Journal of Science and Technology, Vol. 32 (2012), pp 93- 102 93

©2012 Kwame Nkrumah University of Science and Technology (KNUST)

http://dx.doi.org/10.4314/just.v32i3.10

RESEARCH PAPER

A NOTE ON THE ART OF NETWORK DESIGN PROBLEMS

A. A. Osagiede,[1] V. U. Ekhosuehi,[1] O. D. Germany[1] and C. O. Todo[2]

[1]*Department of Mathematics, University of Benin, P.M.B. 1154, Benin City, Nigeria*

Augustine.osagiede@uniben.edu

[2]*Department of Statistics, Delta State Polytechnic, Otefe, P.M.B. 03 Oghara, Delta State, Nigeria*

ABSTRACT

In this study, we describe some Network Design Problems (NDPs) as well as the network flow-based improvement algorithm for neighbourhood search defined by cycles. The main part of the study is structured around the formulation of the expected duration of stay in the educational system as a NDP. The fundamental matrix of the absorbing Markov chain is employed in computing the expected duration of each flow in the system. We shall illustrate the new graph-theoretic formulation for the educational system using datasets from a university setting. The paper concludes with suggestions for future directions of research.

Keywords: absorbing Markov chain; educational system; graph theory; network design.

INTRODUCTION

Network Design Problems (NDPs) require the study of graph theory. A graph G consists of a set of vertices and edges (or arcs). Generally, a graph G is denoted as $G=(V,E\cup A)$, where V is a set of vertices, E is a set of edges and A is a set of arcs or directed edges (Amponsah *et al.*, 2010).

A graph G is said to be directed if $V\neq\phi$, $E=\phi$, and $A\neq\phi$, where ϕ is the empty set. Conversely, when $V\neq\phi$, $E\neq\phi$, and $A=\phi$, the graph G is said to be an undirected graph. However, whenever $V\neq\phi, E\neq\phi$, and $A\neq\phi$, the graph G is said to be a multi-graph.

A network (N,A) is said to be simple, weighted digraph G if it consists of a finite set of nodes N and a set of arcs $A\subseteq N\times N$. A path in (N,A) is a sequence $(\alpha_1,\ldots,\alpha_n)$ of $n\ (n\geq 1)$ distinct arcs having, for $m=1,2,\ldots,n$, arc $\alpha_m\in A$ and either $\alpha_m=(i_m,i_{m+1})$ or $\alpha_m=(i_{m+1},i_m)$ (Hassin, 1981a).

Let $G=(N,A)$ be a directed network with a positive integer capacity c_{ij} on every arc $(i,j)\in A$. A flow F is a function $F:A\to R$,which assigns each directed arc (i,j) a nonnegative number x_{ij} such that:

$$\sum_{\{j:(j,i)\in A\}} x_{ji} = \sum_{\{j:(j,i)\in A\}} x_{ij} \qquad \forall\ i\in N-\{s,t\}, \quad (1)$$

$$\sum_{\{j:(j,t)\in A\}} x_{jt} = \rho, \quad (2)$$

$$x_{ij} \leq c_{ij} \qquad \forall (i,j) \in A, \qquad (3)$$

for some $\rho \geq 0$, where s and t are the source and sink nodes respectively. The property expressed by equation (1) is called conservation of flow. If, however, the function $F: A \to R$ satisfies equations (2) and (3) then equation (1) becomes:

$$\sum_{\{j:(j,i)\in A\}} x_{ji} - \sum_{\{j:(j,i)\in A\}} x_{ij} \geq 0, \qquad (4)$$

then F is a preflow. The excess for each node $i \in N - \{s,t\}$ in the preflow F is

$$e_i = \sum_{\{j:(j,i)\in A\}} x_{ji} - \sum_{\{j:(j,i)\in A\}} x_{ij}. \qquad (5)$$

If $e_i = 0, i \in N,$ in an assignment of flow values x_{ij} to the arc $(i, j) \in A$, then we have a circulation. Various forms of circulation models have been studied (Hassin, 1981a).

This paper seeks to expose network design problems. We strongly feel that wide exposure to, and continued awareness of, the diversity of network design problems can help increase the network analyst's range of choice among methods to formulate network-based problems. We shall also illustrate how transitions in the educational system can be viewed as a network design.

