An Equilibrium Model of a Self-organizing Network Architecture

Guidance

Professor	Masao FUKUSHIMA
Assistant Professor	Nobuo YAMASHITA

Naomi MIYAMOTO

1999 Graduate Course

in

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2001

Abstract

Future network services and applications must be flexible, highly available and survivable with minimal human configuration and management. This paper proposes a self-organizing future network architecture that is flexible, highly available and survivable.

The proposed network consists of the following architectural components: *cyber entities* and *Bio-net platforms*. A cyber entity is a mobile agent with autonomous behavior. A Bio-net platform is an execution environment for cyber entities; it runs on a network node and manages resources on the node (e.g. CPU, memory).

In the proposed network, a cyber entity provides a service (e.g. a web server) to users in exchange for energy. It also consumes network resources such as CPU and memory, and pays energy to the platform for using the resource that the platform manages. Associated with a Bio-net platform is a utility function. A platform determines the prices of resources based on its utility function. Similarly, a utility function is also associated with a cyber entity, and a cyber entity determines the amount of resource it consumes based on its utility function. A utility for a cyber entity, and thus, the amount of resource that a cyber entity consumes, depends on various system variables such as the amount of resource consumed by other cyber entities within N hops from the cyber entity, and the price of resources and the number of users that are within N hops. Similarly, a utility of a Bio-net platform also depends on various variables such as resource prices on other platforms and the amount of resources that cyber entities consume. When each cyber entity and Bio-net platform are able to make their best decision on the amount and the price of the resources they consume, we say that our network architecture has an *equilibrium point*.

In this paper, we first build a mathematical model of the proposed network architecture. Next, we build a simplified model and derive its equilibrium conditions. Moreover, we show that our original mathematical model has an equilibrium point when the simplified model has an equilibrium point. Finally, we demonstrate through simulations that our original model has an equilibrium point under the conditions for the existence of an equilibrium point in the simplified model.

Contents

1 Introduction

The explosive growth of the Internet has enabled resource sharing on a worldwide basis. We envision that further evolution of the Internet and mobile computing technologies allow wide access to personalized and collaborative computing services anywhere in the world. In such a network, network services are required to be flexible, highly available and survivable with minimal human configuration and management. However, today's networks are unable to provide such features. Research to design a network with such features has only begun recently [1, 2, 3]. Among existing research, approaches based on biological systems seem to be most promising [5, 6]. In real-world biological systems, biological agents act through local interactions and based on local information of the environment, and they adapt to dynamic environment changes. The aim of approachs based on biological systems is to examine whether concepts and mechanisms from real-world biological systems (e.g. local interactions) can be applied to the future network.

In this paper, we propose a self-organizing network architecture that is flexible, highly available and survivable. Our network consists of architectural components: *cyber entities* and *Bio-net platforms.* A cyber entity is a distributed autonomous entity (mobile agent) and provides a service such as a hotel reservation service to users in exchange for energy. A cyber entity is analogous to an individual bee in the biological world. Like their biological counterparts, cyber-entities follow biological behaviors, such as *replication, death* and *replication.* A Bio-net platform is an execution environment for cyber entities running on a network node and manages resources on a node (e.g. CPU, memory).A cyber entity consumes some network resources such as CPU and memory, and pays energy for using such resources to the Bio-net platform it resides on (see Fig.1). Associated with a Bio-net platform is a utility function[1] . A Bio-net platform decides the prices of resources based on its utility function. Similarly, a cyber entity possesses a utility function and decides the amount of resource it consumes based on its utility function. When an energy level of a cyber entity is high, indicating that there was a large demand for the service that the cyber entity provides, this cyber entity may replicate (increase its amount of consuming resource). Conversely, when its energy level is low, indicating that the demand for the cyber entity's service was small, this cyber entity may die (decrease its consumption). Like in real-world biological systems, different platforms and cyber entities may have different utility functions. (For example, one Bio-net platform may choose its resource prices so that the price maximizes its energy income, while another platform may choose its price so that only a few cyber entities may use its resources.)

The rest of the paper is organized in the following manner. In section 2, we present

[1] A utility function is a measurement of happiness. The function depends on vareous variables, such as the amount of resource a cyber entity consumes, or the resource prices, or the resource prices at neighboring platforms.

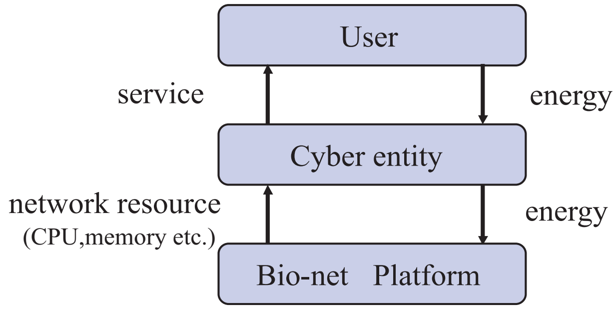

Figure 1: The relations between users, cyber entities and Bio-net platforms.

a brief description of our network architecture. In section 3, we formulate our network architecture into a mathematical model and derive a utility maximization problem that a cyber entity and a Bio-net platform solves. In section 4, we simplify our mathematical model. For this simplified model, we apply Debreu's theorem and derive equilibrium conditions under which the solution to the utility maximization of all cyber entities and Bio-net platforms become finite. Here, we simplify our model so that when the simplified model has an equilibrium point, our original model also has an equilibrium point. In section 5, we demonstrate through simulations that our original model has an equilibrium point under the conditions obtained for our simplified model in section 4. Section 6 concludes the paper.

2 Biologically-Inspired Architecture

While real-world biological systems consist of large number of entities, they have many desirable characteristics and features: decentralized control and survivability. The Biologically-Inspired Architecture proposed in [6] and studied in this paper examines whether concepts and mechanisms from real-world biological systems such as immune systems, bee colonies and human societies, can be applied to the future Internet environment [4].

In the proposed Biologically-Inspired Architecture, there are two architectural components: *cyber entities* and *Bio-net platforms*. In the following, we will describe about these components and about users in details.

- Cyber entity: Cyber entities are distributed autonomous mobile agents. They contain some service (ex. a web server), and provide (deliver) the service to human users in exchange for energy units. A cyber entity consists of three parts, *body*, *attributes* and *behavior*. The *body* contains materials relevant to the service that the cyber entity provides. The *body* may contain data, application code, or user profiles. The *attribute* part contains information about the cyber entity itself, such as the name, a unique identification number, and its utility function.A cyber entity decides the amount of resource it consumes so that it maximizes the value of its utility function. Like in real-world biological systems, different cyber entities may have different utility functions. The behavior part contains behaviors, or blocks of executable code that allow the cyber entity to behave autonomously, like an individual bee in the biological world. It also contains some behaviors to mimic simple biological behaviors such as *migration, replication* and *death*.

- Bio-Networking platform software: In order to support cyber entities, a network node must install a software, which we refer to as a Bio-Networking platform (Bio-net platform for short). The Bio-networking platform provides an execution environment and basic system functions necessary to support cyber entity's behavior such as migration, replication, and death. Similarly to a cyber entity, a Bio-net platform also contains its utility function. A Bio-net platform decides the prices of resources based on its utility function. A cyber entity pays energy units to the Bio-net platform in order to use its network resources such as CPU, memory, disk space, and network bandwidth. Bio-networking platform controls the use of all network node resources by adjusting the price of its resources. For example, when a Bio-net platform tries to limit the use of the network node resources, it may raise the price of the resources. This may cause cyber entities with less energy to migrate away from the node, resulting in limiting use of its resources.

- User: Similarly to the energy exchange between cyber entities and Bio-net platforms, users and cyber entities exchange energy as well. When a user receives a service from a cyber entity, the user gives energy to the cyber entity.

3 The Model

Let us explain our model. Throughout this paper, the following assumptions and notations are used.

3.1 Assumptions

(A1) There are m network nodes, each running a Bio-net platform, in the network. Each of these m nodes supports l different types of resources. Amount of type k resource at platform i is α_i^k, and α_i^k is finite.

(A2) There are n kinds of services (n types of cyber entities) in the network. Cyber entities of the same types provide the same service. Note that there may be multiple cyber entities of the same type in the network.

(A3) The amount of resource that a cyber entity consumes varies from a lower limit β_k to an upper limit γ_k $(k = 1, \ldots, l)$.

(A4) The price of resource varies from a lower limit 0 to an upper limit p^{max}. Here, p^{max} is arbitrary large, but finite.

(A5) Cyber entities and Bio-networking platforms do not have global information of the network: they only have local information. Specifically, cyber entities on platform i only knows the prices of resources on platform i and on platforms that are within N hops from platform i. In the analysis, we assume N is reasonably small, and thus, cyber entities only possess local information.

(A6) We assume that the amount of resources that a cyber entity consumes is a continuous variable in our model. This assumption also applies when a cyber entity migrates. In other words, when a cyber entity migrates from platform A to platform B, resource that the cyber entity consumes at platform A gradually reduces, and the resource that the cyber entity consumes at platform B gradually increases, keeping the sum of the two constant.

(A7) Cyber entities may only migrate to the neighboring platforms that are within 1 hop away.

(A8) Time required for cyber entities to migrate and for user request packets (for service) to propagate through a network is assumed to be zero. Note that with this assumption, all cyber entities and user request packets are on Bio-net platforms at any given time, and there is no cyber entities and user request packets that are in transit on a channel between Bio-net platforms.

Assumption (A4) and (A6) makes our analysis easier. In Assumption (A8), we claim that cyber entities and user's request packets reside on a Bio-net platform at every moment.

3.2 Notations

In the Biologically-Inspired Architecture, Bio-net platforms decide the prices of resources, and cyber entities are required to give energy to Bio-net platforms in order to run their execution codes (See Section2.1). Since there are l types of resources in the network, Bio-net platform i determines the prices of these l types of resources $p_i^k (k = 1, \ldots, l)$ on the platform. p_i^k represents the price of resource of type k on platform i and p_i denotes the prices on platform i,

$$p_i = (p_i^1, p_i^2, \ldots, p_i^l) \in \Re_+^l. \tag{3.1}$$

Here, $\Re^l_+$ is a set of non-negative vectors, and $y \in \Re^l_+$ indicates that all the components of vector y takes a nonnegative value. Let $p = (p_1, \ldots, p_m) \in \Re^{lm}_+$ be a collection of prices of cyber entities in the entire network.

Let $u_i^j \in \Re_+$ denote the number of request packets (see section2.1) that are on Bio-net platform i and are for service j. Also, let $u_i = (u_i^1, \ldots, u_i^n) \in \Re^n_+$ denote the set of request packets on platform i and $u = (u_1, \ldots, u_m) \in \Re^{nm}_+$ denote a collection of request packets for all platforms in the network.

