algorithms under this framework.

### 6.1 Lower Bound on GPE Reduction

This section proves that the $GPE$ reduction in period $[t, t + B]$ has a lower bound and is proportional to $(h_{\text{max}}[t] - h_{\text{min}}[t])^2$, where $h_{\text{min}}[t]$ and $h_{\text{max}}[t]$ are the minimum and maximum heights of the nodes in $G$ at time $t$ respectively. The first lemma relates the maximum difference in the heights of the adjacent node pairs in $E$ with $h_{\text{min}}$ and $h_{\text{max}}$.

**Lemma 2:** For any $G = (N, E)$, if $h_{\text{max}} = \max_i h_i$ and $h_{\text{min}} = \min_i h_i$ then

$$\max_{(n_i, n_j) \in E} \{h_i - h_j\} \geq \frac{h_{\text{max}} - h_{\text{min}}}{\text{dia}(G)}$$

where $\text{dia}(G)$ is the diameter of $G$.

**Proof:** Assume the contrary, that is, $\max_{(n_i, n_j) \in E} \{h_i - h_j\} < \frac{h_{\text{max}} - h_{\text{min}}}{\text{dia}(G)}$. Without loss of generality, let $n_u$ and $n_v$ be the nodes in $N$ such that $h_u = h_{\text{max}}$ and $h_v = h_{\text{min}}$. Furthermore, let the shortest path connecting $n_u$ and $n_v$ be $n_{x_0} \sim n_{x_p}$ with length $p$, where $n_u = n_{x_0}$ and $n_v = n_{x_p}$. Obviously $p \leq \text{dia}(G)$. The difference in height between
$n_u$ and $n_v$ is

\[
\begin{align*}
    h_u - h_v &= h_{x_0} - h_{x_p} \\
    &\leq \sum_{y=0}^{p-1} |h_{x_y} - h_{x_{y+1}}| \\
    &< p \cdot \frac{h_{\text{max}} - h_{\text{min}}}{\text{dia}(G)} \\
    &\leq h_{\text{max}} - h_{\text{min}}
\end{align*}
\]

which is a contradiction. □

In fact, after $n_u$ and $n_v$ have finished balancing their workload, the reduction of $GPE$ is greater than or equal to the lower bound. To prove this property, we first prove that for any adjacent node pair $(n_i, n_j) \in E$ with $h_i \geq h_j$, the accumulated flow of liquid for at least one of the nodes after one balancing step is proportional to $|h_i - h_j|$. The load balancing process is shown in Figure 4, where $n_i$ sends the workload transfer notification to $n_j$ at time $t'$, and $n_j$ receives the notification and determines the workload to be accepted at time $t''$. Of course, both nodes may have already exchanged workload with other neighbors in the period $[t, t'']$. In Figure 4, $n_i$ ($n_j$) has exchanged a total of $\sum_k x_k$ ($\sum_k y_k$) units of workload with its neighbors in the period $[t, t']$. After $n_i$ sent the workload
transfer notification to $n_j$ at time $t'$, $n_j$ could have exchanged an additional $\sum_k z_k$ units of workload with its neighbors in the period $[t', t'']$. The activities of $n_i$ after $t'$ are irrelevant and are omitted. It is easy to see that $h_i[t'] = h_i[t] + \sum_{c_i} x_k$, $h_j[t'] = h_j[t] + \sum_{c_j} y_k$, and $h_j[t''] = h_j[t'] + \sum_{c_j} z_k$. Moreover, there is no assumption on the signs of $x_k$, $y_k$ and $z_k$ so that $h_i[t']$ may be greater than $h_i[t]$ (as well as lower than $h_i[t]$, which is shown in the figure).

Since the communication channels have nonzero transfer time, $n_i$’s information on the height of $n_j$ at time $t'$, denoted as $h_j^i[t']$, may be outdated. However, this should not affect the load balancing process. In terms of the model shown in Figure 4, part of the workload $y_k$ is shifted to $z_k$. This will not change the result since $n_j$ considers the sum of $\sum_k y_k$ and $\sum_k z_k$ in migration decision at time $t''$.

If we denote the accumulated flow of workload into and out of $n_i$ from time $t_1$ to time $t_2$ as $I_i(t_1, t_2)$ and $O_i(t_1, t_2)$ respectively, the following lemma can be proved:

**Lemma 3:** Given any $t', t'' \in [t, t + B]$ with $t' \leq t''$ and $(n_i, n_j) \in E$ such that $h_i[t] \geq h_j[t]$. At time $t''$ at least one of the following inequalities is true: (i) $O_i(t, t'') \leq -\gamma_{c_i+c_j}(h_i[t] - h_j[t])$, or (ii) $I_j(t, t'') \geq \gamma_{c_i+c_j}(h_i[t] - h_j[t])$.