RELATED WORKS

Networks arise in numerous fields of human endeavour. Ahuja (1982) presented the design and analysis of computer communication networks.

Borndorfer (2008) discussed planning problems in public transit using network optimization models.

Hillier and Lieberman (2005) enumerated areas where network representations are employed to include: production, distribution, project planning, facilities location, resource management and financial planning.

In the planning, scheduling and control of projects, Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) have been employed for activities with deterministic durations and probabilistic durations, respectively. CPM and PERT are network-based methods designed to provide analytic means for scheduling activities of a project (Taha, 2002).

Osagiede and Ekhosuehi (2009) determined the optimum travelling cost to several schools with emphasis on the most preferred travelling mode for a household using network design.

Ahuja *et al.* (2003) presented neighbourhood search defined by cycles as well as the application of cyclic exchange neighbourhood search to airline fleet assignment, multicommodity flow and minimum cost flow problems.

Many network optimization problems of practical interest are NP-hard, that is, the existence of an algorithm that finds an optimal solution in polynomial time is very unlikely. For this reason, nearly optimal solutions (within a reasonable computational time) to such NP-hard problems are obtained by employing heuristic (approximation) algorithms. There are also improvement algorithms. An improvement algorithm is a heuristic algorithm that generally starts with a feasible solution and iteratively tries to obtain a better solution. Hassin and Tamir (1986), for instance, mentioned that problems that are NP-hard on general graphs can be solved efficiently on trees by polynomial algorithms. Such polynomial algorithms rely on the existence of an efficient construction that recursively generates larger components from previous ones, and terminate with a given tree.

In the remaining part of this section, we shall discuss some network design problems including NP-hard problems under various sub-headings and propose a network optimization problem for the educational system.

The maximum flow problem

The maximum flow problem is one of the most fundamental problems in network flow theory and has been investigated extensively. Hassin (1981b) proved that a maximum flow function F can be constructed as follows: For each edge $(i, j) \in E$, let $(i', j') \in E'$ be the dual edge associated with it. Let $F(i, j) = u(j') - u(i')$, where $u(v)$ is the length of a shortest path from s' to v, for every $v \in V'$, where s' is the source of the dual graph $G^d = (V', E')$ of G. Then a tree of shortest paths rooted at s', which can be found in $o(n \log n)$ time, defines not only a minimum cut but also a complete (maximum) flow function.

Basically, the maximum flow problem is to determine a flow F for which ρ in equation (2) is maximized. Ahuja and Orlin (1989) tabulated a list of maximum flow algorithms in literature and their running times. There are also preflow-push algorithms for the maximum flow problem. Preflow-push algorithms maintain a preflow at every step and proceed by pushing the node excess closer to the sink using only local information. The iterative steps involve choosing an active node to send its excess closer to the sink until the network contains no active nodes. A fast and simple excess scaling algorithm was developed by Ahuja and Orlin (1989) for maximum flow problems. The algorithm ensures that flows from active nodes with sufficiently large excesses are pushed to nodes with sufficiently small excesses while never letting the excesses become too large.

The minimum spanning tree problem

Given the nodes of a network and the length for each potential links (or any other alternative measures for the length of a link such as distance, cost, and time). The problem associated with linking the nodes of the network directly or indirectly using shortest length of connecting branches is referred to as the minimum spanning tree problem. Algorithms for the minimum spanning tree problem are found in Hillier and Lieberman (2005) and Taha (2002).

The shortest route problem

The shortest route problem involves finding the shortest route between a source and the destination in a transportation network. Floyd's algorithm is one method that can be used to determine the shortest route between any two nodes in a network. The algorithm represents an $n-$node network as a square matrix with $n-$ rows and $n-$columns. Entry (i, j) of the matrix gives the distance d_{ij} from node i to node j, which is finite if i is linked directly to j, and infinite otherwise. Dynamic programming can also be employed to solve the shortest route problem. The dynamic programming approach is expressed mathematically as: Let $f_i(x_i)$ be the shortest distance to node x_i at stage i and define $d(x_{i-1}, x_i)$ as the distance from node x_{i-1} to node x_i. Then $f_i(x_i)$ is computed from $f_{i-1}(x_{i-1})$ using the forward recursive equation

$$f_i(x_i) = \min \text{ all feasible routes } (x_{i-1}, x_i) \; \{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}, \; i = 1, \ldots, n, \quad (6)$$

where n is the number of stages in the system arising from the decomposition of the $n-$variable problem (Sharma, 2009; Taha, 2002). In the dynamic programming approach, there is also a backward recursive equation for the shortest route problem.