As mentioned earlier, each cyber entity decides the amount of resources it consumes. More explicitly, a cyber entity examines the prices of resources and the number of request packets on the platform it resides, and determines the amount of resource it consumes. Furthermore, it decides if it remains on the platform or moves to a neighboring platform. First, let CE_i^j denote a cyber entity whose service type is j and resides on platform i. $x_{i_k}^j$ represents the amount of resource of type k that CE_i^j consumes on platform i. Since there are l resources in the network, the amount of resource CE_i^j consumes is given by

$$x_i^j = (x_{i_1}^j, x_{i_2}^j, \ldots, x_{i_l}^j) \in \Re^l_+. \tag{3.2}$$

The vector $x_i = \sum_{j=1}^n x_i^j = (\sum_{j=1}^n x_{i_1}^j, \sum_{j=1}^n x_{i_2}^j, \ldots, \sum_{j=1}^n x_{i_l}^j) \in \Re^l_+$ represents a total amount of resource that cyber entities consume on platform i. Let x^j denote a vector representing the amount of resources that cyber entities of type j consume at different platforms. Namely, $x^j = (x_i^j, \ldots, x_{i'}^j) \in \Re^{l|S_j(t)|}_+ \quad i, i' \in S_j(t)$ where $|\{Y\}|$ indicates the number of components in set Y. Here, $S_j(t)$ denotes a set of platforms where cyber entities of service type j reside at time t (Refer to Fig.2). [2] The vector $x = (x^1, \ldots, x^n) \in \Re^{l\sum_{j=1}^n |S_j(t)|}_+$ is a collection of the amount of resources that cyber entities consume in the entire network.

In order to describe the neighboring environment of every architectural component, let the upper script loc of a component X_i, namely X_i^{loc} represent the variables of Xs within N hops of platform i. For example, u_i^{loc} is a vector, each component of which representing the number of user's request packets on the platform within N hops of platform i. Similarly, let $S_j^{loc}(t)$ denote a set of platforms, consisting of $S_j(t)$ and also the nodes that are within N hops from a platform in $S_j(t)$ (See Fig.3). Furthermore, let us denote $S(t) = S_1(t) \cup S_2(t) \cup \cdots \cup S_n(t)$ and $S^{loc}(t) = S_1^{loc}(t) \cup S_2^{loc}(t) \cup \cdots \cup S_n^{loc}(t)$. $S(t)$ represents a set of nodes, on each of which at least one cyber entity resides at time t. $S^{loc}(t)$ represents a set of platforms, consisting of platforms in $S(t)$ and the platforms that are within N hops from a platform in $S(t)$. $S^{loc}(t)$ is a set of platforms whose information is known by at least one cyber entity at time t. Note that none of the cyber entities at time t have any information of $\{1, 2, \ldots, m\}/S^{loc}(t)$ (See Fig.3). Here, $Z = \{X/Y\}$ indicates that Z is a

[2] Since cyber entities may migrate, $S_j(t)$ depends on time. In order to emphasize this time dependency, we include time variable t in the notation, $S_j(t)$.

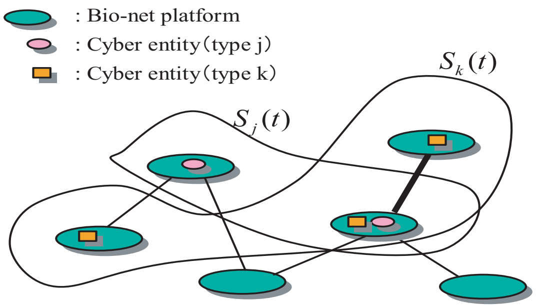

Figure 2: Explanation of $S_j(t)$.

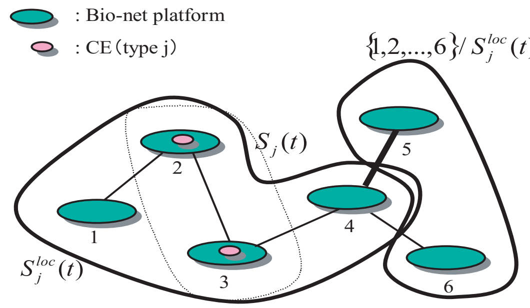

Figure 3: Explanation of $S_j^{loc}(t)$ and $\{1, \ldots, m\}/S_j^{loc}(t)$.

set of X excluding Y.

Platforms and cyber entities decide their prices and the amount of resource they consume based on local information. Furthermore, the amount of resource a cyber entity consumes and the price of resources depend on each other. Thus, the prices of a platform is influenced by cyber entities which reside within N hops. Similarly, the amount of resource a cyber entity consumes is influenced by prices that are within N hops from the cyber entity. Thus, a cyber entity is influenced by other cyber entities indirectly through resource prices. To represent such dependency, let $x_i^{-j} = (x_1^1, \ldots, x_{i-1}^j, x_{i+1}^j, \ldots, x_m^n) \in \Re_+^{l\{\sum_{j=1}^n |S_j(t)|-1\}}$, $p_{-i} = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m) \in \Re_+^{l(m-1)}$. Here, x_i^{-j} is a vector excluding x_i^j from x, and p_{-i} is a vector excluding p_i from p. Note that CE_i^j may only control x_i^j and platform i may only control p_i.

In the following two subsections, we observe the system at a given time t and a given user request distribution u. Later in section 4, we give a definition of an equilibrium point, and observe the system at an equilibrium point for a given u. In section 4.2, we derive conditions under which our model has an equilibrium point for any u.

3.3 Maximization of Cyber entity's Utility

In this subsection, without loss of generality, we observe a cyber entity of service type j which resides on platform i (namely, CE_i^j) at a given time t. Associated with each cyber entity is a *utility function*, and cyber entity decides the amount of resource it uses based on it. Cyber entities of the same type share the same utility function. For example, the best size of memory that a cyber entity of type j perform a service may be b^j, and this may be expressed in its utility function. [3] Let U_i^j denote the utility function of CE_i^j. CE_i^j decides the amount of resource it uses in such a way to maximize $U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})$.

[3] An easy example for the utility function is $U_i^j = -(x_i^j - b^j)^2$.

Note that the utility function of CE_i^j depends on the amount of resource that other cyber entities consume, resource prices and the number of user request packets who are within N hops of CE_i^j: x_i^{-jloc} , u_i^{loc} and p_i^{loc}. Because CE_i^j only controls the amount of resource it uses (namely, x_i^j), it maximizes U_i^j over x_i^j. Note also that the dimension of x_i^{-jloc} vary according to time.

From the above discussion, given x_i^{-jloc} , u_i^{loc} and p_i^{loc}, CE_i^j desires to use x_i^j amount of resource that maximizes

$$
\begin{aligned}
(P1) \quad \text{maximize}_{x_i^j} \quad & U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc}) \\
& \beta^j \geq x_i^j \geq \gamma^j .
\end{aligned}
$$

Here, the constraints come from Assumption $(A3)$. Note that the solution of $(P1)$ is a function of $x_i^{-jloc}, p_i^{loc}, u_i^{loc}$, and we denote it as $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$.

3.4 Maximization of Bio-net Platform's Utility (Nonlinear Bilevel Program)

In this subsection, without loss of generality, we observe platform i at time t. Similar to cyber entities, a Bio-net platform decides the prices of resources based on its utility function. Let $\hat{U}_i(p_i; x_i)$ denote the platform i's utility function. Note that platform i tries to maximize $\hat{U}_i(p_i; x_i)$ over p_i.

Cyber entities decide the amount of resource it consumes given the price of resources. On the other hand, a platform's utility $\hat{U}_i$ depends on the amount of resource cyber entities consume. Thus, in maximizing utility, Bio-net platforms must consider how much resource cyber entities consume when deciding their resource prices. More explicitly, when a platform decides its resource prices, cyber entity's maximization problems $(P1)$ are included in its constraint (3.6) [4] . To summarize, a Bio-net platform decides the price of its resources that maximizes

$$
\begin{aligned}
(P2) \quad \text{maximize}_{p_i} \quad & \hat{U}_i(p_i; x_i) \\
\text{subject to} \quad & p^{max} \geq p_i \geq 0 & (3.3) \\
& \alpha_i \geq x_i = \sum_{j \in \{j | i \in S_j(t)\}} x_i^j \geq 0 \qquad i = 1, \ldots, m & (3.4) \\
& x_i^j \in \text{argmax}\{U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc}) | \beta^j \geq x_i^j \geq \gamma^j\} & (3.5) \\
& \quad \text{for all} \quad j = 1, \ldots, n, \quad i \in S(t).
\end{aligned}
$$

$x^* \in \text{argmax}\{f(x)|x \in D\}$ indicates that x^* gives the maximum value of f on D. Namely, $\text{argmax}\{f(x)|x \in D\} := \{x \in D | f(x) = \max f\}$. The constraint (3.4) comes from Assumption $(A3)$, and (3.5) comes from $(A4)$. $j \in \{j|i \in S_j(t)\}$ indicates a set of types

[4] An optimization problem including a optimization problem in its constraint is known as Nonlinear Bilevel Program.

of cyber entity, which locate on platform i at time t. Observing $(P2)$, we may see that platform i maximizes its utility function $\hat{U}_i$ under conditions that the total amount of consumption will not exceed α_i. Note that p_i influences cyber entities not only the ones that are within N hops, but also the ones that cannot be reached within N hops. For example, cyber entities that cannot be reached within N hops but can be reached within $2N$ hops from platform i are influenced by some of the cyber entities who are within N hops from platform i (See Fig.4). Since cyber entities that are within N hops from platform i are influenced by p_i, cyber entities that cannot be reached within N hops are also influenced by p_i indirectly.

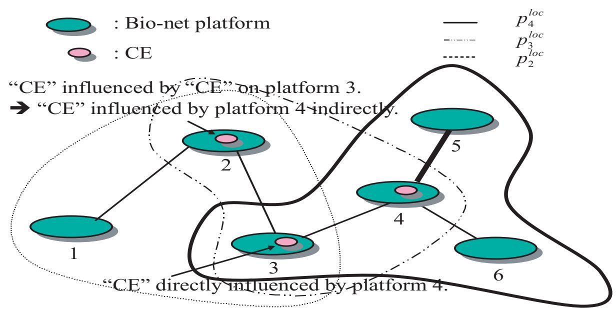

Figure 4: Cyber entity on platform 3 is influenced by cyber entities, user request packets and platforms that are within N hops from it. In the figure, we assume $N = 1$. Thus, cyber entity on platform 3 is influenced by platform 4. Similarly, a cyber entity on platform 2 is influenced by cyber entity on platform 3. Combining these together, we can see that a cyber entity on platform 2 is also influenced by platform 4.

