

# Distributed Algorithms to Solve the FOP Issues on the Weighted Convex-Split Networks

Shin-Jer Yang<sup>1</sup> and Tzu-Chi Guo<sup>1</sup>

<sup>1</sup> Dept. of Computer and Information Science, Soochow University, Taipei, Taiwan

E-mail: [sjyang@cis.scu.edu.tw](mailto:sjyang@cis.scu.edu.tw)

## Abstract

This paper discusses the Flow-Orientation Problem (FOP), which assigns orientations of all links of an undirected network to obtain a directed network for meeting some flow optimization measurements. First, we describe the background and define three related FOP issues: MDFOP, MDSFOP, and MDDFOP. We discuss the complexity of the FOP on general networks, and also list the essences of convex-split networks. These networks are to be either un-weighted or weighted. Then, we propose the algorithm for solving the MDFOP issue on weighted convex-split networks. Similarly, we can extend the MDFOP approach to solve the MDSFOP and MDDFOP issues. Suppose that  $\Gamma$  is any flow-orientation of a network  $N$ . Let  $\lambda(\Gamma)$  and  $\mu(\Gamma)$  be the maximum out-degree and the minimum out-degree, respectively, when  $N$  is un-weighted. In another, let  $\varepsilon(\Gamma) = \max_{x \in V} \{C(x) + \sum_{(z,x), z \rightarrow x} W(z,x)\}$ ,  $\pi(\Gamma) = \min_{x \in V} \{C(x) + \sum_{(z,x), z \rightarrow x} W(z,x)\}$ , and  $\theta(\Gamma) = \sum_{x,y \in V} \{(C(x) + \sum_{x \rightarrow z} W(x,z)) - (C(y) + \sum_{y \rightarrow z} W(y,z))\}$  when  $N$  is weighted. The main purpose of various FOP issues is to minimize  $\lambda(\Gamma) - \mu(\Gamma)$ ,  $\varepsilon(\Gamma) - \pi(\Gamma)$ ,  $\theta(\Gamma)$ . Finally, our findings can be applied to enhance performance of link flow and load balancing on networks.

**Keywords:** undirected network, directed network, FOP, convex-split network.

## 1. Introduction

The resource allocation and congestion controls are complex issues in a computer network, and also have been the subject of much study. The use of information services over networks, including the Internet, the private Intranet and even the Extranets, is supporting efficiency, promoting quality and enhancing throughput. However, we need to pose the essential problems of how a network effectively and fairly allocates its resources. In general, we consider the problem of routing connections with Quality of Service (QoS) requirements across network when the information available for making routing decisions is

inaccurate [5]. In other words, how to provide the accurate link information is a major task to the network routing. Therefore, it is important for all links in a network to select a route or link whose communication cost and weight are minimal. Many inexpensive heuristic algorithms have been proposed to meet two principal metrics of networking: throughput and delay [3, 4, 7]. Obviously, the authors investigate the dynamics of TCP traffic in an integrated service network that simultaneously supports both best-effort and controlled-load sessions [2]. Hence, they illustrated how the proposed service mechanism can be embedded within other packet and link-scheduling frameworks in a fully evolved integrated Internet services. In particular, each link within a network is effectively provided with a weighted fair of the buffers in the switch router. The switch router use peer-link accounting to determine which links are to be assigned an effective weight to each queue (flow). To better see how this implementation of fair queuing works, we consider the variation of fair queuing (FQ), called weighted fair queuing (WFQ). The goal of WFQ is to fairly allocate resource and efficiently enhance link utilization. In this situation, we will propose Flow-Orientation Problem (FOP) approach to perform WFQ for load balancing on the network. Similarly, we also propose the algorithms to solve the FOP on un-weighted and weighted convex-split networks, respectively.