The Survivable Network Design problem

The Survivable Network Design (SND) problem seeks to find a minimum weight subgraph of G such that each pair of nodes i, j has a pre-specified requirement r_{ij} of edge-disjoint $i-j$ paths. When $r_{ij} = 1$ for all i, j, we have the classical minimum spanning tree problem. When $r_{ij} \in \{0,1\}$, we have the NP-hard minimum Steiner tree problem. If given two special nodes p and q such that $r_{pq} = k$ and $r_{ij} = 1$ for all $\{i, j\} \neq \{p, q\}$, then we have the $k-$path tree problem. If $r_{pq} = k$ for the given pair p, q, and $r_{ij} \in \{0,1\}$, then we obtain the $k-$path Steiner tree problem. Several algorithms have been developed for the SND problem (see Arkin and Hassin, 2007).

The minimum balance problem

The minimum balance problem is formulated as follows: Let G be a digraph and $c : E(G) \to R$. The problem is to find a potential $\pi : V(G) \to R$ such that the vector of slacks $slack\,(e) := \pi(w) - \pi(v) - c((v,w))\ (e = (v,w) \in E(G))$ are optimally balance: for any subset of vertices, the minimum slack on an entering edge should be equal to the minimum slack on a leaving edge. The minimum balance problem was applied by Albrecht *et al.*, (2003) to the design of logic chips and its solution was obtained via a parametric shortest path algorithm.

The minimum cost flow problem

The minimum cost flow problem holds a central position among network optimization models, because it encompasses a broad class of application and it can be solved extremely efficiently. Suppose (N, A) is a connected network of n nodes with at least one supply node and at least one demand node. Let x_{ij} be flow through arc $i \to j$, d_{ij} the cost per unit flow through arc $i \to j$, c_{ij} the arc capacity for arc $i \to j$, and b_i the net flow generated at node i. The minimum cost problem can be formulated as a linear programming problem as:

$$\left.\begin{aligned} &\max\ z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \\ &\text{subject to } \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i, \text{ for each node } i, \\ &\text{and } 0 \le x_{ij} \le c_{ij}, \text{ for each arc } i \to j. \end{aligned}\right\} \qquad (7)$$

In some of the applications of minimum cost flow problem, it is necessary to have a lower bound $L_{ij} > 0$ for the flow through each arc $i \to j$. An example is the primal minimal cost network flow problem in Hassin (1983). One way to deal with this lower bound is to use a translation of variables $x'_{ij} = x_{ij} - L_{ij}$ so as to convert the model back to the format in (7).

Another approach to this problem using tree-search algorithm is found in Hassin (1983).

The cyclic exchange problem

Let $A = \{a_1, ..., a_n\}$ be a set of n elements. The collection $\{S_1, ..., S_k\}$ defines a k-partition of A if each set S_j satisfies the following:

(i) $S_j \ne \phi,\ j = 1, ..., k$

(ii) $S_i \cap S_j = \phi,\ i \ne j$, and

(iii) $\bigcup_{j=1}^{k} S_j = A$.

For any subset S of A, let $d[S]$ denote the cost of S. Then the set partitioning problem is to find a partition of A into at most k subsets so as to minimize $\sum_k d[S_k]$. Let $\{S_1, ..., S_k\}$ be any feasible partition. The set $\{T_1, ..., T_k\}$ is a 2-neighbour of $\{S_1, ..., S_k\}$ if it can be obtained by swapping two elements that are in different subsets. The 2-exchange neighbourhood of $\{S_1, ..., S_k\}$ consists of all 2-neighbours of $\{S_1, ..., S_k\}$. The set $\{T_1, ..., T_k\}$ is a cyclic-neighbour of $\{S_1, ..., S_k\}$ if it can be obtained by transferring single elements among a sequence of $k \le K$ subsets in S. Let $(S_h^1, S_m^2, S_n^3, ..., S_p^k)$ be such a sequence of k subsets, then $h = p$, that is the last subset of the sequence is identical to S_h^1. This process is referred to as cyclic exchange.