Remark 3.1 Note that platforms are influenced not only by the amount of resource cyber entities consume, but also by the prices of other platforms. The explanation is given below.

By substituting the solution of $(P1)$, namely $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ for all $j \in \{j|i \in S_j(t)\}$ into (3.6), $(P2)$ may be converted into the following problem

$$
\begin{array}{lll}
(P2') & \text{maximize}_{p_i} & \hat{U}_i \left(p_i; \sum_{j\in\{j|i\in S_j(t)\}} x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})\right) \\
& \text{subject to} & p^{max} \geq p_i \geq 0 \\
& & \alpha_i \geq x_i^* = \sum_{j\in\{j|i\in S_j(t)\}} x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc}) \geq 0 \quad i = 1, \dots, m
\end{array}
$$

From $(P2')$, we may see that the utility function of platform i depends on the consumption amount of resources on other platforms x_i^{-jloc}, the prices of other platforms p_i^{loc} and user's request packets that are within N hops of i, u_i^{loc}.

CE_i^j solves $(P1)$ to determine the amount of resource it consumes. More explicitly, a cyber entity substitutes its local information such as $x_i^{-jloc}, p_i^{loc}, u_i^{loc}$ into $(P1)$ and solve it.

Note that even the same types of cyber entities solve different problems when the platforms they reside on are different. Each cyber entity CE_r^k solves the problem $(P1)$ where $j = k$ and $i = r$. Moreover, all cyber entities solve their problems simultaneously and repeatedly, and changes the consumption amount. Cyber entities solve $(P1)$ repeatedly in order to adapt to their local environment $x_i^{-jloc}, p_i^{loc}, u_i^{loc}$, which may change dynamically. When a cyber entity *reproduces* (*dies*), it consumes more (less) resource.

Since cyber entities decide the amount of resource they consume simultaneously, we may eliminate the variables of cyber entities from $(P2')$ by substituting all the solutions of cyber entities. Thus, we may rewrite $(P2')$ to the following.

$$\begin{array}{lll} (P2'') & \text{maximize}_{p_i} & \hat{U}_i(p_i; p_{-i}, u) \\ & \text{subject to} & p^{max} \geq p_i \geq 0 \\ & & \alpha_i \geq x_i^*(p, u) = \displaystyle\sum_{j \in \{j | i \in S_j(t)\}} x_i^{j*}(p, u) \geq 0 \quad i = 1, \ldots, m. \end{array} \tag{3.6}$$

Let us now turn to the equilibrium aspect of our model. Let us first define an equilibrium point.

4 An Equilibrium Point

Definition 4.1. *The amount of resources used by cyber entities and the resource prices set by platforms for a given user request vector u in an* equilibrium point*, (x^*, p^*, u) satisfies the following condition (C).*

$$(C) \qquad \hat{U}_i(p_i^*; p_{-i}^*) = \max_{p_i} \hat{U}_i(p_i; p_{-i}^*) \quad \textit{forall} \quad i = 1, \ldots, m,$$

where p_i^* represents the solution to $(P2'')$.

Note that our model operates at a different equilibrium point when a distribution of user request packets u is different. At an equilibrium point (x^*, p^*, u), all m problems like $(P2)$ are maximized at once. Thus, given u, solving $(P2'')$ for all $i = 1, \ldots, m$ simultaneously will yield an equilibrium point for that u.

Bio-net platforms $i = 1, \ldots, m$ solve $(P2)$ simultaneously in order to attain an equilibrium point for a given u.

Given u, an equilibrium point for our natwork architecture may be obtained as a solution to the following.

$$
\begin{aligned}
(P3) \quad & \text{maximize}_{p_i} \quad && \hat{U}_i(p_i; x_i) \\
& \text{subject to} && p^{max} \geq p_i \geq 0 \\
& && \alpha_i \geq x_i = \sum_{j \in \{j | i \in \bar{S}_j\}} x_i^j \geq 0 \qquad i = 1, \ldots, m \\
& && x_i^j \in \text{argmax}\{U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc}) | \beta^j \geq x_i^j \geq \gamma^j\} \qquad (4.1) \\
& && \qquad \text{for all} \quad i \in \bar{S}_j, j = 1, \quad \ldots, n
\end{aligned}
$$

Here, $\bar{S}_j \in \{1, \ldots, m\}$ is a set of platforms where cyber entities of type j are located in an equilibrium. Since an equilibrium point is a concept without time, note that time t is not included in $(P3)$. Also, note that $\bar{S}_j (j = 1, \ldots, n)$ depends on the migration policy of cyber entities and initial distribution of cyber entities over platforms. Since $\bar{S}_j (j = 1, \ldots, n)$ is not fixed, this makes it difficult to derive equilibrium conditions. In the following section, we will introduce a simpler model, where dependency on migration policy and initial settings are removed.

4.1 A Simplified Model

Note that the goal is to derive the equilibrium condition for a given u. This is equivalent to deriving a condition, where m sets of $(P3)$ is satisfied at once. In the following, let us call the m sets of $(P3)$, an original model.

In the simplified model, the following assumption is added to our original model.

> We assume that all types of cyber entities are present on each of the Bio-net platforms. We further assume that, when the solution to $(P1)$ (namely $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$) is zero, CE_i^j does not consume any resource on the platform i (i.e., Cyber entities consume zero resources), and do not provide service to any users. When the solution to $(P1)$ (namely $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$) is non-zero, CE_i^j consumes $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ amount of resource, and provide service to users.

The above assumption is incorporated into the simplified model in the following manner. Let

$$
y_i^j = \begin{cases} 1, & \text{when } CE_i^j \text{ exists} \\ 0, & \text{otherwise} \end{cases}
$$

In our simplified model, let us rewrite a vector representing the amount of resources that cyber entities of type j consume at different platforms (namely x^j in our original model) as $x^j = (x_1^j y_1^j, \ldots, x_m^j y_m^j) \in \Re_+^{lm}$. Also, let us rewrite a collection of the amount of resources that cyber entities consume in the entire network (namely x) as $x = (x^1, \ldots, x^n) \in \Re_+^{lmn}$.

Note that the above assumption introduced in our simplified model implies that cyber entities do not migrate in the simplified model. Since cyber entities of all types reside on any platform in the simplified model, migration of a cyber entitiy in the simplified model is considered a combination of a death of a cyber entity on one platform and a birth of a new cyber entity of the same type at a different platform. Moreover, the assumption introduced in the simplified model implies that the simplified model does not depend on the initial location distribution of cyber entities, since cyber entities reside on any platform any time in the simplified model.

From the above discussions, we consider the following simplified model, which is associated with our orignal model (namely m sets of $(P3)$).

$$
\begin{aligned}
(P_s) \quad \text{maximize}_{p_i} \quad & \hat{U}_i(p_i; x_i) \\
\text{subject to} \quad & p^{max} \geq p_i \geq 0 \\
& \alpha_i \geq x_i = \sum_{j=1}^{n} x_i^j \geq 0 \qquad i = 1, \ldots, m \\
& x_i^j \in \text{argmax}\{U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc}) | \beta^j \geq x_i^j \geq \gamma^j\} \qquad (4.2) \\
& \quad \text{for all} \quad j = 1, \ldots, n, \quad i = 1, \ldots, m. \qquad (4.3)
\end{aligned}
$$

where $x^j = (x_1^j y_1^j, \ldots, x_m^j y_m^j) \in \Re_+^{lm}$ and $x = (x^1, \ldots, x^n) \in \Re_+^{lmn}$.

In our simplified model, all platforms $i = 1, \ldots, m$ solve (P_s) simultaneously.

Now, let us prove that the equilibrium point of the simplified model is also a equilibrium point of the original model.

Proof.

In our original model, all platforms $i = 1, \ldots, m$ solve $(P3)$. Thus, a whole system may be written as m sets of $(P3)$. On the other hand, in our simplified model, a whole system may be written as m sets of (P_s). Now, let us compare $(P3)$ and (P_s). Since $\bar{S}_j \in \{1, \ldots, m\}$ in $(P3)$ holds, there are more constraints in (P_s) than those in $(P3)$ and the constraints in (P_s) is a super set of those in $(P3)$. Thus, the feasible set of $(P3)$ is larger than the feasible set of (P_s), and the feasible set of $(P3)$ includes the feasible set of (P_s). Here, a *feasible set* is a space that solutions are possible (a space which satisfies the constraints). Thus, when our simplified model has an equilibrium point (x^*, p^*, u), this equilibrium point is also feasible in our original model. Moreover, since (P_s) maximizes the same utility function in (P_s), our simplified model maximizes the same utility functions in our original model. Thus, our original model also has the equilibrium point (x^*, p^*, u) in our simplified model. (Q.E.D.)

4.2 Equilibrium Conditions

In the following, we obtain conditions $(ASS1)$ - $(ASS7)$ under which our simplifed model (m sets of (P_s)) has an equilibrium point.

In order to discuss the equilibrium point of our simplified model, a solution toa cyber entity's utility maximization (namely $(P1)$) and a sloution to a Bio-net platform's utility maximization of our simplified model(namely (P_s)) must exist. In the following, we first give conditions so that solutions to $(P1)$ and (P_s) exist.

Theorem4.1. *There exists a solution to the following problem (let us call it a NLP problem) when a function $f(x)$ is $i)$ proper and the feasible set C of an optimization problem is $ii)$ nonempty and $iii)$ compact, [12]. Function f is proper if $f(x) > -\infty$ for at least one x. Also, a set is compact when the set is bounded and closed.*

$$\begin{aligned} \textit{maximize} \quad & f(x) \\ \textit{subject to} \quad & x \in C \end{aligned}$$

When $(P1)$ has the properties $i)$ - $iii)$, $(P1)$ has a solution. In the following, we examine if $(P1)$ has the properties $i)$ - $iii)$. It is easily seen that the feasible set of $(P1)$, namely $\beta^j \geq x_i^j \geq \gamma^j$ is nonempty and compact. Thus, conditions $ii)$ and $iii)$ hold. As for condition $i)$, U_i^j is not proper only when $U_i^j \equiv -\infty$. Utility function of $U_i^j \equiv -\infty$ is not realistic. Thus, the properness of utility functions holds. Therefore, $(P1)$ always have a solution.