**Proof:** The proof is given in the Appendix. \(\square\)

The next step is to relate the amount of liquid flow to GPE reduction. First, we define a flow pattern $f$ with respect to $n_i$ to be the collection of workload transfers between $n_i$ and the other nodes in which every node $n_j \neq n_i$ transfers workload equivalent to height

---

1 For the purpose of the proof, the convention adopted for both $I_j(t_1, t_2)$ and $O_i(t_1, t_2)$ is that a negative value means flowing out and a positive value means flowing in. The same convention also applies to $x_k$, $y_k$ and $z_k$ defined in Figure 4.
$f_j$ into (or out of) $n_j$. The height $f_j$ is measured with respect to $n_i$, and the volume of workload flow is equal to $f_j \cdot c_i$. An example flow pattern is shown in Figure 5, where in a system with 4 nodes $n_1$ received workload from $n_2$, $n_3$ and $n_4$. It is important to note that the liquid level (before and after the transfer) at $n_j$ must be higher than or equal to $H_j$ (by Assumption 1). Therefore, we can apply Lemma 1 to obtain the corresponding lower bound on $GPE$ reduction.

The next two lemmas provide a lower bound on $GPE$ reduction given the flow amount. In particular, if $W$ units of workload flows into (or out of) any node, Lemma 4 identifies the scenario which minimizes the $GPE$ reduction.

**Lemma 4**: If $W$ units of workload flows into (or out of) $n_i$, then the reduction of $GPE$ is minimized for the following flow pattern $\zeta$: Each node $n_j \neq n_i$ migrates workload with height $\zeta_j = \frac{\Delta h_i}{|N|-1}$ into (or out of) $n_i$ where $\Delta h_i = W/c_i$.

**Proof**: The proof is given in the Appendix. □

---

1. In this paper we use “into (or out of)” to mean either all nodes $n_j \neq n_i$ transfer workload with height $f_j$ to $n_i$ or they all receive workload with height $f_j$ from $n_i$.

2. Notice that the workload may be transferred through many nodes between $n_i$ and $n_j$. 

---

Figure 5: Liquid flow pattern.
Lemma 5: If $W$ units of workload flows into (or out of) $n_i$, then the change in $GPE$, denoted as $\Delta GPE$, is strictly less than $-\frac{W^2}{2c_i(|N|-1)}$.

Proof: By Lemma 4 it is known that the reduction in $GPE$ is minimized when every neighbor $n_j$ of $n_i$ migrates workload with height $\frac{W}{c_i(|N|-1)}$ into (or out of) $n_i$. Therefore,

\[
\Delta GPE = \sum_{n_j \neq n_i} \Delta PE(n_j) < \sum_{n_j \neq n_i} -c_i \left[ \frac{W}{c_i(|N|-1)} \right]^2 / 2 \quad \text{by Lemma 1 and Assumption 1}
\leq -\frac{W^2}{2c_i(|N|-1)}. \quad \square
\]

Given the above lemmas, we can compute a lower bound on $GPE$ reduction in period $B$ by considering the workload flow between the adjacent node pair with the greatest difference in height (i.e., $n_u$ and $n_v$). Notice that when $n_u$ balances the workload with a group of neighboring nodes $N' \subseteq adj(n_u)$ at time $t'$ and $n_v \in N'$, $h_v[t']$ may not be the smallest in $N'$ if $|N'| > 1$. Assume $n_w \in N'$ is the node with the lowest height at time $t'$, we can combine $n_v$ and $n_w$ into a composite node $n_{v,w}$ where $c_{v,w} = \min\{c_v, c_w\}$, $adj(n_{v,w}) = adj(n_v) \cup adj(n_w)$, and $h_{v,w}[t] = \min\{h_v[t], h_w[t]\} = h_v[t]$. This is shown in Figure 6.

Therefore, we can apply Lemma 3 to $n_u$ and $n_{v,w}$ to obtain the result that after time $t''$, at least one of the following inequalities holds: (i) $O_u(t, t'') \leq -\gamma \frac{c_u c_{v,w}}{c_u + c_{v,w}} (h_u[t] - h_v[t])$, or (ii) $T_{v,w}(t, t'') \geq \gamma \frac{c_u c_{v,w}}{c_u + c_{v,w}} (h_u[t] - h_v[t])$. With this result, the following theorem can be proved:
Theorem 2: After period $B$, the following inequality holds:

$$\Delta GPE[t + B] < -\frac{\gamma^2}{8(|N| - 1) \text{dia}(G)^2} \cdot \frac{c_{\text{min}}^2}{c_{\text{max}}} \cdot (h_{\text{max}} - h_{\text{min}})^2.$$ 

where $c_{\text{min}} = \min_i \{c_i\}$ and $c_{\text{max}} = \max_i \{c_i\}$.

Proof: According to Assumption 3, after period $B$ (i.e., at time $t + B$), every adjacent node pair in $E$ has interacted to balance the workload. Consider the node pair $(n_u, n_{v,w})$.