Ahuja *et al.*, (2003) described the construction of improvement graph for cyclic exchange as follows: Let $A = \{a_1, ..., a_n\}$ be the set of elements for the original set partitioning problem and let $S[i]$ denote the subset containing element a_i. The improvement graph is a graph $G = (V, E)$ where $V = \{1, ..., n\}$ is a set of nodes corresponding to the indices of the elements of A of the original problem.

Let $E = \{(i, j) : S[i] \ne S[j]\}$, where an arc (i, j) corresponds to the transfer of node i from $S[i]$ to $S[j]$ and the removal of j from $S[j]$. For each arc $(i, j) \in E$, let

$C[i, j] = d[\{i\} \cup s[j] \setminus \{j\}] - d[S[j]]$, that is, the increase in the cost of $S[j]$ when i is added to the set and j is deleted. A cycle W in G is subset-disjoint if for every pair i and j of nodes of $W, S[i] \neq S[j]$, that is, the elements of A corresponding to the nodes of W are all in different subsets. There is a one-to-one cost-preserving correspondence between cyclic exchanges for the partitioning problem and subset-disjoint cycles in the improvement graph. In particular, for every negative cost cyclic exchange, there is a negative cost subset-disjoint cycle in the improvement graph. It is important to mention here that the problem of finding a negative cost subset-disjoint cycle is NP-hard (Ahuja *et al.*, 2003).

The Travelling Salesman's Problem

The Travelling Salesman's Problem (TSP) basically deals with a salesman starting at a certain city who wants to find a route of minimum length, which traverses each of the destination cities exactly once and leads him back to the starting city. Amponsah *et al.*, (2010) formulated the TSP as a graph problem as follows: Given an undirected complete graph $K_n = (V, E)$ and edge weights $w : E \to R_0^+$, where n is the cardinality of the set of vertices V. The task is to find a Hamiltonian cycle with minimum weight in K_n. The authors solved the TSP by computing the total cost for distinct circuits so as to select the one with minimum cost.

Formulation of the expected duration of stay in the educational system as a NDP

The educational system is a well-known hierarchical system in literature (Gani, 1963; Ekhosuehi and Osagiede, 2011). In analyzing flows in the educational system, the Markov chain model is often employed (Osagiede and Ekhosuehi, 2006; Nicholls, 2009). In this sub-section, we shall formulate the minimum period of stay in the educational system using graph-theoretic concepts.

Consider a digraph $G = (V, E)$ with a set of V vertices (which are the levels in the educational system) and a set E of edges (which are the arrows indicating the flows) such that each unique edge $e = (i, j) \in E \subseteq V \times V$, is a unique ordered pair. Let the cardinality of set V be k and $\sigma_{[i]}(e)$ be the weight (duration) for the flow in edge e starting from vertex i. The duration of each path in G starting from vertex i^* is given by the sequence $(\sigma_{[i]}(e))_{i=i^*}^{k}$, where k is the highest grade in the system. To estimate the duration $\sigma_{[i]}(e)$, we employ the Fundamental Matrix, FM, of the absorbing Markov chain (Ibe, 2009). The matrix FM is given as: $\mathrm{FM} = (\mathrm{I} - \mathrm{P})^{-1}$, where I is a $k \times k$ identity matrix and P is a $k \times k$ sub-stochastic internal transition matrix. Since the FM matrix gives the expected period of stay in the system before absorption, the entries in the FM matrix give the respective expected duration $E[\sigma_{[i]}(e)]$ for each flow.