In the following, we introduce several assumptions so that $(P2)$ satisfies all $i)$ - $iii)$ conditions explained above. We first note that the feasible set C_i of (P_s) is

$$\begin{aligned} C_i \quad = \{(p_i, x_i)| \quad & p^{max} \geq p_i \geq 0, \\ & \alpha_i \geq x_i = \textstyle\sum_{j=1}^n x_i^j \geq 0, \quad i = 1, \ldots, m \\ & x_i^j \in \operatorname{argmax}\{U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})|\beta^j \geq x_i^j \geq \gamma^j\} \\ & \quad \text{for all} \quad j = 1, \ldots, n, i = 1, \ldots, m\}. \end{aligned}$$

As for condition $i)$, we have already seen in the discussion of U_i^j being proper that properness of utility functions hold.

With respect to condition $ii)$, it is easily seen that the feasible set of (P_s) is nonempty if and only if, for any pairs of $(x_i^{-jloc}, p_i^{loc}, u_i^{loc})$, the following holds:

$$\left\{ \bigcup_{p_i=0}^{pmax} \left\{ \sum_{j=1}^n x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc}) \right\} \right\} \cap [0, \alpha_i] \neq \phi, \tag{4.4}$$

where $[a, b]$ indicates the interval from a to b. (4.4) implies that for a given set of local information $x_i^{-jloc}, p_i^{loc}, u_i^{loc}$, there exists a $0 \leq p_i \leq p^{max}$ such that the amount of resource that cyber entities consume on platform i will not exceed α_i. A sufficient condition for (4.4) is

$$(ASS1) \qquad x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})|_{p_i=p^{max}} = 0 \quad \text{for all} \quad j = 1, \ldots, n \quad \text{and} \quad i = 1, \ldots, m,$$

where $f(x)|_{x=a} = f(a)$. Here, note that p_i^{loc} includes p_i. $(ASS1)$ implies that when the resource prices of platform i are p^{max}, which is an extremely large value (See Assumption

(A4)), none of the cyber entities consume the platform's resource. This is because paying for such high cost resource is not reasonable for cyber entities, and cyber entities would not purchase such high cost resources. Note when the following holds, (4.4) holds. We can see this in the following procedure. By evaluating (4.4) at $p_i = p^{max}$ under condition ($ASS1$), we obtain $0 \cap [0, \alpha_i] = 0 \neq \phi$. Therefore, under condition ($ASS1$), (4.4) holds and thus, the feasible set of (P_s) is nonempty.

With respect to iii), the boundedness of the feasible set C_i directly follows from $p^{max} \geq p_i \geq 0$ and $\alpha_i \geq x_i \geq 0$. Furthermore, C_i is closed when U_i^j is continuous. Thus, we introduce the following assumption.

($ASS2$) U_i^j is continuous in x_i^j.

Concluding the discussion above, ($P1$) and (P_s) has a solution under newly introduced conditions ($ASS1$) and ($ASS2$).

Next, we will further introduce conditions, so that there exists an equilibrium point to our simplified model. In order to derive equilibrium conditions, we utilize Debreu's theorem [14, 10, 19]in the following.

Debreu's Theorem. *Consider a system consisting of n components, each of which maximizes its own utility function $f_i(x_i; x_{-i}) : D_i \rightarrow R$. Here, x_i is a decision that the ith component makes. Let $f_i(x_i; x_{-i})$ be a) continuous and b)concave in x_i [5] (See Fig. 5) for all $i = 1, \ldots, n$. Moreover, suppose D_i is c) compact and d) a convex set [6] (See Fig. 6) with e) nonempty interior. Then, this system has an equilibrium point.*

We use Debreu's theorem to derive equilibrium conditions for our simplified model. C_i and $\hat{U}_i$ in (P_s) corresponds to D_i and f_i in Debreu's theorem.

Since we have already shown that feasible set of (P_s), namely C_i, is c) compact and e) nonempty in the last section, we derive conditions under which C_i becomes a convex set. First, let us convert the feasible set C_i to a more compact form. To this end, we assume the following:

($ASS3$) $U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ is concave in x_i^j and continuously differentiable.

Under ($ASS3$), we may use the Kuhn-Tucker conditions [11]. The Kuhn-Tucker conditions is a well known result in nonlinear programming and provide necessary and sufficient

[5] A function f on D is *concave* relative to D, if for every choice of $x_0 \in D$ and $x_1 \in D$, the following holds.

$$f((1-\tau)x_0 + \tau x_1) \geq (1-\tau)f(x_0) + \tau f(x_1) \qquad \text{for all} \quad \tau \in (0,1).$$

[6] A set D is *convex*, if it includes, for every pair of points, the line segment that joins them. In other words, D is *convex* if, for every choice of $x_0 \in D$ and $x_1 \in D$, the following holds.

$$(1-\tau)x_0 + \tau x_1 \in D \qquad \text{for all} \quad \tau \in (0,1).$$

conditions for x_i^j to be a solution (See Apendix II). Using the Kuhn-Tucker conditions, x_i^{j*} is a solution to $(P1)$ for given $x_i^{-jloc}, p_i^{loc}, u_i^{loc}$, if and only if, there exists a vector (x_i^j, ζ_i, η_i) , which satisfies

$$\frac{\partial U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})}{\partial x_i^j} - \zeta_i + \eta_i = 0 \tag{4.5}$$

$$\zeta_i \geq 0, \eta_i \geq 0, \quad \zeta_i(\beta^j - x_i^j)^T = \eta_i(x_i^j - \gamma^j)^T = 0, \tag{4.6}$$

for all $j = 1, \ldots, n$ and $i = 1, \ldots, m$, where T indicates the transpose. In other words, (4.3) is equivalent to (4.5) and (4.6) when $(P1)$ has a solution. Thus, by replacing (4.3) with (4.5) and (4.6), (P_s) becomes

$(P4)$ maximize$_{p_i}$ $\hat{U}_i(p_i, x_i)$
subject to

$$p^{max} \geq p_i \geq 0$$

$$\alpha_i \geq x_i = \sum_{j=1}^n x_i^j \geq 0 \quad i = 1, \ldots, m$$

$$\frac{\partial U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})}{\partial x_i^j} - \zeta_i + \eta_i = 0 \quad \text{for all} \quad j = 1, \ldots, n, i = 1, \ldots, m$$

$$\zeta_i \geq 0, \eta_i \geq 0, \quad \zeta_i(\beta^j - x_i^j)^T = \eta_i(x_i^j - \gamma^j)^T = 0 \quad \text{for all} \quad j = 1, \ldots, n, i = 1, \ldots, m$$

Note that $(P4)$ is a maximization problem with three variables: x_i^j, ζ_i and η_i. By observing $(P4)$, we see that the feasible set has a non-complimentarity property, that is, the non-complementarity property, $\zeta_i(\beta^j - x_i^j) = \eta_i(x_i^j - \gamma^j) = 0$. It can be also seen from (Fig. 7) that these equations make the feasible set of $(P4)$ non-convex [7] . Thus, we first remove the non-complementarity property from $(P4)$ to make the feasible set of $(P4)$ a convex set. By adding the following condition, we can remove the non-complementarity property.

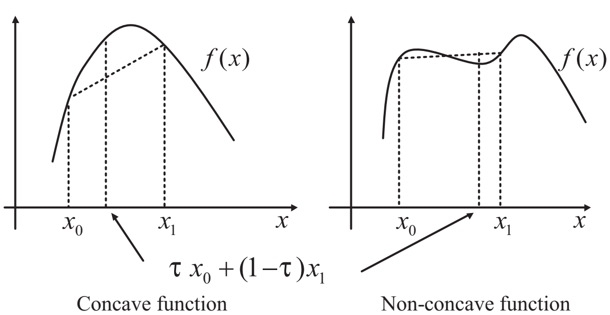

Figure 5: An example of concave function and non-concave function.

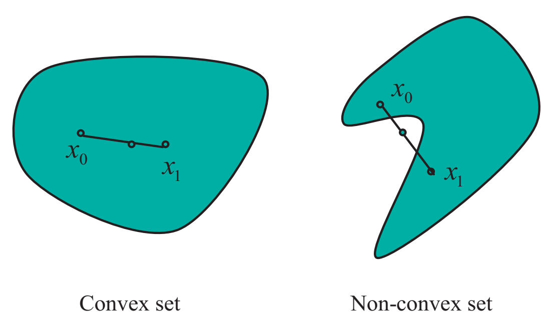

Figure 6: An example of convex set and non-convex set.

[7] Although there is a number of algorithms that have been developed for finding a global optimum of the problem with non-complementary constraints [16, 15, 20, 13], the problem still remains very difficult to solve. The most difficulty of the problem with non- complementary constraints is that the feasible set is non-convex.

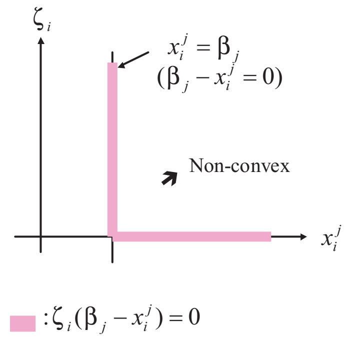

Figure 7: When the non-complementary property, $\zeta_i(\beta^j - x_i^j) = 0$ holds, either of the following equations holds: $zeta_i = 0$ or $(\beta^j - x_i^j) = 0$. Thus, by considreing $\zeta_i \geq 0$ and $(\beta^j - x_i^j) \geq 0$ in $(P4)$, we may see that the set $\{(\zeta_i, x_i^j) | \zeta_i(\beta^j - x_i^j) = 0, \zeta_i \geq 0, (\beta^j - x_i^j) \geq 0\}$ is non-convex.

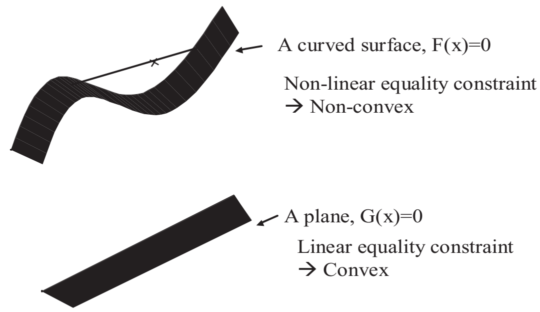

Figure 8: An equality constraint describes a surface. The upper part of the figure describes that a non-linear equality constraint makes the surface non-convex. The lower part of the figure describes that a linear equality constraint is a plane, and thus, the surface is convex.

$(ASS4)$ Let $\beta^j = +\infty$ and $\gamma^j = -\infty$ for $j = 1, \dots, n$.