The rest of the paper is organized as follows. In Section 2, we first define the various FOPs and describe their practical applications. In Section 3, a conjecture claims that the MDFOP is NP-hard is developed. Section 4 states the properties and essences of convex-split networks. In Section 5, we design an efficient algorithm to solve the MDFOP on weighted convex-split networks. Based on MDFOP solution, MDSFOP and MDDFOP can be handled similarly. Finally, we discuss the main issues of our proposed approaches and make conclusion in Section 6.

## 2. Backgrounds and The Variations of FOP

It is very important to implement WFQ that allows an effective weight to be assigned to each flow (queue) on convex-split and split networks. Formally, the problem to be considered is to find the maximum flow ( $\text{MAX}f_i$ ) of a set of links that satisfy the requirements and minimize the flow deviation ( $\rho_i$ ). Specifically, given a link  $i$  with capacity (bandwidth)  $b_i$  and weight  $w_i$ , the objective of WFQ is to minimize the flow deviation  $\rho_i$  for  $m$  links in a network as follows:

$$\rho_i = \min_{1 \leq i \leq m} \{ \text{MAX}f_i - (b_i \times w_i) \},$$

$$\text{where } \text{MAX}f_i = \max_{1 \leq i \leq m} \left\{ b_i - \left( w_i / \sum_{i=1}^m w_i \right) \right\}.$$

Note that a router performing WFQ must learn what weights to assign to each flow (queue) from every node on networks considered in this paper. However, it is assumed that it is possible to split the flow based on the actual weight of each link. In practice, not all link flow procedures permit this, but we can log the packets of all nodes on the routers. The ability to split flow greatly simplifies the problem. Although WFQ provides max-min fairness, protection, and the potential for specific (quantitative) QoS bounds, it is also relatively complex to implement with efficient link flow [1]. In practice, we can enhance performance and balance link flow of WFQ via assigning flow orientation of each link. This implies that we need to solve the FOP issues for turning an undirected network into a directed network. Basically, let  $N(V, L)$  be a network in which  $V$  is the set of  $n$  nodes and  $L$  is the set of  $m$  links. To achieve the required task, each link  $l = (u, v)$  can be assign a flow orientation either from  $u$  to  $v$  or from  $v$  to  $u$ . Each orientation may cause different serving costs. There are  $2^m$  possible total ways to assign the flow orientations of all links. Which one will be chosen? This paper proposes very useful and meaningful ways for selecting flow orientations of a network. The networks considered can be either unweighted or weighted. Suppose that  $\Gamma$  is any flow-orientation of a network  $N$ . Let  $\lambda(\Gamma)$  and  $\mu(\Gamma)$  be the maximum out-degree and the minimum out-degree, respectively, when  $N$  is unweighted. In another, let  $\varepsilon(\Gamma) = \max_{x \in V} \{ C(x) + \sum_{(z,x), z \rightarrow x} W(z,x) \}$ ,  $\pi(\Gamma) = \min_{x \in V} \{ C(x) + \sum_{(z,x), z \rightarrow x} W(z,x) \}$ , and  $\theta(\Gamma) = \sum_{x,y \in V} \{ (C(x) + \sum_{x \rightarrow z} W(x,z)) - (C(y) + \sum_{y \rightarrow z} W(y,z)) \}$  when  $N$  is weighted. Based on theoretical concept in [12], we address the related FOP issue and its applications in this paper. The first problem is called Min-Difference Flow-Orientation Problem (MDFOP) and it can be defined formally as follows:

**The Min-Difference Flow-Orientation Problem (The MDFOP):** Given a network  $N(V, L)$  in which

each node  $v$  is associated with a real cost  $C(v)$  and each link  $l$  is associated with a real weight  $W(l)$ , the task required is to assign a **flow orientation** for each link such that  $N$  becomes a directed network and  $\varepsilon(\Gamma) - \pi(\Gamma)$  is minimized. We denote that  $\mathfrak{N}(N)$  to be  $\min \{ \varepsilon(\Gamma) - \pi(\Gamma) \mid \Gamma \text{ is a flow-orientation scheme of } N \}$  hereafter.