In practice, the policy framework in the educational system may allow new entrants into specific grades of the system so that the starting point i^* is not unique. Furthermore, the maximum duration for a student to complete the programme may be fixed so much so that any student who exceeds the maximum period allowed is withdrawn from the system. Similarly, the minimum duration for a student to graduate is fixed. In this light, the problem associated with the duration of stay in the system is formulated as a NDP by minimizing the expected duration of each path that leads to graduation as:

$$\min_{\forall e} \left(\bigcup_{i=i^*}^{k} E[\sigma_{[i]}(e)] \right) \tag{8}$$

subject to

$$n \leq \sum_{i=i_r}^{k} E[\sigma_{[i]}(e)] \leq m, \tag{9}$$

$$i^* = \max_{r}(i_r), \text{ for each } r, \tag{10}$$

where n and m are respectively the minimum and maximum number of sessions required for a student to graduate and i_r is the r^{th} grade in which new entrants are allowed.

RESULTS AND DISCUSSION

We shall illustrate the use of graph theory by considering a six-year-graded academic programme of a university setting in Nigeria with a set of levels $S = \{1, 2, ..., 6\}$. New entrants are allowed into the programme only through Year 1 and Year 2. The maximum periods allowed in the programme are nine sessions for a student starting from Year 1 and eight sessions for a student starting from Year 2. A student starting from Year 2 spends a minimum of five sessions. The flow F in the programme is a function $F: E \to Z_+$, where Z_+ is the set of positive integers, satisfying the state-transition relation $\Diamond R$ on the set $\Re = S \cup \{0\}$, $\Diamond R \subset \Re \times \Re$, such that:

$\Diamond R =$ {(0, 1), (0,2,), (1,2), (2,3), (3,4), (4,5), (5,6), (6,6), (1,0), (2,0), (3,0),(4,0),(5,0),(6,0)}

Observe that the relation $\Diamond R$ is neither reflexive nor symmetric, since for some $i \neq 6$, $i \in \Re$, $(i,i) \notin \Diamond R$, and for $(i, j) \in \Diamond R$, $i < j$ and $j \neq 0$, $(j,i) \notin \Diamond R$. The reason for this is that, in the programme, there is no repetition of levels except at the apex level and there is no demotion.

Thus, the inverse relation $(\Diamond R)^{-1}$ cannot hold. We can graph the relation $\Diamond R$ on the set $\Re = S \cup \{0\}$ by drawing an arrow from each element i to each element j, whenever i is related to *j*. The resulting graph is depicted in Fig. 1.

The transition diagram in Figure 1 is a digraph $G = (V, E)$ with $|V| = 6$, $|E| = 15$. The edges e_8 and e_{10}, and e_{14} and e_{15} are respectively parallel edges, while the edge e_7 forms a loop. Edges e_8 and e_{10} represent a direct entry admission into the programme in Year 2 and wastage from Year 2 of the programme, respectively; and the edges e_{14} and e_{15} represent wastage in Year 6 and graduation, respectively. The loop e_7 denotes an aggregation of referred cases with the provision that the maximum number a student can repeat is not exceeded.

Data was obtained for the academic programme for a period of six sessions from the approved results by Senate of the institution. We represent the data in the following flow matrices and vectors:

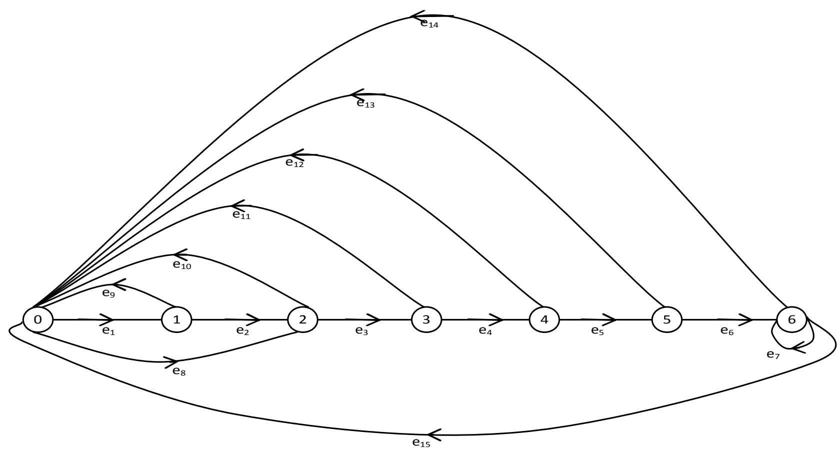

Fig. 1: A transition graph, G of the relation $\Diamond R$.