When $(ASS4)$ holds, the constraint in (P_s), $\beta^j \geq x_i^j \geq \gamma^j$, always holds, and thus adding $(ASS4)$ is equivalent to removing the constraints $\beta^j \geq x_i^j \geq \gamma^j$ from (P_s). Note that by adding condtion $(ASS4)$, the compactness of the feasible set is lost in $(P1)$. Therefore, with compactness removed, $(P1)$ does not necessarily have a solution. As we have discussed in the begining of section 4.2, cyber entities need to have a solution to $(P1)$. Thus, in order that $(P1)$ still has a solution without compactness, let us utilize the following theorem.

Theorem4.2. *zThere exists a solution to a NLP (See Theorem 4.1) when a utility function is strongly concave. A function f on D is "strongly" concave relative to D if there exists a $\sigma > 0$ such that for every choice of $x_0 \in D$ and $x_1 \in D$, if the following holds.*

$$f((1-\tau)x_0 + \tau x_1) \geq (1-\tau)f(x_0) + \tau f(x_1) + \frac{1}{2}\sigma\tau(1-\tau) \parallel x_0 - x_1 \parallel^2 \qquad \text{for all} \quad \tau \in (0,1).$$

Thus, we add the following condition to the utility function of cyber entities so that $(P1)$ has a solution.

$(ASS5)$ $U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ is "strongly" concave in x_i^j. Now that we have seen that cyber entities are able to maximize their own utility functions (namely, a solution to $(P1)$ exists), we examine whether (P_s) has a solution or not.

By removing $\beta^j \geq x_i^j \geq \gamma^j$ from (P_s) (namely, by adding $(ASS4)$), (P_s) is converted to

$$
\begin{aligned}
(P5)\quad \text{maximize}_{p_i}\quad & \hat{U}_i(p_i; x_i) \\
\text{subject to}\quad & p^{max} \geq p_i \geq 0 \\
& \alpha_i \geq x_i = \sum_{j=1}^{n} x_i^j \geq 0 \qquad i = 1, \ldots, m \\
& x_i^j \in \text{argmax}\{U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})\} \qquad (4.7) \\
& \quad \text{for all} \quad j = 1, \ldots, n \quad i = 1, \ldots, m.
\end{aligned}
$$

By using the Kuhn-Tucker conditions to (4.7), we obtain

$$
\begin{aligned}
(P6)\quad \text{maximize}_{p_i}\quad & \hat{U}_i(p_i, x_i) \\
\text{subject to}\quad & p^{max} \geq p_i \geq 0 \\
& \alpha_i \geq x_i = \textstyle\sum_{j=1}^{n} x_i^j \geq 0 \qquad i = \ldots, m \\
& \tfrac{\partial U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})}{\partial x_i^j} = 0 \\
& \quad \text{for all} \quad j = 1, \ldots, n \quad i = 1, \ldots, m
\end{aligned}
$$

Although we have added $(ASS4)$ to (P_s) and removed complemantary property from (P_s), we still cannot claim that the feasible set to (P_s) (namely $(P6)$) is a convex set. In order to make the feasible set of $(P6)$ convex, we further add the following condition.

$(ASS6)$ $\frac{\partial U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})}{\partial x_i^j} = 0$ is linear in both x_i^j and p_i.

Concluding the discussion above, we have added assumptions $(ASS4)$ - $(ASS6)$ in order to make the feasible set of (P_s) convex. Now that we have obtained $c)$, $d)$ and $e)$ in Debreu's theorem, we now examine $a)$ and $b)$ in Debreu's Theorem. To this end, we assume the following.

$(ASS7)$ $\hat{U}_i(p_i, x_i(p_i, p_{-i}, u))$ is continuous and concave in p_i.

Now, all conditions $a)$ - $e)$ in Debreu's theorem holds. Using Debreu's theorem, we conclude that under assumptions $(ASS1)$ - $(ASS7)$, our simplified model has an equilibrium point for any given u. Thus, our original model also has an equlibrium point for any u.

Now, let us review the above discussion. We have derived the follwoing conditions to obtain an equilibrium point to our simplified model. In other words, under the conditions below, there exists a solution to the m sets of (P_s).

(ASS 1) $x_i^{j*}(x_i^{-jloc}, p_i^{loc}, u_i^{loc})|_{p_i = p^{max}} = 0$ for all $j = 1, \ldots, n$ and $i = 1, \ldots, m$

(ASS 2) U_i^j is continuous in x_i^j.

(ASS 3) $U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ is concave in x_i^j and continuously differentiable.

(ASS 4) Let $\beta^j = +\infty$ and $\gamma^j = -\infty$ for $j = 1, \ldots, n$.

(ASS 5) $U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})$ is "strongly" concave in x_i^j.

(ASS 6) $\frac{\partial U_i^j(x_i^j; x_i^{-jloc}, p_i^{loc}, u_i^{loc})}{\partial x_i^j} = 0$ is linear in both x_i^j and p_i.

(ASS 7) $\hat{U}_i(p_i, x_i(p_i, p_{-i}, u))$ is continuous and concave in p_i.

$(ASS1)$ - $(ASS6)$ are imposed to cyber entities. $(ASS7)$ is imposed to Bio-net platforms. Note that x_i depends on p_i (Refer to $(P2'')$), and this makes it difficult to obtain a function which satisfies $(ASS7)$. Examples of utility function U_i^j and $\hat{U}_i$, which satisfies all these conditions will be given in Remark4.1.

Remark4.1 Let us give an example of a function that satisfies conditions $(ASS1)$ - $(ASS7)$. For simplicity, let us consider the case where a network consists of only one platform with l resources. Furthermore, only one type of cyber entity exists. Let the utility function of a cyber entity be

$$U = k_1 u^T - xMp^T + k_2 x^T - xVx^T - pQp^T \tag{4.8}$$

where $x, p \in \Re_+^l$. $V : \Re^l \to \Re^l, Q : \Re^l \to \Re^l, M : \Re^l \to \Re^l$ are positive definite matrices [8] and $k_1 \in \Re_+^l, k_2 \in \Re_+^l$ are coefficient vectors. Furthermore, let us assume that

$$k_2 = p^{max} M^T \tag{4.9}$$

holds. The first term of (4.8) represents the energy that the cyber entity receives from user request packets. The second term indicates that when the cyber entity consume x resource, it must pay xMp^T amount of energy to the platform, and this reduces the cyber entity's utility. The third term implies that a cyber entity needs resource to provide its service to users, and thus, the utility increases as it acquires the amount of required resources. Note that when the cyber entity cannot acquire the amount of resource it needs, the efficiency ofproviding a service falls, and this reduces the utility of the cyber entity. However, a cyber entity do not need too much resource, and thus, the utility of a cyber entity decreases when it consumes too much resources. This is expressed in the forth term. Note that the term becomes significant when the cyber entity starts to consume a lot of resource. Lastly, the fifth term indicates that when the price of resource becomes too high, the utility of a cyber entity will decrease very much. Let us assume that a cyber entity determines the amount of resource it consumes by solving

$$\text{maximize}_x \qquad k_1 u^T - xMp^T + k_2 x^T - xVx^T - pQp^T \tag{4.10}$$

Next, let the utility function of platform i be

$$\hat{U}_i = xMp^T \tag{4.11}$$

[8] A matrix $V : \Re^l \to \Re^l$ is *positive definite* when yVy^T holds for any $0 \neq y \in \Re^l$

As we may see, the utility function of a Bio-net platform is equal to the second term of the utility function of a cyber entity, (4.8). The utility of a Bio-net platform increases as the energy income from cyber entities increase.

Thus, the platform determines its resource prices by maximizing

$$\begin{array}{ll} \text{maximize}_{p_i} & xMp^T \\ \text{subject to} & p^{max} \geq p_i \geq 0 \\ & \alpha_i \geq x_i = \sum_{j=1}^{n} x_i^j \geq 0 \\ & x_i^j \in \text{argmax}\{k_1 u^T - xMp^T + k_2 x^T - xVx^T - pQp^T\} \end{array}$$

Now, let us examine whether (4.8) and (4.11) satisfies conditions ($ASS1$) - ($ASS7$). It can be easily seen from (4.8) that conditions ($ASS2$) - ($ASS6$) hold. In the following, we will show that ($ASS1$) and ($ASS7$) also holds. To this end, let us first examine ($ASS1$). Since ($ASS2$) - ($ASS6$) holds, we can see that a solution to (4.10) exists. Thus, by using the Kuhn-Tucker conditions, we obtain

$$\frac{\partial U}{\partial x} = -Vx^{T*} - Mp^T + k_2^T = 0. \tag{4.12}$$

Therefore, the solution to (4.10) is

$$x^{T*} = V^{-1}(-Mp^T + k_2^T). \tag{4.13}$$

By evaluating (4.13) at $p = p^{max}$, we obtain $x^{T*} = 0$ from (4.9), and thus, ($ASS1$) holds.

Next, let us examine ($ASS7$). By substituting (4.13) to (4.11), we obtain

$$\begin{aligned} \hat{U}_i &= xMp^T && (4.14)\\ &= \left(V^{-1}(-Mp^T + k_2^T)\right)^T Mp^T \\ &= (-Mp^T + k_2^T)^T V^{-1} Mp^T \\ &= -pM^T V^{-1} Mp^T + k_2^T V^{-1} Mp^T && (4.15) \end{aligned}$$

Here, we may see that (4.15) is concave in p. Thus, (4.8) and (4.11) satisfies ($ASS1$) - ($ASS7$).

5 Simulation

To see that our Biologically-Inspired Architecture has an equilibrium point under the conditions for the existence of an equilibrium point in the simplified model, a simulation was designed and implemented. In concrete, we examine if our original model has an equilibrium point under conditions derived in our analysis, namely ($ASS1$) - ($ASS7$). In our simulation, we remove assumption ($A6$) do that a cyber entity changes the amount of resource it consumes discretely.

The major components of the simulation are described below.

5.1 Simulation Assumption

We assume the following network.

1. 20×20 Bio-net platforms are arranged in grid topology. (see Fig 9)
2. There are 150 user request packets on every network node (Bio-net platform).
3. There is a single type of cyber entity in the network ($n = 1$).
4. There is a single type of network resource in the network ($l = 1$).
5. Let all Bio-net platforms and cyber entities have information of other components that are within 1 hop. Namely, we set $N = 1$ in assumption ($A5$).
6. We initially set cyber entities on Bio-net platforms with probability 0.1.

We add Assumptions 4 and 5 so that we can easily verify our simulation results.