Secondly, other important problems that are highly related to the MDFOP are defined as follows:

**The Min-Difference-Sum Flow-Orientation**

**Problem (The MDSFOP):** Given a network  $N(V, L)$  in which each node  $v$  is associated with a real cost  $C(v)$  and each link  $l$  is associated with a real weight  $W(l)$ , the task required is to obtain a flow-orientation scheme for all links such that  $\theta(\Gamma)$  is minimized. We denote that  $\mathfrak{S}(N)$  to be  $\min \{ \theta(\Gamma) \mid \Gamma \text{ is a flow-orientation scheme of } N \}$  hereafter.

To illustrate the essence and application of the MDFOP, a special form of it is introduced in the following.

**The Min-Degree-Difference Flow-Orientation**

**Problem (The MDDFOP):** Given an unweighted network  $N(V, L)$ , the objective is to derive a flow orientation scheme for all links such that  $\lambda(\Gamma) - \mu(\Gamma)$  is minimized. We denote that  $\mathfrak{Y}(N)$  to be  $\min \{ \lambda(\Gamma) - \mu(\Gamma) \mid \Gamma \text{ is a flow orientation scheme of } N \}$  hereafter.

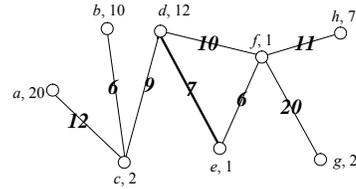


Figure 1. An input network  $N$  of MDFOP

An input network  $N$  with costs on nodes and weights on links of MDFOP is shown in Figure 1. Two flow-orientation schemes are depicted in Figure 2 and Figure 3, respectively.

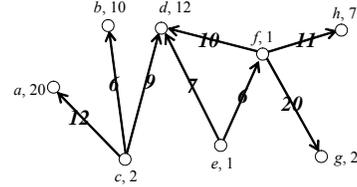


Figure 2. A flow-orientation scheme  $\Gamma_1$  of the network  $N$  in

Figure 1

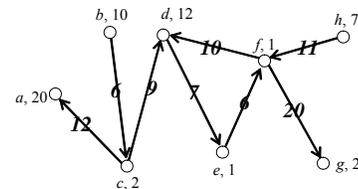


Figure 3. Another flow-orientation scheme  $\Gamma_2$  of the network  $N$  in Figure 1

For the flow-orientation scheme  $\Gamma_1$  illustrated in Figure 2,  $\varepsilon(\Gamma_1) = \max \{ (C(a) + \sum_{a \rightarrow v} W(a,v)) + (C(b) +$

$\sum_{b \rightarrow v} W(b, v) + (C(c) + \sum_{c \rightarrow v} W(c, v)) + (C(d) + \sum_{d \rightarrow v} W(d, v)) + (C(e) + \sum_{e \rightarrow v} W(e, v)) + (C(f) + \sum_{f \rightarrow v} W(f, v)) + (C(g) + \sum_{g \rightarrow v} W(g, v)) + (C(h) + \sum_{h \rightarrow v} W(h, v))\} = \max\{(20 + 0), (10 + 0), (2 + 12 + 6 + 9), (12 + 0), (1 + 7 + 6), (1 + 10 + 20 + 11), (2 + 0), (7 + 0)\} = 42$  and  $\pi(\Gamma_1) = \max\{(20 + 0), (10 + 0), (2 + 12 + 6 + 9), (12 + 0), (1 + 7 + 6), (1 + 10 + 20 + 11), (2 + 0), (7 + 0)\} = 2$ . Therefore,  $\varepsilon(\Gamma_1) - \pi(\Gamma_1) = 42 - 2 = 40$ .