$$\mathbf{F}1=\begin{bmatrix}0&112&0&0&0&0\\0&0&53&0&0&0\\0&0&0&56&0&0\\0&0&0&0&30&0\\0&0&0&0&0&35\\0&0&0&0&0&8\end{bmatrix},\ \mathbf{W}1=\begin{bmatrix}0\\0\\0\\0\\0\\10\end{bmatrix},$$

$$\mathbf{F}2=\begin{bmatrix}0&106&0&0&0&0\\0&0&90&0&0&0\\0&0&0&45&0&0\\0&0&0&0&48&0\\0&0&0&0&0&26\\0&0&0&0&0&8\end{bmatrix},\ \mathbf{W}2=\begin{bmatrix}4\\22\\8\\8\\4\\35\end{bmatrix},$$

$$\mathbf{F}3=\begin{bmatrix}0&234&0&0&0&0\\0&0&78&0&0&0\\0&0&0&87&0&0\\0&0&0&0&45&0\\0&0&0&0&0&43\\0&0&0&0&0&13\end{bmatrix},\ \mathbf{W}3=\begin{bmatrix}2\\28\\3\\0\\5\\21\end{bmatrix},$$

$$\mathbf{F}4=\begin{bmatrix}0&346&0&0&0&0\\0&0&226&0&0&0\\0&0&0&78&0&0\\0&0&0&0&87&0\\0&0&0&0&0&43\\0&0&0&0&0&20\end{bmatrix},\ \mathbf{W}4=\begin{bmatrix}7\\8\\0\\0\\2\\36\end{bmatrix},$$

$$\mathbf{F}5=\begin{bmatrix}0&470&0&0&0&0\\0&0&404&0&0&0\\0&0&0&211&0&0\\0&0&0&0&78&0\\0&0&0&0&0&80\\0&0&0&0&0&35\end{bmatrix},\ \mathbf{W}5=\begin{bmatrix}1\\2\\15\\0\\7\\28\end{bmatrix},$$

$$\mathbf{F}6=\begin{bmatrix}0&179&0&0&0&0\\0&0&489&0&0&0\\0&0&0&397&0&0\\0&0&0&0&205&0\\0&0&0&0&0&67\\0&0&0&0&0&44\end{bmatrix},\ \mathbf{W}6=\begin{bmatrix}2\\3\\7\\6\\11\\71\end{bmatrix},$$

where **F** is a matrix, which contains the flow from level i to level j, and **W** is a column vector of wastage flow out of level i.

By considering the state-transition relation $\Diamond R$ on the set $\Re = S \cup \{0\}$, we form a matrix, A= (a_{ij}), whose rows and columns are labelled by the elements of $\Re$, where a_{ij} is given as:

$$a_{ij}=\begin{cases}p_{ij} & \text{if } \exists \text{ a relation from } i \text{ to } j,\ i\in S,\ j\in\Re\\ 0 & \text{otherwise}\end{cases}$$

So, we have

$$\mathbf{A}=\begin{array}{c}\\ \text{Nonabsorbing}\\ \\ \text{Absorbing}\end{array}\overset{\text{Nonabsorbing}\qquad\qquad \text{Absorbing}}{\begin{bmatrix}p_{11}&p_{12}&\cdots&p_{16}&\vdots&p_{10}\\p_{21}&p_{22}&\cdots&p_{26}&\vdots&p_{20}\\\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\p_{61}&p_{62}&\cdots&p_{66}&\vdots&p_{60}\\\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&\ldots&0&\vdots&1\end{bmatrix}}$$

The transition matrix **A** is called the absorbing Markov chain as it contains an absorbing state. We shall then estimate the transition probabilities of A from the flow matrices using the maximum likelihood method described in Zanakis and Maret (1980). Thus we obtain the canonical form of **A** as

$$\mathbf{A}=\begin{bmatrix}0&0.9891&0&0&0&0&0.0109\\0&0&0.9551&0&0&0&0.0449\\0&0&0&0.9636&0&0&0.0364\\0&0&0&0&0.9724&0&0.0276\\0&0&0&0&0&0.9102&0.0898\\0&0&0&0&0&0.3891&0.6109\\0&0&0&0&0&0&1.0000\end{bmatrix}$$