From Lemma 2 it is known that $h_u - h_{v,w} \geq \frac{h_{\text{max}} - h_{\text{min}}}{\text{dia}(G)}$. From Lemma 3 the workload migration of at least one of $n_v$ and $n_{v,w}$ is greater than or equal to $\gamma \frac{c_{\text{min}}}{c_{\text{min}} c_{\text{max}}}(h_u - h_v)$, and $\frac{c_{\text{min}}}{c_{\text{min}} c_{\text{max}}}$ is no less than $\frac{\min\{c_u, c_v, c_w\}}{2}$ which has a lower bound of $\frac{c_{\text{min}}}{2}$. Therefore, by Assumption 1 and Lemma 5 the change in $GPE$ is bounded by:

$$\Delta GPE[t + B] < -\frac{1}{2c_{\text{max}}(|N| - 1)} \left[\gamma \frac{c_{\text{min}}}{2}(h_u - h_v)\right]^2$$

$$< -\frac{1}{2c_{\text{max}}(|N| - 1)} \left(\gamma \frac{c_{\text{min}} h_{\text{max}} - h_{\text{min}}}{\text{dia}(G)}\right)^2$$

$$= -\frac{\gamma^2}{8(|N| - 1) \text{dia}(G)^2} \cdot \frac{c_{\text{min}}^2}{c_{\text{max}}} \cdot (h_{\text{max}} - h_{\text{min}})^2. \qed$$
6.2 Maximal Distance to the Optimal State

This section proves that the difference between $GPE$ and $GPE_{\text{min}}$ has an upper bound proportional to $(h_{\text{max}}[t] - h_{\text{min}}[t])^2$. The following lemma describes the configuration of $G$ that maximizes $GPE$ given $h_{\text{max}}$ and $h_{\text{min}}$:

**Lemma 6**: Given $G$ with any $h_{\text{max}}$ and $h_{\text{min}}$, the $GPE$ is maximized for the configuration shown in Figure 7 where every node has a height equal to either $h_{\text{max}}$ or $h_{\text{min}}$. The total cross-sectional areas (i.e., capacity) of the nodes with heights $h_{\text{max}}$ and $h_{\text{min}}$ are equal to $c_a$ and $c_b$ respectively.

**Proof**: The proof is obvious since this configuration can be converted to all other configurations by flowing the liquid downward, and by Lemma 1 all the moves result in negative changes in $GPE$. \qed

The maximum difference in $GPE$ for any configuration given the maximum and minimum heights can now be calculated.

---

$GPE_{\text{min}}$ is the value of $GPE$ at global fairness. From Theorem 1, the global potential energy is minimized at global fairness.

Notice that we may not be able to divide the cylinders into two groups whose capacities sum to $c_a$ and $c_b$ exactly. However, it is possible to create the proposed configuration theoretically by dividing the cylinders into infinitely thin cylinders and recombining them.
Theorem 3: Given \( G \) with any \( h_{\text{max}} \) and \( h_{\text{min}} \),

\[
GPE - GPE_{\text{min}} \leq \frac{\sum i c_i}{4} (h_{\text{max}} - h_{\text{min}})^2.
\]

Proof: Let the \( GPE \) of the configuration specified in Lemma 6 be \( GPE_{\text{max}} \). Since the volume of liquid above \( h_{\text{opt}} \) is equal to that below \( h_{\text{opt}} \), we have \( c_a(h_{\text{max}} - h_{\text{opt}}) = c_b(h_{\text{opt}} - h_{\text{min}}) \), and so

\[
h_{\text{opt}} = \frac{c_a h_{\text{max}} + c_b h_{\text{min}}}{c_a + c_b}.
\]

On the other hand, \( \frac{c_a c_b}{c_a + c_b} \) cannot exceed \( \frac{\max\{c_a, c_b\}}{2} \) which has an upper bound of \( \frac{\sum c_i}{2} \). Therefore,

\[
GPE - GPE_{\text{min}} \leq GPE_{\text{max}} - GPE_{\text{min}}
\]

\[
= \left( \frac{c_a}{2} h_{\text{max}}^2 + \frac{c_b}{2} h_{\text{min}}^2 \right) - \left( \frac{c_a + c_b}{2} h_{\text{opt}}^2 \right)
\]

\[
= \frac{c_a c_b}{2(c_a + c_b)} (h_{\text{max}} - h_{\text{min}})^2
\]

\[
\leq \frac{\sum c_i}{4} (h_{\text{max}} - h_{\text{min}})^2. \quad \Box
\]

6.3 Geometric Convergence

Based on the theorems derived in the previous sections, it is possible to prove that the \( GPE \) converges to the optimal state geometrically.

Lemma 7:

\[
GPE[t + B] < \eta GPE[t] + (1 - \eta) GPE_{\text{min}}
\]
for \( t \geq 0 \) where
\[
\eta = 1 - \frac{\gamma^2}{2(|N| - 1) \text{dia}(G)^2} \cdot \frac{c_{\text{min}}^2}{c_{\text{max}} \sum c_i}
\] (2)
and \( 1 > \eta > 0 \).