It is easy to check that  $\varepsilon(\Gamma_2) - \pi(\Gamma_2) = 31 - 2 = 29$  for the flow-orientation scheme shown in Figure 3. So,  $\Gamma_2$  is “better” than  $\Gamma_1$  and the MDFOP is to find a flow-orientation scheme  $\Gamma$  such that  $\varepsilon(\Gamma) - \pi(\Gamma)$  is minimized.

### 3. Complexity Issues of the MDFOP on General Networks

This section considers and analyzes the complexity of the MDFOP on the general networks. Some useful and essential properties of complexity for the MDFOP are derived from a decision problem concept, which is corresponding to the Out-Degree Only LOP is introduced [11].

**The Constant MDFOP:** Given a network  $N(V, L)$  in which each node  $v$  is associated with a real cost  $C(v)$  and each link  $l$  is associated with a real weight  $W(l)$ , the task required is to determine whether there exists a flow orientation such that  $\varepsilon(\Gamma) - \pi(\Gamma) \leq k$ , where  $k$  is a given real constant.

**Lemma 1:** The Constant MDFOP belongs to the class of NP problems.

*Proof:*

We will show that the Constant MDFOP belongs to the NP class of problems by designing a NP algorithm for it as follows:

**Algorithm NP\_Constant\_MDFOP**

**Input:**

An undirected weighted network  $N(V, L)$  and a real constant  $k$ .

**Output:**

‘YES’ if there exists a flow orientation  $\Gamma$  such that  $\varepsilon(\Gamma) - \pi(\Gamma) \leq k$ , and ‘NO’, otherwise.

**Guessing Phase:**

```

for  $j \leftarrow 1$  to  $m$  /* assume that  $e_j = (u_j, v_j)$  */
     $e_j = \text{choice}(\{0, 1\})$ ;
    /*  $e_j = 1$  iff  $u_j \rightarrow v_j$  is selected */
    /*  $e_j = 0$  iff  $v_j \rightarrow u_j$  is selected */

```

**endfor**

**Checking Phase:**

if  $\varepsilon(\Gamma) - \pi(\Gamma) \leq k$  return ‘YES’ else return ‘NO’;

**End NP\_Constant\_MDFOP**

It is a simple task to show that the Checking Phase of the above algorithm can be completed in polynomial time with respect to  $n$  and  $m$  in which  $n$  is the number of nodes and  $m$  is the number of links of

the input network. This finding means that the Constant MDFOP belongs to the NP class of problems.

**Lemma 2:** The Constant MDFOP can be reduced to MDFOP in polynomial time.

**Conjecture 1:** The MDFOP is NP-hard on weighted networks.

## 4. Properties and Essences of Convex-Split Networks

A network  $N$  is called *complete* if there exists a link connecting every pair of nodes  $x$  and  $y$ . Complete network is an important class of networks in many real network designs and topologies. This paper addresses one of its supersets, the class of convex-split networks. A node-set  $H$  of any network is called an *independent set* if no link exists joining any two nodes in  $H$ . Generally speaking, a network  $N(V, L)$  is called a *split network* shown in Figure 4 if the node-set  $V$  consists of two disjoint sets  $K$  and  $I$  such that  $K$  is a complete network and  $I$  forms an independent set [10].

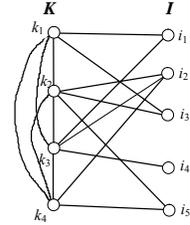


Figure 4. A split network

Given any network  $N$ , the distribution of the neighborhoods of all nodes is a very important consideration to many real-world applications [7, 9]. This paper restricts attention to the split networks with convex property, which are called convex-split networks hereafter. The simplest convex-split networks are complete split networks. A split network is called a *complete split network* if  $(k, i) \in L$ , for all  $k \in K$  and  $i \in I$ . Figure 5 shows a complete split network. In addition, for any node  $x$  of  $N$ , the *neighbors* of  $x$ , denoted by  $\text{Nbs}(x)$ , is the set of nodes adjacent to  $x$ . Suppose that  $N(V = K \cup I, L)$  is a split network, where  $K = \{k_1, \dots, k_m\}$  and  $I = \{i_1, \dots, i_n\}$ . Define  $\text{Nbs}(K, x) = \text{Nbs}(x) \cap I$ , for each node  $x \in K$  and  $\text{Nbs}(I, y) = \text{Nbs}(y) \cap K$ , for each node  $y \in I$ . The split network  $N$  is called a *convex-split network* shown in Figure 6 if  $\text{Nbs}(K, x) = \{i_p, \dots, i_q\}$  such that  $p, \dots, q$  are  $q - p + 1$  consecutive numbers, for each node  $x \in K$ .