From matrix **A**, we have, on the average that over 90% of students are promoted to the next higher level (and those who are not promoted drop-out), while about 39% of the students repeat the final year. Next, we compute the Fundamental Matrix (FM) using the formula $(\mathrm{I}-\mathrm{P})^{-1}$, where I is an 6×6 identity matrix. Entries in the FM give the respective expected duration $E[\sigma_{[i]}(e)]$ for each flow. So we have

$$\text{FM}=\begin{bmatrix} 1.0000 & 0.9891 & 0.9447 & 0.9103 & 0.8851 & 1.3187 \\ 0 & 1.0000 & 0.9551 & 0.9203 & 0.8949 & 1.3333 \\ 0 & 0 & 1.0000 & 0.9636 & 0.9370 & 1.3960 \\ 0 & 0 & 0 & 1.0000 & 0.9724 & 1.4487 \\ 0 & 0 & 0 & 0 & 1.0000 & 1.4899 \\ 0 & 0 & 0 & 0 & 0 & 1.6368 \end{bmatrix}$$

From the FM, the expected number of sessions a student starting from Year 1 will spend in Year 2 is 0.9891, then 0.9447 in Year 3, 0.9103 in Year 4 and so on, before completing the programme. Similarly, a student starting from Year 2 is expected to spend 0.9551 of the session in Year 3, 0.9203 in Year 4, and so on before completing the programme. The same explanation holds for a student in Years 3, 4, 5 and 6. However, in Year 6, a student is expected to spend over one session and a semester because the entries in the sixth column are greater than one. Thus there is a delay in Year 6. The delay in Year 6 may be occasioned by the graduating requirement where students are expected to accumulate a minimum of 130 credit passes and therefore repeat all failed courses at the lower level(s).

From the foregoing, we obtain the expected duration of each starting point that leads to graduation as:

$$\sum_{i=1}^{6} E[\sigma_{[i]}(e)] = 6.0479, \quad \sum_{i=2}^{6} E[\sigma_{[i]}(e)] = 5.1037,$$

$$\sum_{i=3}^{6} E[\sigma_{[i]}(e)] = 4.2966, \quad \sum_{i=4}^{6} E[\sigma_{[i]}(e)] = 3.4211$$

$$\sum_{i=5}^{6} E[\sigma_{[i]}(e)] = 2.4899, \quad E[\sigma_{[6]}(e)] = 1.6368.$$

These results do not exceed the maximum period allowed of nine sessions for a student starting from Year 1 and eight sessions for a student starting from Year 2 for the programme. But $r = 1, 2$, since new entrants are allowed into either Year 1 or Year 2. So, using the second constraint (10), $i^* = \max_{r=1,2}(i_r) = 2$.

Thus, we compare the expected duration for graduation using Year 2 as the starting point.
So we obtain:

$$\sum_{i=2^*}^{6} E[\sigma_{[i]}(e)] = 5.0479 \text{ and } \sum_{i=2^*}^{6} E[\sigma_{[i]}(e)] = 5.1037$$

Hence, the expected duration of each path that leads to graduation is optimal when new entrants start from Year 1, since

$$\min_{\forall e}\left(\bigcup_{i=1}^{6} E[\sigma_{[i]}(e)]\right) =$$

$$\{1.0000, 0.9891, 0.9447, 0.9103, 0.8851, 1.3187\}$$

The implication of this result is that students admitted through Year 1 are expected to complete the programme earlier than new entrants into Year 2.

CONCLUSION

In this paper, we have presented several aspects of NDPs with particular reference to the formulation of the expected duration of stay in the educational system as a NDP. We have also mentioned preflow-based algorithms without pointing-out the length of time in executing such algorithms as a snag. An improvement on the length of time for preflow-based algorithms and effective implementation of the resulting algorithm to a real-world system are topics for further research. Another grey area is the maximum flow problem on a network where the nodes are partitioned so much so that $G = (N, A)$ is such that $N = N_1 \cup N_2$ and $A \subseteq N_1 \times N_2$.