**Proof:** From Theorems 2 and 3 it is known that
\[
\Delta \text{GPE}[t + B] < -\alpha(h_{\text{max}}[t] - h_{\text{min}}[t])^2
\] (3)
and
\[
\text{GPE}[t] - \text{GPE}_{\text{min}} \leq \beta(h_{\text{max}}[t] - h_{\text{min}}[t])^2
\] (4)
where
\[
\alpha = \frac{\gamma^2}{8(|N| - 1) \text{dia}(G)^2} \cdot \frac{c_{\text{min}}^2}{c_{\text{max}}} > 0
\]
and
\[
\beta = \sum \frac{c_i}{4} > 0.
\]

By combining Equations (3) and (4), we obtain
\[
\frac{\text{GPE}[t] - \text{GPE}_{\text{min}}}{\beta} \leq (h_{\text{max}}[t] - h_{\text{min}}[t])^2 < -\frac{\Delta \text{GPE}[t + B]}{\alpha}
\]
\[
\Leftrightarrow \Delta \text{GPE}[t + B] < \frac{\alpha}{\beta}(\text{GPE}[t] - \text{GPE}_{\text{min}}).
\]

Therefore,
\[
\text{GPE}[t + B] = \text{GPE}[t] + \Delta \text{GPE}[t + B]
\]
\[
< \text{GPE}[t] - \frac{\alpha}{\beta}(\text{GPE}[t] - \text{GPE}_{\text{min}})
\]
\[
= \eta \text{GPE}[t] + (1 - \eta) \text{GPE}_{\text{min}}
\]
where $\eta = 1 - \frac{2}{\beta}$. □

Using the technique of induction, we can prove that the proposed framework guarantees geometrical convergence given any configuration.

**Theorem 4 :**

\[
GPE[B\tau] < \eta^\tau GPE[0] + (1 - \eta^\tau)GPE_{\min} \tag{5}
\]

for $\tau \geq 1$.

**Proof:** The case for $\tau = 1$ is true which is the result of Lemma 7 when $t = 0$. Assume the assertion is true for $\tau = k$, then

\[
GPE[B(\tau + 1)] = GPE[B\tau + B] < \eta GPE[B\tau] - (1 - \eta)GPE_{\min} \quad \text{by Lemma 7}
\]

\[
< \eta (\eta^\tau GPE[0] + (1 - \eta^\tau)GPE_{\min}) + (1 - \eta)GPE_{\min}
\]

\[
= \eta^{\tau+1} GPE[0] + (1 - \eta^{\tau+1})GPE_{\min}.
\]

Hence, the case for $\tau = k + 1$ is also true and the theorem is proved. □

Therefore, as the value of $\tau$ increases, $\eta^\tau$ and $(1 - \eta^\tau)$ converge to 0 and 1 respectively, and so $GPE[B\tau]$ tends to $GPE_{\min}$. Given any finite error requirement, it is possible to compute the value of $\tau$ by Equation (5) such that after finite period $A = B\tau$, the requirement is satisfied.
7 Effects of Parameters on Convergence Rate

It follows from Theorem 4 that all algorithms under the hydro-dynamic framework converge geometrically to the optimal state. The variable \( \eta \) defined in Equation (2) indicates the rate of convergence. According to Equation (5), the smaller the value of \( \eta \), the faster the value of \( GPE[t] \) converges to \( GPE_{\min} \). From Equation (2), there are 4 factors affecting the value of \( \eta \):

- The balancing factor \( \gamma \),
- The capacities of the processors (which affect the value of \( \frac{c_{\min}^2}{c_{\max} \sum_i c_i} \)),
- The number of processors (which affects \( |N| \) and \( \sum_i c_i \)), and
- The diameter of the network \( dia(G) \).

Experiments isolating the effect of the factors were designed. The results of the experiments show that these factors influence the performance of convergence as specified in Equation (2).

7.1 Simulation Environment

A discrete-event simulator which implements the baseline algorithm described in Section 5 has been constructed. In particular, the simulator assumes \( G \) to be an asynchronous network. The smallest divisible time unit is set at 1 ms. The time to start up the algorithm in each node is taken to be \( U(0, 5) \) ms, where \( U(x, y) \) returns a uniformly distributed number in \([x, y] \). The time to transfer a message between adjacent nodes is set at \( U(10, 30) \) ms, and each node spends \( U(10, 20) \) ms in processing a message. Each
node $n_i$ starts the load balancing procedure when its neighbor with the lowest height, says $n_j$, satisfies the condition $\frac{h_i - h_j}{h_j} > 0.05$ (i.e., the difference between $h_i$ and $h_j$ is greater than 5%). In this case, $n_i$ transfers $\frac{c_i}{c_i + c_j}(h_i - h_j)$ units of workload to $n_j$. Since there is delay in workload transfer, by the time $n_j$ receives the workload transfer notification from $n_i$ the height of $n_j$ may have increased. If so, $n_j$ would accept part of the workload from $n_i$ according to Assumption 1. The termination of the algorithm is described by a convergence factor $\omega \in (0, 1]$ such that the algorithm terminates at time $t$ where

$$GPE[t] < GPE_{\min} + \frac{GPE[0] - GPE_{\min}}{\omega}.$$ 

In this way, the effect of different $GPE[0]$’s and $GPE_{\min}$’s in different experimental configurations can be eliminated. $\omega$ was set such that the experiments terminated in reasonable periods. Intuitively, the larger the value of $\omega$, the longer the experiment will take to terminate.