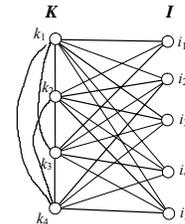


Figure 5. A complete split network

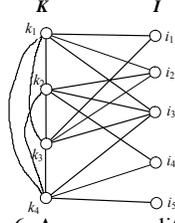


Figure 6. A convex-split network

## 5. The Complexity of FOPs on Weighted Convex-Split Networks

In this section, we first describe the definition for MDFOP. Then, we discuss and solve the MDFOP issues on the weighted convex-split network. Accordingly, the MDSFOP and MDDFOP are handled similarly, then they can experience same reasons as algorithm `MDFOP_on_Convex_Split_CSm,n`. In general, let  $N(V = K \cup I, L)$  be a convex-split network, where  $K = \{k_1, \dots, k_m\}$  and  $I = \{i_1, \dots, i_n\}$ .

**Definition 1.** Suppose that  $\text{Nbs}(K, k_j) = \{i_s, \dots, i_t\}$  and  $\text{Nbs}(K, k_{j+1}) = \{i_p, \dots, i_q\}$ , for each  $j$ .  $K$  and  $I$  are called a *convex ordering* of  $N$  if one of the following properties hold.

- $\text{Nbs}(K, k_j) \supset \text{Nbs}(K, k_{j+1})$  iff  $s \leq p$  and  $t > q$ .
- $\text{Nbs}(K, k_j) \subset \text{Nbs}(K, k_{j+1})$  iff  $s \leq p$  and  $t < q$ .
- $\text{Nbs}(K, k_j) \cap \text{Nbs}(K, k_{j+1}) = \emptyset$  iff  $t < p$ .

The three possibilities in Definition 1 are described in Figure 7(a), (b), and (c), respectively.

Let  $\mathfrak{N}(u_1)$  and  $\mathfrak{N}(u_2)$  be the first two maximums among  $\mathfrak{N}(k_2), \dots, \mathfrak{N}(k_m), \mathfrak{N}(i_1), \dots, \mathfrak{N}(i_n)$ . The possibilities that should be considered are listed in Figures 8, 9, and 10, respectively, where  $\mathfrak{N}(v)$  is to be  $\min\{\varepsilon(\Gamma) - \pi(\Gamma) \mid \Gamma \text{ is a flow-orientation scheme of any node } v \in K, I\}$ . The following algorithm can be designed accordingly.

**Algorithm `MDFOP_on_Convex_Split_CSm,n`**

**Input:** A weighted convex-split network  $\text{CS}_{m,n} = (K \cup I, L)$  with  $K = \{k_1, \dots, k_m\}$  and  $I = \{i_1, \dots, i_n\}$ .

**Output:** A flow-orientation scheme of  $\text{CS}_{m,n}$  such that the value  $\varepsilon(\Gamma) - \pi(\Gamma)$  is minimized.

