

# A BRIEF NOTE ON BUS BASED EVACUATION PLANNING

**Bam Bahadur Thapa and Shree Ram Khadka**

**Journal of Institute of Science and Technology**

*Volume 21, Issue 1, August 2016*

*ISSN: 2469-9062 (print), 2467-9240(e)*

**Editors:**

Prof. Dr. Kumar Sapkota

Prof. Dr. Armila Rajbhandari

Assoc. Prof. Dr. Gopi Chandra Kaphle

*JIST, 21 (1), 28-34 (2016)*



**Published by:**

**Institute of Science and Technology**

Tribhuvan University

Kirtipur, Kathmandu, Nepal

## A BRIEF NOTE ON BUS BASED EVACUATION PLANNING

**Bam Bahadur Thapa<sup>1</sup> and Shree Ram Khadka<sup>2,\*</sup>**

<sup>1</sup>Padma Kanya Campus, Tribhuvan University, Bagbazar, Kathmandu, Nepal

<sup>2</sup>Central Department of Mathematics, Tribhuvan University, Kirtipur, Nepal

\*Corresponding email: shreeramkhadka@gmail.com

### ABSTRACT

Evacuation route planning is essential for emergency preparedness, especially in regions threatened by hurricanes, earthquake etc. The evacuees who do not travel on their own are gathered at few collection points, where they are brought on buses to take them to safe region in bus based evacuation. Bus based evacuation planning is more essential for developing country like Nepal. It is necessary to reach the destination as early as possible for bus based evacuation. The bus based evacuation planning (BEP) problem is a variant of vehicle routing problem that arises in emergency planning. The problem in this variant is that not all the evacuees can gather at the same time, elderly and handicaps may need special help like wheelchair and may need more pickup time. Another problem is that at the shelter, some people may need special care like medicine, oxygen etc. In this paper we review a mathematical model to minimize the duration of evacuation together with the dynamic programming and the branch and bound as solution procedures. Moreover, a brief report of a case study for Kathmandu has also been given.

**Key words:** Emergency preparedness, Route planning, Bus based evacuation, Disaster management

### INTRODUCTION

Evacuation is a process in which threatened people are shifted from dangerous places to safer places in order to reduce the health and life vulnerability of affected people as quickly as possible. Evacuation planning is the emergency management in which people are transported from danger zone to safe zone as soon as possible in a minimum time. Evacuation planning is essential for emergency preparedness, especially in regions threatened by hurricanes, typhoon, or after earthquake, landslide, flood, fire, bomb blast, industrial accident, terrorist attack etc.

In bus based evacuation evacuees who do not travel on their own due to age, sickness, handicapped, children, tourists or who do not have their own car are gathered at few collection points, where they are brought on buses to take them to safe region. Bus based evacuation planning (BEP) is more essential for developing country like Nepal because most of the people in Nepal cannot afford their own car and are bus-based. Route planning for bus evacuation is necessary to reach the destination on shortest time. The problem of bus based evacuation is that all the evacuees cannot gather same time, elderly and handicaps may need special help like wheelchair and may need more pickup time. Another problem is that at the shelter, some people

may need special care like medicine, oxygen etc. There are many studies regarding vehicle routing. However, only few studies have been done till date regarding the bus based evacuation planning. Bish (2011) introduced a model for bus based evacuation planning. Pyakurel *et al.* (2016) performed a case study on transit dependent evacuation planning for Kathmandu valley by using branch and bound algorithm for BEP and tabu search for robust BEP. Goerigk *et al.* (2013) proposed several branch and bound algorithm for bus based evacuation. Goerigk and Grun (2014) considered robust bus evacuation models in which the number of evacuees is assumed to be not known but it can be estimated.

This paper reviews a mathematical model for BEP problem with the dynamic programming and the branch and bound as solution procedures. Moreover, a brief report of a case study for Kathmandu has also been given. This study is especially useful when advance notice of a threat like hurricane, typhoon is available or after land slide, earthquake and flood etc. The objective of this paper is to transport evacuees from the pickup locations to the shelters in the minimal amount of time. We define the duration as the time span between the first bus leaves its yard until the last evacuee is sheltered. We assume that evacuees that do not travel on their own vehicle due to age,

sickness, the lack of a private car or any other reasons, are gathered at few collection points (demand nodes), where they are picked up by buses to take them to safe region (shelter). We have applied Bellman's equation and branch and bound by solving an example.

The paper has been organized as follows. A mathematical formulation of the BEP problem has been described in Section 2, solution procedures the dynamic programming and the branch and bound and also a report for the case study for Kathmandu have been studied in Section 3. The last section concludes the paper.