### 7.2 Effect of the Balancing Factor on Convergence Rate

In general, if $\gamma$ is too small, workload distribution will be slowed down substantially and so will the convergence rate. In case of large $\gamma$, $n_i$ may transfer too much workload to $n_j$ so that $n_i$ will not have sufficient workload to transfer to the remaining neighbors. However, the performance would not be degraded substantially since $n_j$ may transfer workload back to $n_i$ if necessary. The graph $G_1$ in Figure 1 was used to demonstrate the effect of $\gamma$ on the convergence rate. $\omega$ was set at 128 and $\gamma$ was varied from 0.1 to 1 in steps of 0.05. As can be observed in Figure 8, the completion time (i.e., algorithm...
termination time) drops dramatically for $\gamma^2 < 0.16$ (i.e., $\gamma < 0.4$). For larger values of $\gamma$, the completion time fluctuates at a relatively constant level. A few more experiments were performed on different graph topologies, including mesh, star and linear chain. Similar phenomena were observed, although the point when the curve flattens out varies with different graph topologies. Therefore, it is important to avoid small values of $\gamma$. For some special topologies and workload configurations, there may exist a specific $\gamma$ value which results in optimal convergence rate [23, 19, 24]. However, it is safe to set $\gamma$ to a sufficiently large value to obtain reasonable performance.

7.3 Effect of Processor Capacities on Convergence Rate

In a system where the processors have different speeds, the slower processors are likely to be the bottleneck in workload distribution. Consider the baseline algorithm (refer to Section 5) for example, $\gamma \frac{e_i e_j}{c_i + c_j} (h_i - h_j)$ amount of workload is transferred from $n_i$ to $n_j$ by Assumption 2. For the same values of $\gamma$, $h_i$, $h_j$, and $(c_i + c_j)$, the factor
Figure 9: Effect of processor capacities on convergence rate.

c_i c_j is a parabolic function with maximum value at c_i = c_j. The larger the difference between c_i and c_j, the smaller the value of c_i c_j, and hence less workload is transferred. In see if this is true, the system G_1 was used again in the simulator with γ set to 0.1 and ω to 1024. The initial workload was concentrated at n_5 where l_5 = 20.2, and the capacities of the nodes adjacent to n_5 (i.e., n_3, n_4) were set to smaller values. In particular, 

\{c_1, c_2, c_3, c_4, c_5\} = \{x_4, x_3, x_1, x_0, x_2\} where x_i = \frac{7}{10} [2j + i(1 - j)], where j describes the difference in capacities among the processors, and was varied from 0.05 to 1 in steps of 0.05. Notice that \sum_{i=1}^{5} c_i is a constant independent of j and is equal to 7. Thus the only variables in η are limited to c_{min} and c_{max}. It can be seen from Figure 9 that the convergence time increases with increase in the difference between c_{min} and c_{max}.

7.4 Effect of System Size on Convergence Rate

Intuitively, it takes longer time to balance the workload in a system with more processors for the same value of dia(G). This is because as the number of nodes hanging off
the diameter path (i.e., the path with length equal to $\text{dia}(G)$) increases, more workload is diverted to the additional nodes (see Figure 11). Therefore, the propagation of workload along the diameter path slows down, and it takes more iterations to transfer the same amount of workload to the far end of the diameter path. The system topology $G_2$ in Figure 10 was used to isolate the effect of $|V|$ in $\eta$. In $G_2$, the maximal degree and $\text{dia}(G)$ are kept constant at 3 and $X$ respectively. By setting $c_i = 1$ for all $i$, the factor $\frac{\delta_{\min}^2}{\delta_{\max} \sum c_i}$ degenerates to $\frac{1}{|V|}$, and the variable factor in $\eta$ is limited to $\frac{1}{|V||V|-1}$. Figure 11 shows the experimental results for $\gamma = 1$, $\omega = 1024$, $X = 10$, $l_{\text{leftmost}} = |V|$, and $l_{\text{others}} = 0$. As
expected, the larger the number of processors, the slower the convergence rate.

7.5 Effect of Network Diameter on Convergence Rate

![Diagram](Image)

Figure 12: The system $G_3$.