**Method:**

```

{
  for j ← 1 to m
     $\mathfrak{N}(k_j) \leftarrow C(k_j)$ ;
     $\text{outdeg}(k_j) \leftarrow 0$ ;
  endfor
  for q ← 1 to n
     $\mathfrak{N}(i_q) \leftarrow C(i_q)$ ;
     $\text{outdeg}(i_q) \leftarrow 0$ ;
  endfor
  if (m == 1) then
    recursively solve the MDFOP on  $\text{CS}_{1,n}$ ;
    return;
  endif
}

```

```

}
recursively solve the MDFOP on  $\text{CS}_{m-1,n}$ ;
let  $\mathfrak{N}(u_1)$  and  $\mathfrak{N}(u_2)$  be the first two maximums
among  $\mathfrak{N}(k_2), \dots, \mathfrak{N}(k_m), \mathfrak{N}(i_1), \dots, \mathfrak{N}(i_n)$ ;
to determine the orientations of the links  $(k_i, i_s), \dots, (k_i, i_t)$ 
based on the minimal value of  $\varepsilon(\Gamma) - \pi(\Gamma)$ ;
return;
}

```

**End `MDFOP_on_Convex_Split_CSm,n`**

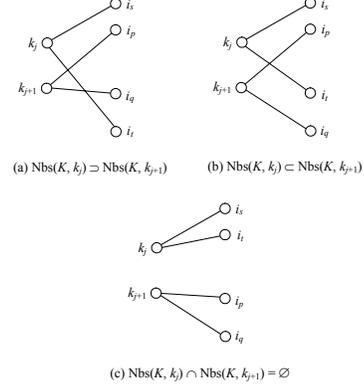


Figure 7. Relationship between  $\text{Nbs}(K, k_j)$  and  $\text{Nbs}(K, k_{j+1})$  in a convex ordering

The time-complexity of the **Algorithm `MDFOP_on_Convex_Split_CSm,n`** can be analyzed as follows, where  $T_m^n$  denotes the time-complexity for solving the MDFOP on  $\text{CS}_{m,n}$ .

- The first two for loops can be done in  $O(m)$  and  $O(n)$  time, respectively.
- The MDFOP on  $\text{CS}_{1,n}$  can be easily solved in  $O(n)$  time.
- The time required for recursively solving the MDFOP on  $\text{CS}_{m-1,n}$  is  $T_{m-1}^n$ .
- Finding  $\mathfrak{N}(u_1)$  and  $\mathfrak{N}(u_2)$  can be performed in  $O(m + n)$  time.
- Determining the orientations of the links  $(k_i, i_s), \dots, (k_i, i_t)$  can be achieved in  $O(3(m + n))$  by considering the three possibilities as shown from Figure 8 to Figure 10.

**Theorem 3:** The MDFOP can be solved in  $O(mn)$  for any weighted convex-split network  $\text{CS}_{m,n}$ .

*Proof:* Based on the above reasoning, the following recurrence relation can be derived.

$$T_m^n = T_{m-1}^n + O(3(m + n)) + O(m) + O(n)$$

It is easy to derive that  $T_m^n = O(m * n)$  from the above recurrence relation.

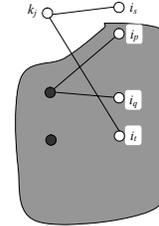


Figure 8.  $u_1$  and  $u_2$  belong to  $K$ .

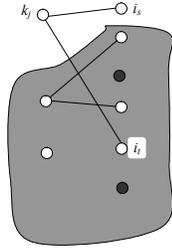


Figure 9.  $u_1$  and  $u_2$  belong to  $I$ .

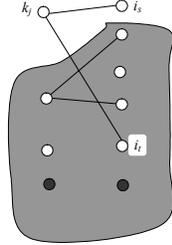


Figure 10.  $u_1$  belong to  $K$  and  $u_2$  belong to  $I$

## 6. Conclusions

Usually, most networks provide congestion-control and resource allocation mechanisms to deal with traffic workload situation. But, how to allocate resources effectively and fairly among a collection of active nodes is a key issue on the network. The resources being shared include link bandwidth and queue buffers on the routers or switches where packets are queued awaiting transmissions. Each router must implement some queuing discipline that governs how packets are buffered while awaiting to be transmitted. Although, Lyengar, Challenger, Dias, and Dantzig presented techniques for designing Web sites that need to handle large request volumes and to provide high availability, they should consider each link flow's capacity [6]. The link queuing algorithm can be thought of as both bandwidth and buffer space [8]. Hence, how to implement the fair queuing (FQ) in a network is key design issue to allocate its resources effectively. Based on BSGP approach, we developed the FOP algorithm that it can be applied to balance the link flow and queue buffer of a variation of fair queuing, called weighted fair queuing (WFQ).