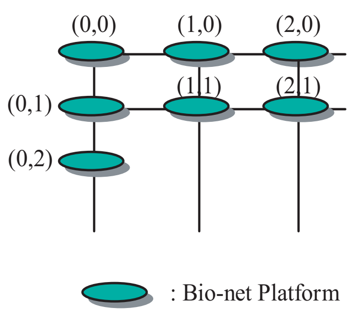

Figure 9: Network topology

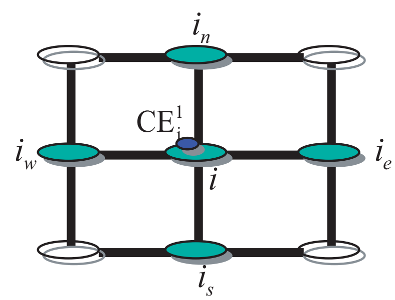

Figure 10: Neighboring platforms (North, East, South, West).

5.2 Simulator

In our simulation, we used the objective programming toolkit, Swarm 2.1.1 [21].

The simulator begins by instantiating the objects in the simulation. It then runs in a loop for 100 cycles. In each cycle, the simulator calls the Bio-net platforms first. After all the Bio-net platforms make their decision on the prices of their network resources by solving ($P5$), the simulator then calls the cyber entities in the entire network. All cyber entities decide the amount of network resource they consume by solving ($P1$), and as a result, they replicate, die and migrate. (The details are explained in section 5.2.2.) Lastly, the simulator calls the cyber entities and platforms, which locate on coordinates (12,12), (12,11), (12,13), (11,12) and (13,12) in order to track their behaviors. The results are given in the end of this paper.

5.2.1 users

As mentioned earlier, our purpose of this simulation is to study whether our architecture "always" has an equilibrium point or not. Note that this is equivalent to examine if our architecture has an equilibrium point for "any" user distribution over the entire network. Remind that user request packet vector u changes for different t and is also a given vector for all cyber entities and Bio-net platforms. During our simulation, opposed to the reality, we fix the user distribution (user's request packets) and see if our architecture accumulates to an equilibrium point after several cycles.

5.2.2 cyber entities

Since there is only one type of cyber entity and one type of network resource in our network, let $x_i \in \Re_+$ denote the amount of resource a cyber entity consumes on platform i. Also, let $U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc})$ denote the utility function of CE_i.

Cyber entities have three behaviors; replication, death and migration.

- Replication behavior: As mentioned in section 2, a cyber entity decides the amount of network resources it consumes. More explicitly, each cyber entity solves $(P1)$ simultaneously and repeatedly. When a cyber entity decides to consume more resource than that of the last iteration, the cyber entity *replicates*.

- Death behavior: Similar to the replication behavior, a cyber entity *dies* when it decides to consume less resource than that of the last iteration.

- Migration behavior: When a cyber entity migrates, it reduces the amount of resource it consumes from the platform it resides on and consumes network resource from the neighboring platform it decided to migrate to. Every cyber entity considers migrating to a neighboring platform, according to the following policy. In each iteration, given its own local information $x_i^{-loc}, p_i^{loc}, u_i^{loc}$, each cyber entity calculates its utility only concerning of the platform it resides on. Concretely, let $\bar{U}_i(x_i, p_i, u_i)$ denote the utility only concerning of platform i. Note that local information is not included in $\bar{U}_i(x_i, p_i, u_i)$. The function may be viewed as a utility function of cyber entities which do not have local information. (See p.19.) Next, each cyber entity calculates its utility concerning its neighboring platforms and the platform it resides on. If this takes a larger value than the utility only concerning of the platform it resides on, namely $\bar{U}_i(x_i, p_i, u_i)$, the cyber entity really migrates to a neighboring platform. More explicitly, cyber entity on platform i, namely CE_i, migrates to its neighboring platform i' when the following inequality holds.

$$\bar{U}_i(x_i - \Delta x; p_i, u_i) + \frac{\bar{U}_{i'}(x_{i'} + \Delta x; p_{i'}, u_{i'})}{x_{i'} + \Delta x} > \bar{U}_i(x_i; p_i, u_i) \tag{5.1}$$

where Δx is a small value. In each cycle, every cyber entity checks whether (5.1) holds for each of its neighboring platforms. A cyber entity migrates to all of the neighboring platforms where (5.1) holds in each iteration. During the simulation, when CE_i decides to migrate to 2 of its neighboring platforms, for example, CE_i decreases its consumption amount at platform i by $2\Delta x$ and consumes the network resource of its neighboring platforms which satisfy (5.1), each of them by Δx.

Now, let us explain the connections between $U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc})$ and $\bar{U}_i(x_i; p_i, u_i)$. To this end, let us introduce the notations first. Let us focus on CE_i. Furthermore, let i_n, i_e, i_s, i_w denote the platforms that can be reached within 1 hop from CE_i (See Fig.10). Here, the lower subscript represents north, east, south and west of platform i. Note that CE_i and platform i have the information of components, which locate on i_n, i_e, i_s and i_w.

Using these notations, $U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc})$ may be expressed using $\bar{U}_i(x_i; p_i, u_i)$ as the following.

$$\begin{aligned} U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc}) \quad = \quad & \bar{U}_i(x_i; p_i, u_i) - \nu_n \max\{0, \bar{U}_{i_n}(x_{i_n}; p_{i_n}, u_{i_n}) - \bar{U}_i(x_i; p_i, u_i)\} \\ & -\nu_e \max\{0, \bar{U}_{i_e}(x_{i_e}; p_{i_e}, u_{i_e}) - \bar{U}_i(x_i; p_i, u_i)\} \\ & -\nu_s \max\{0, \bar{U}_{i_s}(x_{i_s}; p_{i_s}, u_{i_s}) - \bar{U}_i(x_i; p_i, u_i)\} \\ & -\nu_w \max\{0, \bar{U}_{i_w}(x_{i_w}; p_{i_w}, u_{i_w}) - \bar{U}_i(x_i; p_i, u_i)\} \qquad (5.2) \end{aligned}$$

where $\nu_n, \nu_e, \nu_s, \nu_w > 0$. Note that all the terms starting with ν takes a negative value. More specifically, taking the second term for example, when

$$\bar{U}_{i_n}(x_{i_n}; p_{i_n}, u_{i_n}) > \bar{U}_i(x_i; p_i, u_i) \qquad (5.3)$$

holds,

$$\nu_n \max\{0, \bar{U}_{i_n}(x_{i_n}; p_{i_n}, u_{i_n}) - \bar{U}_i(x_i; p_i, u_i)\} > 0. \qquad (5.4)$$

We can see that when CE_i's utility only concerning of platform i, namely $\bar{U}_i(x_i; p_i, u_i)$ takes a smaller value than the utility only concerning of its neighboring platform, the utility of CE_i (namely $U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc})$) decreases. Thus, CE_i may be able to maximize its utility function $U_i(x_i; x_i^{-loc}, p_i^{loc}, u_i^{loc})$ by maximizing $\bar{U}_i(x_i; p_i, u_i)$ (through replication and death behavior) and also by migrating to a neighboring platform where (5.3) holds. This may be viewed as the following. When cyber entities on other platforms are happier than CE_i, CE_i may want to migrate to other platforms.

5.2.3 Bio-net platforms

Bio-net platforms try to solve $(P5)$. However, they do not know how to solve it directly, since they must consider the responce of cyber entities. Thus, let us assume that they try to maximize their utility gradually. We introduce two different policys on how Bio-net platforms solve $(P5)$, in simulation1 and 2.

5.3 Simulation 1

Let the utility function of a cyber entity on platform i be described as

$$\bar{U}_i(x_i, p_i, u_i) = k_1 u_i - k_2 px + ux - k_3 x_i^2 - k_4 p_i^2. \tag{5.5}$$

where k_1, k_2, k_3, k_4 are coefficients. Note that (5.5) is the same shape of (4.8). Note also that $u_i i = 1, \ldots, m$ is a constant value (150) throughout our simulation. The first term of (5.5) indicates the amount of energy CE_i receives from the request packets, u_i. The second term indicates the amount of energy CE_i pays to platform i. The third term implies that a cyber entity needs ux resource to provide u amount of users, and when the cyber entity cannot acquire ux amount of resource, the efficiency of providing a service falls, and this reduces the utility of CE_i. On the other hand, a cyber entity do not need too much resource, and thus, the utility of a cyber entity decreases when it consumes too much resources. This is expressed in the forth term. Lastly, the fifth term indicates that when the price of resource becomes too high, the utility of a cyber entity will decrease very much.

By substituting (5.5) to (5.2), we attain the utility function of CE_i. CE_i maximizes its utility function through replication, death and migration behavior explained in section 5.2.2.

5.3.1 Bio-net platform's policy

Let the utility function of a Bio-net platform i be described as

$$\hat{U}_i(p_i; x_i) = k_2 p_i x_i. \tag{5.6}$$

The right hand of (5.6) is the energy income of platform i. Note that (5.6) also follows the shape of remark4.1, and thus, the utility function is concave in p_i. Note also that the utility functions of both cyber entities and Bio-net platforms satisfy $(ASS1)$ - $(ASS7)$.

Let us assume that every Bio-net platform tries to solve $(P5)$ according to the following policy. In simulation 1, Bio-net platform i increases the price of its resource when its utility has increased compared to the last cycle. In other words, the Bio-net platform increases its resource prices when the energy income has increased compared to the last cycle. More explicitly, platform i increases the price of resource by Δ_p when the follwing equation holds.

$$\hat{U}_i\left(p_i(t); x_i(t)\right) > \hat{U}_i\left(p_i(t-1); x_i(t-1)\right) \tag{5.7}$$

holds, where t indicates the iteration number. When the opposite inequality holds, the platform decreases its resource prices.

Table 1: Coefficients

u_i(for all$i$)	α_i(for all$i$)	Δx	Δp	p^{max}	k_1	k_2	k_3	k_4
150	1000	1	0.1	10000	1	1	1	1

Table 2: Initial Settings

p_i(for all$i$)	x_i(for all$i$)
[0, 10]	[1, 10]

5.3.2 Initial Settings

Table 1 shows the coefficients set in our simulation. Table 2 shows initial settings (when $t = 0$) in our simulation. From Table 2, we set the price of resources in random numbers from 0 to 100.Similarly, we set the amount of resource a cyber entity consumes in random number from 0 to 10.

5.3.3 Simulation Results 1

Figure 11 shows the prices of network resources on platforms $(12, 12)$, $(12, 11)$, $(12, 13)$, $(11, 12)$, $(12, 11)$ and $(13, 12)$ during the simulation runs. Similarly, Figure 12 shows the amount of resource cyber entities consume. We may expect the number of all cyber entities and resource prices accumulate to a single equilibrium point. This is due to the following reasons. First, user's request packets are uniformly distributedin the network. Secondly, only one type of cyber entity is introduced in the network. Note that when cyber entities of the same type reside on the same platform, they maximize the same utility function, since their local information is the same. Lastly, only one type of utility function is introduced for Bio-net platforms. From Figure 11 and 12, we may easily see that our network system accumulates to an equilibrium point. Moreover, as we have expected, the number of all cyber entities and resource prices accumulate to a single equilibrium point.