It is intuitive that it takes longer time for a graph with larger diameter to converge, since the number of iterations to propagate the workload to all the nodes is proportional to $\text{dia}(G)$. To show this relationship, the graph topology $G_3$ in Figure 12 was used, where there are a total of $(3X + 1)$ nodes, $c_i = 1$ for all $i$, and the initial workload concentrates on the leftmost node. In this way, the workload has to transverse $(2X + L + 1)$ nodes in order to reach the rightmost node. If we vary the value of $L$ from 0 to $(X - 1)$, the only variable in $\eta$ is $\text{dia}(G)$. The experimental result obtained by setting $\gamma$ to 1, $\omega$ to 1024 and $X$ to 8 is shown in Figure 13, which confirms the assertion.

8 Conclusions

Most of the existing nearest-neighbor algorithms assume that the processors run at the same speed. We have presented a general hydro-dynamic framework to model the dy-
dynamic load balancing problem for workstations with different processing capacities. The objective is to distribute the workload so that each computer gets its share of the load proportional to its capacity. This paper provides a comprehensive analysis on the performance of this approach.

The hydro-dynamic approach is conceptually easy to understand and provides a framework applicable to many dynamic systems requiring search for an equilibrium state which is globally optimal in some properties. Our main contribution has been in analyzing this framework mathematically. It is proved that all algorithms under the proposed framework converge geometrically for all configurations. Moreover, it is shown that the convergence rate of the framework depends on several factors including the balancing factor, the capacities of the processors as well as the system topology. The relative importance of these parameters as well as their effects on convergence rate have been discussed and validated by simulations.
Appendix

Proof of Lemma 3: We consider three cases: (i) \( h_i[t'] < h_j[t'] \), (ii) \( h_i[t'] \geq h_j[t'] \) where \( h_i[t'] - h_j[t'] \geq h_i[t] - h_j[t] \), and (iii) \( h_i[t'] \geq h_j[t'] \) where \( h_i[t'] - h_j[t'] < h_i[t] - h_j[t] \).

For case (i). It is easy to see that \( \sum_k x_k \leq 0 \) and \( \sum_k y_k \geq 0 \), and so

\[
h_i[t'] < h_j[t']
\]
\[
\Leftrightarrow h_i[t] - h_j[t] < \frac{\sum_k y_k}{c_j} + \frac{-\sum_k x_k}{c_i}
\]
\[
\Leftrightarrow h_i[t] - h_j[t] < \max\left\{ \sum_k y_k, -\sum_k x_k \right\} \left( \frac{1}{c_j} + \frac{1}{c_i} \right)
\]
\[
\Leftrightarrow \max\left\{ \sum_k y_k, -\sum_k x_k \right\} > \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]).
\]

Therefore, at time \( t \) at least one of the quantities \( \sum_k y_k \) and \( -\sum_k x_k \) is greater than \( \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) \) for \( 1 \geq \gamma > 0 \), and hence the lemma is proved.

For case (ii), the flow of workload from \( n_i \) to \( n_j \) is equal to \( \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t']) \) by Assumption 2, and this is greater than or equal to \( \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) \). Therefore, the flow of workload into \( n_j \) at time \( t'' \) must satisfy the lemma requirement.

For case (iii), we consider the effect of \( \sum_k z_k \) on workload migration. If \( \sum_k z_k < 0 \) (see Figure 14(a)), the height of \( n_j \) decreases during the period \([t', t'']\). Since all workload from \( n_i \) is accepted at time \( t'' \), therefore, by Assumption 2

\[
O_i(t, t'') \leq \sum_k x_k - \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t'])
\]  \( \tag{6} \)

and

\[
I_j(t, t'') \geq \sum_k y_k + \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t']).
\]  \( \tag{7} \)
Figure 14: Effect of $\sum_k z_k$ on workload migration.

On the other hand, if $\sum_k z_k \geq 0$ (Figure 14(b)), $n_j$ must reject part of the workload from $n_i$ in order to satisfy Assumption 1. In this case, $n_j$ can only accept an amount of workload equal to $\gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t''])$ from $n_i$. Hence,

$$O_i(t, t'') \leq \sum_k x_k - \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t'']) = \sum_k x_k - \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t'']) + \gamma \frac{c_i}{c_i + c_j} \sum_k z_k$$

and

$$I_j(t, t'') \geq \sum_k y_k + \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t'']) + \sum_k z_k = \sum_k y_k + \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t'] - h_j[t'']) + \left(1 - \gamma \frac{c_i}{c_i + c_j}\right) \sum_k z_k.$$

Since Equations (6) and (7) are special cases of Equations (8) and (9) when $\sum_k z_k = 0$, Equations (6) and (8) can be combined to produce Equation (12), and Equations (7) and (9) can be combined to produce Equation (10) by replacing $\sum_k z_k$ in Equations (8)
and (9) to $Z$ where $Z \geq 0$:

$$
\mathcal{O}_i(t, t^\prime) \leq \sum_k x_k - \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t^\prime]) + \gamma \frac{c_i}{c_i + c_j} Z
$$

$$
= \sum_k x_k - \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] + \frac{\sum_k x_k}{c_i} - h_j[t] - \frac{\sum_k y_k}{c_j}) + \gamma \frac{c_i}{c_i + c_j} Z
$$

$$
= -\gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) + \sum_k x_k - \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \gamma \frac{c_i}{c_i + c_j} \sum_k y_k
$$

$$
+ \gamma \frac{c_i}{c_i + c_j} Z
$$

$$
= -\gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) + \Phi_i 
$$

where

$$
\Phi_i = \left(1 - \gamma \frac{c_j}{c_i + c_j}\right) \sum_k x_k + \gamma \frac{c_i}{c_i + c_j} \sum_k y_k + \gamma \frac{c_i}{c_i + c_j} Z. 
$$