In this paper, the main objectives are to solve the various FOP issues to minimize the maximum out-degree (unweighted version), and also to minimize the maximum sum of nodes and out-degree (weighted version). Meanwhile, we propose the efficient algorithms to minimize  $\lambda(\Gamma) - \mu(\Gamma)$ ,  $\varepsilon(\Gamma) - \pi(\Gamma)$ ,  $\theta(\Gamma)$  for various FOP issues on the weighted convex-split networks. Finally, our findings are applicable to a broader problem of dynamic load balancing on the networks, which meet performance improvement and flow efficiency at minimal communication cost.

## 7. References

- [1] S. N. Bhatt. and J. Crowcroft, "QoS Sensitive Flows: Issues in IP packet Handling," *IEEE Internet Computing Journal*, Vol. 4, No. 4, pp. 48-57, July/August 2000.
- [2] W. C. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "Understanding and Improving TCP Performance over Networks with Minimum Rate Guarantees," *IEEE/ACM Transactions on Networking Journal*, Vol. 7, No. 2, pp. 173-187, April 1999.
- [3] X. Jia, "A Distributed Algorithm of Delay Bounded Multicast Routing for Multimedia Applications in Wide Area Networks," *IEEE/ACM Transactions on Networking Journal*, Vol. 6, No. 6, pp. 828-837.
- [4] W. Korfhage, "Dynamically Load Balancing Prioritized Processes in Distributed Systems," *ISCA Journal*, Vol. 2, No. 3, pp. 137-144, Dec. 1995.
- [5] H. Lorenz and A. Orda, "QoS Routing in Networks with Uncertain Parameters," *IEEE/ACM Transactions on Networking Journal*, Vol. 6, No. 6, pp. 768-778, Dec. 1998.
- [6] A. Lyengar, J. Challenger, D. Dias, and P. Dantzig, "High-Performance Web Site Design Techniques," *IEEE Internet Computing Journal*, Vol. 4, No. 2, pp. 17-26, March/April 2000.
- [7] D. L. Mills, "Adaptive Hybrid Clock Discipline Algorithm for the Network Time Protocol," *IEEE/ACM Transactions on Networking Journal*, Vol. 6, No. 5, October 1998.
- [8] L. L. Peterson and B. S. Davie, "Computer Networks: A Systems Approach," Morgan Kaufmann Publishers, CA. USA, 2<sup>nd</sup> Ed., 2000.
- [9] T. Tsichiya and T. Kikuno, "Dependable Evaluation of the Weighted Voting System in the Presence of Node and Link Facilities," *International Journal of Computer Systems Science & Engineering*, Vol. 15, No. 3, pp. 181-190, May 2000.
- [10] William C. K. Yen and C. Y. Tang, "The Searchlight Guarding Problem on Weighted Split Graphs and Weighted Cographs", *Networks*, Vol. 35, No. 4, pp. 195-206, 2000.
- [11] William C. K. Yen and S. J. Yang, "The Link-Orientation on Weighted Networks," In *Proceedings of the 16<sup>th</sup> International Conference on Computer and Their Applications*, Seattle, Washington USA, April 2001, pp. 325-329.
- [12] Shin-Jer Yang, "The Flow-Orientation Problems on the Weighted Real Networks," In *Proceedings of the 7<sup>th</sup> Joint Conference on Information Sciences*, North Carolina U.S.A., Sept. 2003, pp. 538-542.