5.4 Simulation 2

In simulation 2, we use the same utility functions of cyber entities and Bio-net platforms in simulation 1. Explicitly, cyber entities maximize (4.8) and Bio-net platforms maximize (4.11). The only thing that differs from simulation 1 is the way each platform maximizes its utility function. The policy used in this simulation is introduced below.

Table 3: Coefficients

u_i(for all$i$)	α_i(for all$i$)	Δx	Δp	p^{max}	k_1	k_2	k_3
150	1000	1	0.1	10000	1	10	1

5.4.1 Bio-net platform's policy

Let us focus on Bio-net platform i. Every Bio-net platform memorizes whether it has increased or decreased its resource price in cycle $t-1$. In the next cycle t, platform i repeats the same price adjustment (increase or decrese its price) when (5.7) holds. When the opposite inequality holds, the platform carries out the opposite price adjustment. This method may be viewed as follows. When the price adjustment of the last iteration has succeeded, then the platform repeats the same adjustment. Revesely, when the adjustment of the last iteration has failed, the platform makes the opposite adjustment.

Table 1 and 2 of simulation 1 are used in this simulation as well.

5.4.2 Simulation Results 2

The results are given in Figure 13 to 14. As we may see, the results are very simular to the ones given in Simulation 1.

5.5 Simulation 3

In simulation 1 and 2, we have simulated the case where $(ASS1)$ - $(ASS7)$ hold. In Simulation 3, we use a different utility function for Bio-net platforms, which do not satisfy the conditions derived in our analysis. Specifically, we set platform i's utility function as

$$\hat{U}_i(p_i; x_i) = (x_i + u_i)p_i. \tag{5.8}$$

Table 2 and 3 are used in this simulation.

5.5.1 Simulation Results 3

The results are given in Figure 15 to 16. From both of these figures, we may see that the prices of resource diverge.

6 Conclusions and Future Work

This paper has shown that under conditions $(ASS1)$ - $(ASS7)$, our network architecture accumulates to an equilibrium point. We have also demonstrated that an equilibrium point exists under $(ASS1)$ - $(ASS7)$. In simulation 3, we have seen that there are cases that an equilibrium point doesn't exist where ($ASS1$) - ($ASS7$) don't hold. However, as we might expect intuitively, the case is very rare and there are many cases that an equilibrium point exists even all the conditions ($ASS1$) - ($ASS7$) do not hold. Our future work is to relax our assumptions ($ASS1$) - ($ASS7$).

Acknowledgements

I would like to express my sincere appreciation to Professor Tastuya Suda of University of California Irvine for his continual guidance and invaluable suggestions to accomplish this thesis. He carefully read this thesis and commented in detail on the whole work in this thesis. He also gave me some chances to perform a talk on this paper at various companies, such as NTT, NTT Docomo, Fujitsu, Hitachi, and so on. Without his considerable help, none of the work could be completed. I am deeply indebted to Professor Masao Fukushima for his earnest guidance and an enthusiastic discussion. I respect him not only from the academic aspect, but also his personality. I am grateful to Associate Professor Tetsuya Takine for his significant advice. He is also the one who has introduced me to Professor Tatsuya Suda. I would also like to thank all the members in Professor Fukushima's Laboratory for cheering me up. Especially, I would like to show a great deal of appreciation to Mr Akihiro Enomoto for helping me write all the figures in this thesis, and all the support he has given to me. I am also very thankful to my fiance, who is also an Associate Professor Nobuo Yamashita for his considerable support and guidance. Finally, I would like to thank my parents for encouraging and supporting me all the time.

Appendix I

CE_i^j	the cyber entity of service type j, which reside on platform i
p_i	the vector of resource prices of platform i
p_{-i}	the vector of resource prices without platform i
p	the vector of resource prices of all platforms in the entire network
x_i^j	the vector of consumption amount of CE_i^j
x_i^{-j}	the vector of consumption amount without CE_i^j
x_i	the vector of a total amount of resource consumption on platform i
x_{-i}	the vector of consumption amount in the entire network except for platform i
x^j	the vector representing the amount of resources that cyber entity of type j consumes at different nodes
x	the vector of all cyber entities in the entire network
u_i^j	the number of request packets on platform i and for service j
u_i	the vector of request packets on platform i
u	the vector of all request packets in the entire network
$S_j(t)$	a set of nodes where cyber entities of type j resides at time t
$S_j^{loc}(t)$	a set of nodes, consisting of $S_j(t)$ and the nodes that are within N hops from a node in $S_j(t)$
$S(t)$	a set of nodes where at least one cyber entity resides at time t
$S^{loc}(t)$	a set of nodes consisting of nodes in $S(t)$ and the nodes that are within N hops from a node in $S(t)$
α_i	the maximum amount of resources that platform i may provide

Appendix II

The Kuhn-Tucker Conditions. *Consider*

$$\begin{array}{llll} (NLP) & \text{maximize} & f(x) & \\ & \text{subject to} & g_j(x) \geq 0, & j = 1, \ldots, r \\ & & h_j(x) = 0, & j = 1, \ldots, s, \end{array}$$

where the functions are $f : \Re^n \to \Re^1$, $g_j : \Re^n \to \Re^1$ *and* $h_j : \Re^n \to \Re^1$. *Suppose that for* (NLP), *the functions* f *and* $g_j, j = 1, \ldots, r$ *are concave and continuously differentiable and that* $h_j, j = 1, \ldots, s$ *are linear functions. (These assumptions make the feasible set of* (NLP) *convex.) The Kuhn-Tucker (K-T) conditions provide necessary and sufficient conditions for a point* x *to be optimal.*

The K-T conditions hold at x if there exist $\lambda = (\lambda_1, \ldots, \lambda_r) \in \Re^r$, and $\mu = (\mu_1, \ldots, \mu_s) \in \Re^s$, such that (x, λ, μ) satisfies

$$\begin{aligned} \nabla f(x)^\tau + \sum_{j=1}^{r} \lambda_j \nabla g_j(x)^\tau + \sum_{j=1}^{s} \mu_j \nabla h_j(x)^\tau &= 0 \\ \lambda_j \geq 0, \quad g_j(x) \geq 0, \qquad j &= 1, \ldots, r \\ \lambda_j g_j(x) = 0, \qquad j &= 1, \ldots, r \\ h_j(x) = 0, j &= 1, \ldots, s. \end{aligned}$$

where τ indicates the transpose. Note that λ_j must be non-negative, while μ_j may be positive, negative, or zero.

A *constraint qualification* to ensure that the constraint functions are well behaved is also needed, although in practice, it holds virtually always. Explicitly, the constraint qualification is as follows:

1. The feasible set has an interior point.

2. The vectors $\nabla h_j(x), j = 1, \ldots, s$ are linearly independent.

Simulation Results

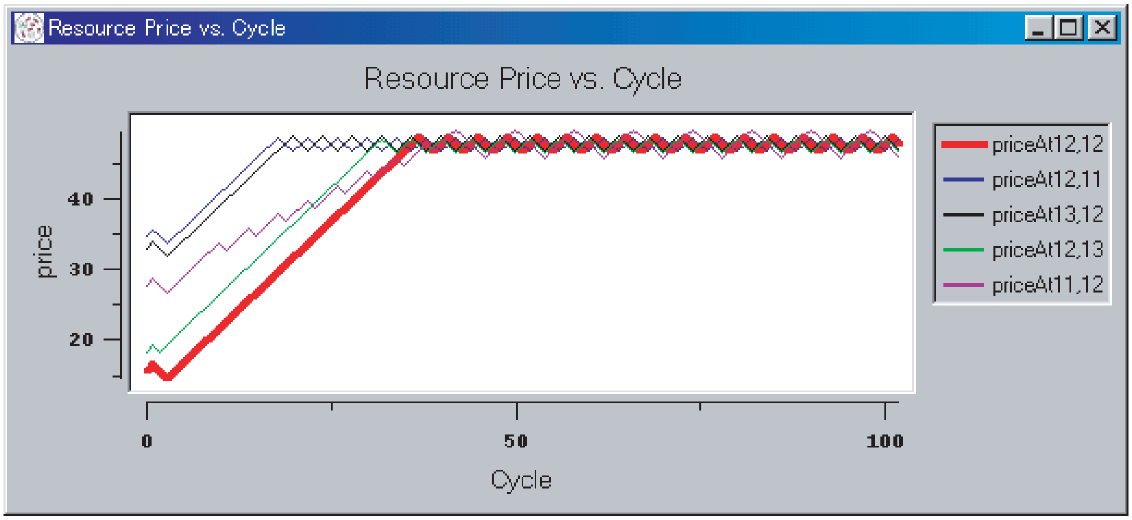

Figure 11: Simulation1

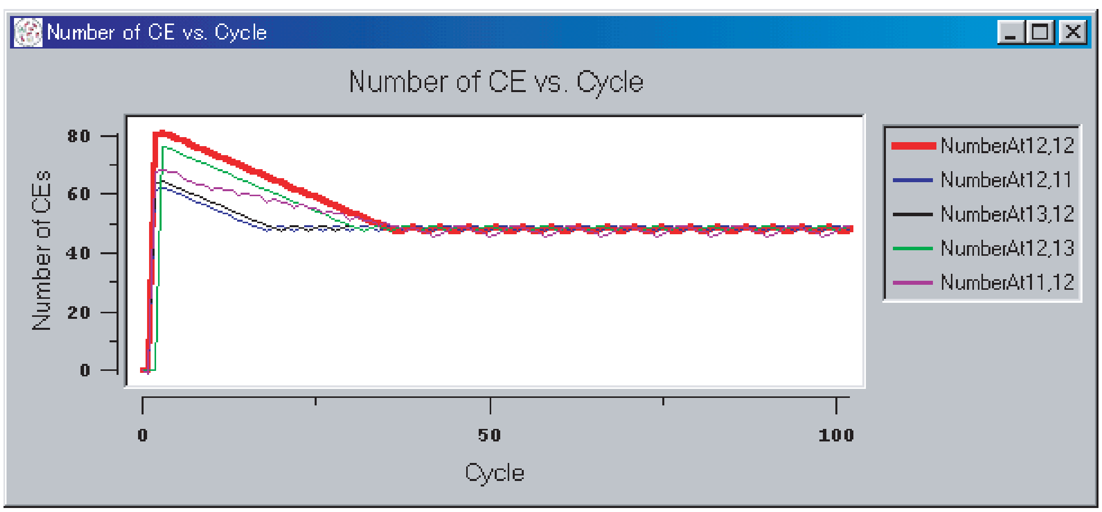

Figure 12: Simulation1-Number VS

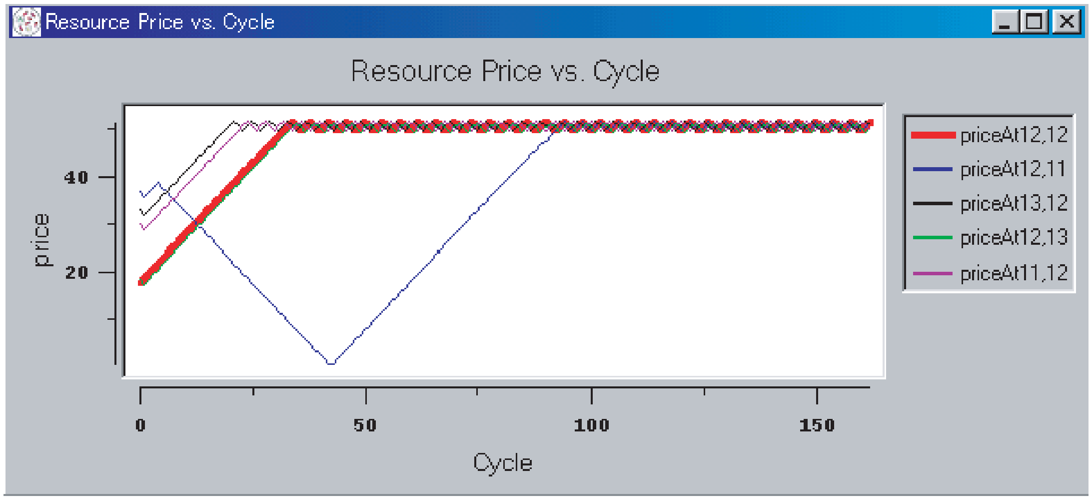

Figure 13: Simulation1-Number VS

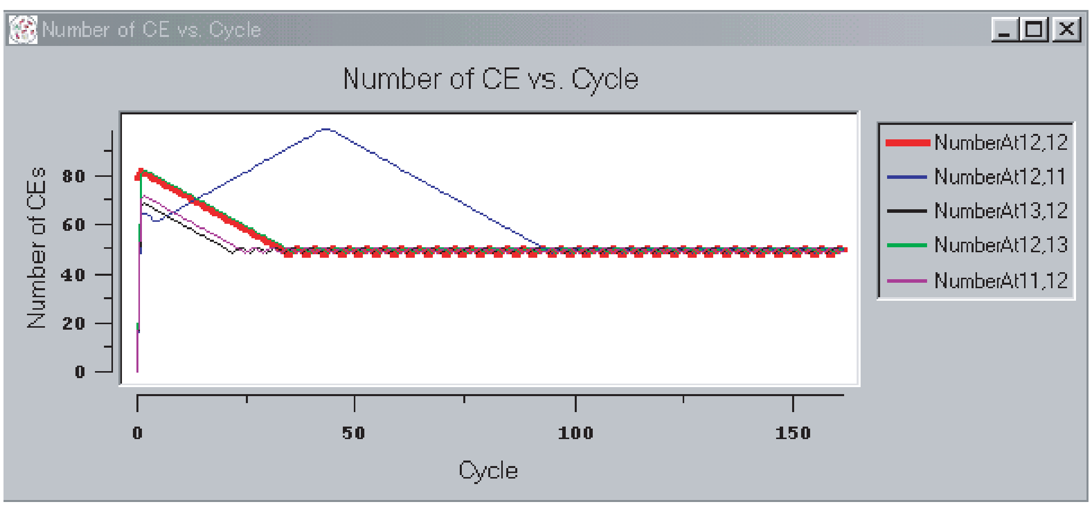

Figure 14: Simulation1-Number VS

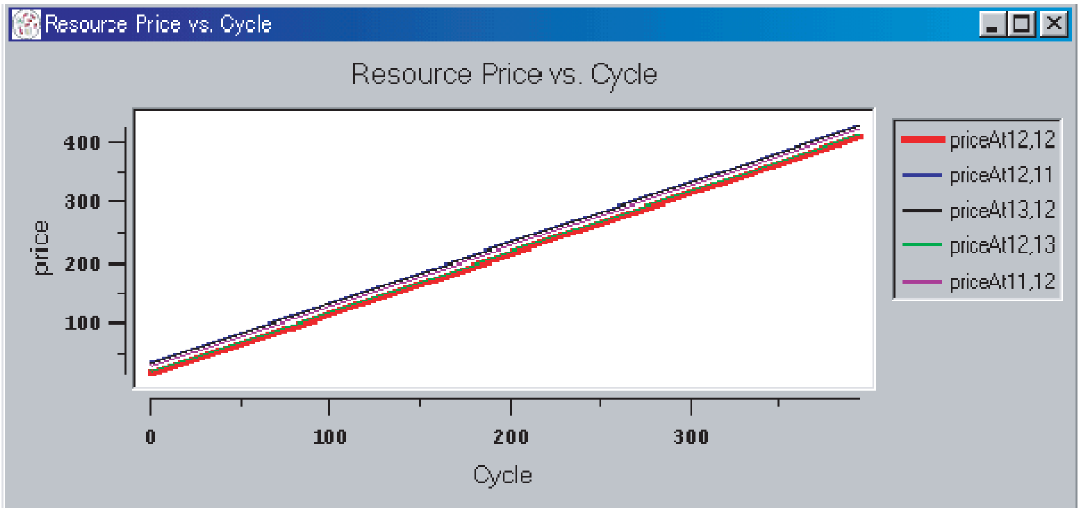

Figure 15: Simulation1-Number VS

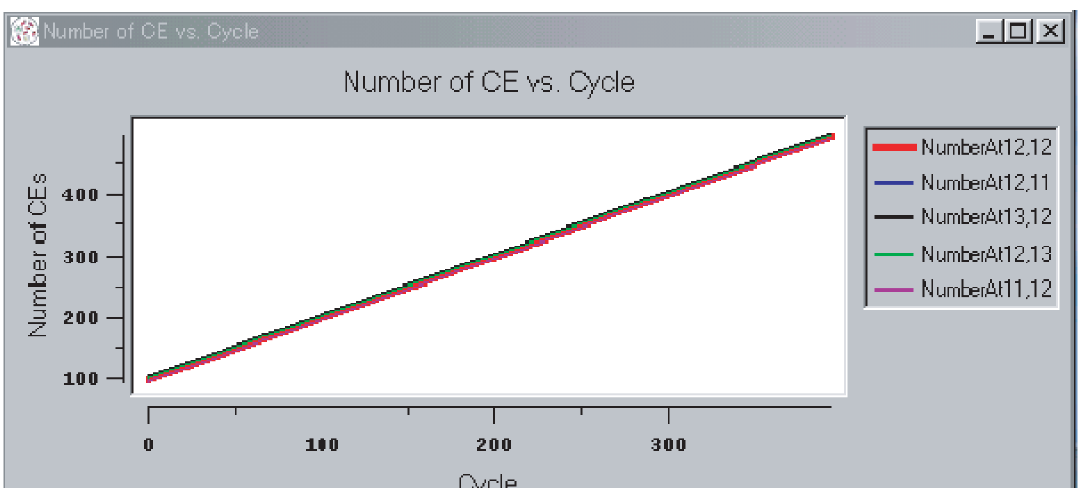

Figure 16: Simulation1-Number VS

References

[1] D.Alexandar, W.Arbaugh, A.Keromytis, and J.Smith, "A Security Active Network Environment Architecture," *IEEE Network Spetial Issue on Active and Programmable Networks.* vol 12, no 3. Pp 37-45, 1998.

[2] R.J.Ellison, D.A.Fisher, R.C.Linger, H.F.Lipson, T.Longstaff and N.R.Mead, "Survivable Network Systems: An Emerging Discpline". Carnegie Mellon University/Software Engineering Institute Technical Report, 1996.

[3] Akira Kimetame, Multi-agent and Complexity, Morikita publishing company, 1999. (in Japanese)

[4] Eric Bonabeau, Marco Dorigo, Guy Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems". Oxford University Press, 1999.

[5] M.Goeke, M.Sipper, D.Mange, A.Stauffer, E.Sanchez, M.Tomassini. "Online Autonomous Evolware." *Proceedings of the First International Conference on Evolvable Systems: from Biology to Hardware (ICES96).* T.Higuchi, M.Iwata and W.Liu (Editors), LCNS 1259, Springer-Verlag, 96-106, 1997.

[6] Michael Wang and Jim Fehlig, A Web Replication Application Based On The Bio-Networking Architecture, 1999.

[7] Peter N.Kugler, Gavan Lintern, Risk Management and the Evolution of Instability in Large-Scale, Industrial Systems Pp417-450.

[8] More Harchol-Balter, Tom Leighton, Daniel Lewin, Resource Discovery in Distributed Networks, 1999.

[9] Masahiro Okuno, Kotaro Suzuki, Micro Economics Book1, Iwanami, 1998. (in Japanese)

[10] W.I.Zangwill and C.B.Garcia, "Equlibrium Programming: The Path following Approach and Dynamics". Graduate School of Buisness, Chicago, 1980.

[11] Nonlinear Programming.

[12] R.Tyrrell Rockafellar, Roger J-B. Wets, "Variational Analysis", 1-37, 1998.

[13] Masao Fukushima, "Mathematical Programming with Equilibrium Constraints (MPEC)", Discreteness and Algorithm IV, Kindai Kagakusha, 1999. (in Japanese)

[14] G. Debreu, "A social equilibrium theorem", Proceedings of the National Academy of Science 38 (1952) 886-893.

[15] Z.Q.Luo, J.S.Pang and D.Ralph, "Mathematical Programs with Equilibrium Constraints", Cambridge University Press, Cambridge, 1996.

[16] K.Shimizu, Y.Ishizuka and J.F.Bard, "Nondifferentiable and Two-Level Mathematical Programming", Kluwer Academic Publishers, Boston, 1997.

[17] Akira Okada, Game Theory. (in Japanese)

[18] D.Fudenberg and J.Tirole, Game Theory, The MIT Press, Cambridge, Masachusetts, 1991.

[19] Kakutani,S., "A Generalization of Brouwer's Fixed Point Theorem," *Duke Mathematical Journal*, 457-59, 1941.

[20] J.J. Judice, A.M.Faustino, "Sequential LCP method for BLP", *Annals of Operations Research 34*, 89-106, 1992.

[21] swarm, SantaFeInstitute.

[22] Small World Phenomenon.