Similarly,

$$
\mathcal{I}_j(t, t^\prime) \geq \sum_k y_k + \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t^\prime]) + \left(1 - \gamma \frac{c_i}{c_i + c_j}\right) Z
$$

$$
= \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) + \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \gamma \frac{c_i}{c_i + c_j} \sum_k y_k - \gamma \frac{c_i}{c_i + c_j} \sum_k y_k
$$

$$
+ \left(1 - \gamma \frac{c_i}{c_i + c_j}\right) Z
$$

$$
= \gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t]) + \Phi_j .
$$

where

$$
\Phi_j = \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \left(1 - \gamma \frac{c_i}{c_i + c_j}\right) \sum_k y_k + \left(1 - \gamma \frac{c_i}{c_i + c_j}\right) Z. 
$$

Therefore, if $\Phi_i \leq 0$, then $\mathcal{O}_i(t, t^\prime) \leq -\gamma \frac{c_i c_j}{c_i + c_j} (h_i[t] - h_j[t])$ and the lemma is proved. For
the case when $\Phi_i > 0$, we have to prove that $\Phi_j \geq 0$. Now the inequality $\Phi_i > 0$ yields

\[
\sum_k x_k - \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \gamma \frac{c_i}{c_i + c_j} \sum_k y_k + \gamma \frac{c_i}{c_i + c_j} Z > 0
\]

\[
\Leftrightarrow \sum_k y_k > \frac{1}{\gamma} \frac{c_i + c_j}{c_i} \left[ \left( \gamma \frac{c_j}{c_i + c_j} - 1 \right) \sum_k x_k - \gamma \frac{c_i}{c_i + c_j} Z \right].
\]

Substituting the value of $\sum_k y_k$ into $\Phi_j$ gives

\[
\Phi_j = \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \sum_k y_k - \gamma \frac{c_i}{c_i + c_j} \sum_k y_k + \left( 1 - \gamma \frac{c_i}{c_i + c_j} \right) Z
\]

\[
> \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \frac{1}{\gamma} \frac{c_i + c_j}{c_i} \left[ \left( \gamma \frac{c_j}{c_i + c_j} - 1 \right) \sum_k x_k - \gamma \frac{c_i}{c_i + c_j} Z \right] \left( 1 - \gamma \frac{c_i}{c_i + c_j} \right)
\]

\[
+ \left( 1 - \gamma \frac{c_i}{c_i + c_j} \right) Z
\]

\[
= \gamma \frac{c_j}{c_i + c_j} \sum_k x_k + \frac{1}{\gamma} \frac{c_i + c_j}{c_i} \left( \gamma \frac{c_j}{c_i + c_j} - 1 \right) \left( 1 - \gamma \frac{c_i}{c_i + c_j} \right) \sum_k x_k
\]

\[
= \frac{\gamma - 1}{\gamma} \frac{c_i + c_j}{c_i} \sum_k x_k.
\]

(14)

Since

\[
h_i[t'] - h_j[t'] < h_i[t] - h_j[t] \Leftrightarrow \sum_k x_k < \frac{\sum_k y_k}{c_j},
\]

(15)

therefore, if $\sum_k x_k \geq 0$, by Equation (15) $\sum_k y_k \geq 0$ also, and so according to Equation (13) it follows that $\Phi_j \geq 0$. On the other hand, if $\sum_k x_k < 0$, then by Equation (14) we know that $\Phi_j \geq 0$. As a result, $\Phi_i > 0$ implies that $\Phi_j \geq 0$ and so $I_j(t, t') \geq \gamma \frac{c_i}{c_i + c_j} (h_i[t] - h_j[t])$.

\[
\square
\]

**Proof of Lemma 4:** Let $\Delta GPE_\zeta$ be the reduction in GPE given the flow pattern $\zeta$. To prove the lemma, we compare the change in GPE for all other flow patterns (say $\xi$) to that of $\zeta$. An important property of the flow pattern $\xi$ is that $\xi_j$ is not the same for all $j$, which is different from the flow pattern $\zeta$ where $\zeta_k$ is equal for all $k$. 