References

Ahuja, V. (1982). *Design and Analysis of Computer Communication Networks*. McGraw-Hill, Inc., New York.

Ahuja, R. K., Ergun, O., Orlin, J. B. and Punnen, A. P. (2003). A survey of very large-scale neighbourhood search techniques. In: E. Boros and P. L. Hammer (Eds), *Discrete Optimization: The State of the Art*. Elsevier, Amsterdam. Pp. 75-102.

Ahuja, R. K. and Orlin, J. B. (1989). A fast and simple algorithm for the maximum flow problem. *Operations Research,* 37(5): *748-759.*

Albrecht, C., Korte, B., Schietke, J. and Vygen, J. (2003). Maximum mean weight cycle in adigraph and minimizing cycle time of a logic chip. In: E. Boros and P. L. Hammer (Eds), *Discrete Optimization: The State of the Art*. Elsevier, Amsterdam. Pp. 103-127.

Amponsah, S. K., Darkwah, K. F. and Ankamah, J. D. (2010). Inspection of electoral materials in Ghana as travelling salesman's problem. *The Journal of the Mathematical Association of Nigeria (abacus), 37(2): 22-33*.

Arkin, E. M. and Hassin, R. (2007). The k -path tree matroid and its applications to survivable network design. Retrieved on 24/09/2009 from http://www.ams.sunysb.edu/ ~estie/ papers/k-path.pdf

Borndorfer, R. (2008). Discrete optimization in public transportation. Paper presented at the 1st Indo-US symposium on *Advances in Mass Transit and Travel Behaviour Research (MTTBR-08)*, February 12-15, 2008, IIT Guwahati, Assam, India.

Ekhosuehi, V. U. and Osagiede, A. A. (2011). On the asymptotic effect of examination malpractice on the structure of a Markovian multi-echelon system. *International Journal of Mathematics in Operational Research, 3(6): 619-635*.

Gani, J. (1963). Formulae for projecting enrolments and degrees awarded in universities *.Journal of the Royal Statistical Society. Series A (General), 126(3): 400-409*.

Hassin, R. (1981a). Generalizations of Hoffman's existence theorem for circulations. *NETWORKS, 11: 243-254*.

Hassin, R. (1981b). Maximum flow in (s,t) planar networks. *Information Processing Letters* 13, 107.

Hassin, R. (1983). The maximum cost flow problem: A unifying approach to dual algorithms and a new tree-search algorithm. *Mathematical Programming, 25: 228-239.*

Hassin, R. and Tamir, A. (1986). Efficient algorithms for optimization and selection on series-parallel graphs. *SIAM J. ALG. DISC. METH., 7(3): 379-389.*

Hassin, R., Ravi, R. and Salman, F. S. (2004). Approximation algorithms for a capacitated network design problem. *Algorithmica* 38: *417-431*.

Hillier, F. S. and Lieberman, G. J. (2005). *Introduction to Operations Research* (8th Ed.). McGraw-Hill, New York.

Ibe, C. O. (2009). *Markov processes for stochastic modelling.* Elsevier Academic Press, Burlington, USA.

Nicholls, M. G. (2009). The use of Markov models as an aid to the evaluation, planning and benchmarking of doctoral programs. *Journal of the Operational Research Society*, 60:1183-1190.

Osagiede, A. A. and Ekhosuehi, V. U. (2006). Markovian approach to school enrolment projection process. *Global Journal of Mathematical Sciences, 5(1): 1–7.*

Osagiede, A. A. and Ekhosuehi, V. U. (2009). On the choice of schools located outside the walkable neighbourhood of the household. *Monografias del Seminario Matematico ''Garcia de Galdeano'', 36: 117-126*.

Sharma, J. K. (2009). *Operations research: theory and applications* (4th ed.). Macmillan Publishers India Ltd., New Delhi.

Taha, H. A. (2002). *Operations Research: An Introduction* (7th ed.). Pearson Education (Singapore) Pte. Ltd., Delhi.

Zanakis, S. H. and Maret, M. W. (1980). A Markov chain application to manpower supply planning. *Journal of the Operational Research Society, 31: 1095-1102*.