34
Assume the flow pattern at step $\tau$ is $X[\tau]$ and $\xi = X[0]$. We partition $X[\tau]$ into three sets: $X_{\text{lower}} = \{ n_j | \xi_j[\tau] \leq \zeta_k \}$, $X_{\text{equal}} = \{ n_j | \xi_j[\tau] = \zeta_k \}$ and $X_{\text{higher}} = \{ n_j | \xi_j[\tau] > \zeta_k \}$. It is easy to see that at $\tau = 0$, $\sum_j \xi_j[\tau] = \Delta h_i$, and

$$
\sum_{n_j \in X_{\text{lower}}} (\zeta_k - \xi_j[\tau]) = \sum_{n_j \in X_{\text{higher}}} (\xi_j[\tau] - \zeta_k).
$$

Select any $n_a \in X_{\text{higher}}$ and $n_b \in X_{\text{lower}}$. There are two cases to consider: (i) If $\xi_a[\tau] - \zeta_k \geq \zeta_k - \xi_b[\tau]$, then for the new flow pattern $X[\tau + 1]$, the height of $\xi_a[\tau + 1]$ and $\xi_b[\tau + 1]$ are set to $\xi_a[\tau] + \xi_b[\tau] - \zeta_k$ and $\zeta_k$ respectively, and (ii) if $\xi_a[\tau] - \zeta_k < \zeta_k - \xi_b[\tau]$, then for the new flow pattern $X[\tau + 1]$, the height of $\xi_a[\tau + 1]$ and $\xi_b[\tau + 1]$ are set to $\zeta_k$ and $\xi_a[\tau] + \xi_b[\tau] - \zeta_k$ respectively. In both cases, $\sum_j \xi_j[\tau + 1] = \Delta h_i$ and the size of $X_{\text{equal}}$ is increased by at least 1. After the procedure is repeated for at most $(d_i - 1)$ times, $X[\tau] = \zeta$ for $\tau \geq d_i - 1$.

Now we consider the change in $\Delta GPE$ for each step. Let

$$
\Psi = \sum_{j \neq a, b} \Delta PE(n_j),
$$

35
For case (i), since $\xi_a[\tau + 1] = \xi_a[\tau] + \xi_b[\tau] - \zeta_k$ and $\xi_k[\tau + 1] = \zeta_k$, so,

$$
\Delta GPE_{X[\tau]} = \Psi + \Delta PE(n_a) + \Delta PE(n_b)
$$

$$
< \Psi - c_i \xi_a[\tau]^2 / 2 - c_i \xi_b[\tau]^2 / 2 \quad \text{by Lemma 1 and Assumption 1}
$$

$$
= \Psi - \frac{c_i}{2} \left[ (\xi_a[\tau + 1] + \zeta_k - \xi_b[\tau])^2 + \xi_b[\tau]^2 \right]
$$

$$
= \left( \Psi - \frac{c_i}{2} \xi_a[\tau + 1]^2 - \frac{c_i}{2} \xi_b[\tau + 1]^2 \right)
$$

$$
- \frac{c_i}{2} \left( 2\xi_a[\tau + 1]\zeta_k - 2\xi_a[\tau + 1]\xi_b[\tau] - 2\xi_k\xi_b[\tau] + 2\xi_b[\tau]^2 \right)
$$

$$
= \Delta GPE_{X[\tau+1]} - c_i \cdot (\xi_a[\tau + 1] - \xi_b[\tau]) (\zeta_k - \xi_b[\tau])
$$

$$
< \Delta GPE_{X[\tau+1]}.
$$

For case (ii), since $\xi_a[\tau + 1] = \zeta_k$ and $\xi_k[\tau + 1] = \xi_a[\tau] + \xi_b[\tau] - \zeta_k$, so,

$$
\Delta GPE_{X[\tau]} < \Psi - c_i \xi_a[\tau]^2 / 2 - c_i \xi_b[\tau]^2 / 2 \quad \text{by Lemma 1 and Assumption 1}
$$

$$
= \Psi - \frac{c_i}{2} \left[ \xi_a[\tau]^2 + (\xi_b[\tau + 1] + \zeta_k - \xi_a[\tau])^2 \right]
$$

$$
= \left( \Psi - \frac{c_i}{2} \xi_a[\tau + 1]^2 - \frac{c_i}{2} \xi_b[\tau + 1]^2 \right)
$$

$$
- \frac{c_i}{2} \left( 2\xi_b[\tau + 1]\zeta_k - 2\xi_b[\tau + 1]\xi_a[\tau] - 2\xi_a[\tau]\zeta_k + 2\xi_a[\tau]^2 \right)
$$

$$
= \Delta GPE_{X[\tau+1]} - c_i \cdot (\xi_a[\tau] - \xi_b[\tau + 1]) (\xi_a[\tau] - \zeta_k)
$$

$$
< \Delta GPE_{X[\tau+1]}.
$$

In both cases $\Delta GPE_{X[\tau]} < \Delta GPE_{X[\tau+1]}$. Therefore, $\Delta GPE_\epsilon = \Delta GPE_{X[0]} < \Delta GPE_{X[\epsilon-1]} = \Delta GPE_\epsilon$ and hence Lemma 4 is proved. □

References