### MATHEMATICAL FORMULATION

#### Variables with symbols

Consider a network  $(N, A)$ , where  $N$  and  $A$  denote the set of nodes and arcs respectively.  $N = Y \cup P \cup S$  where  $Y$  is a set of yard nodes initially located,  $P$  is a set of demand nodes, each of which represent a pick up location serving a neighborhood,  $S$  is a set of shelter nodes.  $V$  is the set of available buses, each having capacity  $Q$ , is subdivided into subsets  $V_i$ ,  $i \in Y$ , where bus is initially located at yard  $i$ . Demand node  $j$  has a

demand  $D_j$ ,  $j \in P$ , and shelter  $i$  has a capacity  $C_i$ ,  $i \in S$ . Here we use term trip as arc traversal.  $\tau_{ij}$  is none negative travel cost of each arc  $(i, j) \in A$ . We suppose three assumptions for the arc set:

- (1)  $\tau_{ij} = \tau_{ji}$  (symmetric),
- (2)  $\tau_{ij} \leq \tau_{ik} + \tau_{kj}$ ,  $\forall (i, k), (k, j) \in A$  (triangular inequality) and
- (3) Travel cost is proportional to the travel time and distance.

Figure of a BEP network with one yard (Y), three demand nodes (P1, P2 and P3), shelters with two capacities (S1 and S2) and number of buses (Vv) is given below. It is written as YP3S2Vv in short form.

The number of evacuees at demand node might be greater than the capacity of a bus, so the split delivery is allowed. BEP network is not fully connected because outgoing arcs are used to go only outside from the yards. The yard plays no further role in the evacuation process because buses do not return to the yard due to the bus and driver's safety. Liman (2006) mentioned that hurricane Katrina flooded bus yards along with buses in New Orleans. So yard is not the best place to store the buses during the threat.

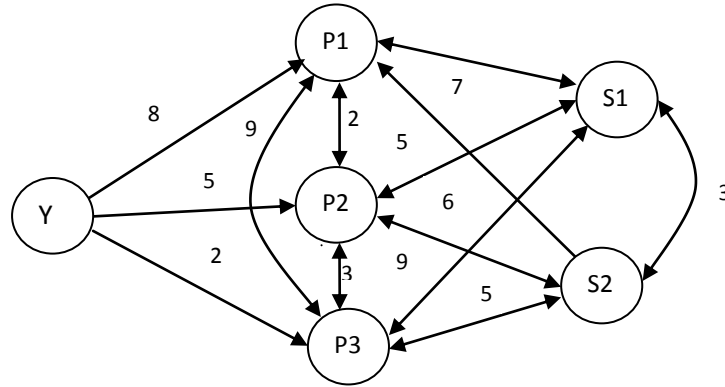


Figure 1: BEP-type network (Scenario YP3S2Vv)

#### Decision variables

The variables used to develop following formulation for BEP are

$x_{ij}^{mt}$ : binary variable that equals 1 if trip  $t$  for bus  $m$  traverses arc  $(i, j)$ , else 0,  $\forall (i, j) \in A$ ,

$$m \in V, t = 1, \dots, T.$$

$b_j^{mt}$ : number of evacuees from node  $j$  assigned to (or, if  $j$  is shelter, released from) bus  $m$  after trip  $t$ ,  $\forall j \in N, m \in V, t = 1, \dots, T$ .

$T_{evac}$ : the duration of the evacuation.

#### Formulation for BEP

The Bus Evacuation Problem (BEP) is a vehicle routing problem that arises in emergency planning. It models the evacuation of a region from a set of collection points to a set of capacitated shelters with the help of buses, minimizing the time needed to bring the last person out of the endangered region (Goerigk *et al.* 2013).

Following formulation have been introduced to minimize the duration of evacuation for BEP:

- (1) Minimize  $T_{evac}$

- (2) Subjected to :  $T_{evac} \geq \sum_{(i,j) \in A} \sum_{t=1}^T \tau_{ij} x_{ij}^{mt}$ ,  
 $\forall m \in V$
- (3)  $\sum_{i:(i,j) \in A} x_{ij}^{mt} = \sum_{k:(j,k) \in A} x_{jk}^{m(t+1)}$ ,  $\forall j \in P$ ,  
 $m \in V$ ,  $t = 1, \dots, T - 1$
- (4)  $\sum_{i:(i,j) \in A} x_{ij}^{mt} \geq \sum_{k:(j,k) \in A} x_{jk}^{m(t+1)}$ ,  $\forall j \in S$ ,  
 $m \in V$ ,  $t = 1, \dots, T - 1$
- (5)  $\sum_{(i,j) \in A} x_{ij}^{mt} \leq 1$ ,  $\forall m \in V$ ,  $t = 1, \dots, T$
- (6)  $x_{ij}^{m1} = 1$ ,  $\forall i \in Y$ ,  $j: (i,j) \in A$ ,  $m \in V_i$
- (7)  $x_{ij}^{mt} = 0$ ,  $\forall i \in Y$ ,  $j: (i,j) \in A$ ,  $m \in V$ ,  $t = 2, \dots, T$
- (8)  $x_{ij}^{mT} = 0$ ,  $\forall j \in P$ ,  $i: (i,j) \in A$ ,  $m \in V$
- (9)  $b_j^{mt} \leq \sum_{(i,j) \in A} Q x_{ij}^{mt}$ ,  $\forall j \in N$ ,  $m \in V$ ,  $t = 1, \dots, T$
- (10)  $0 \leq \sum_{j \in P} \sum_{l=1}^t b_j^{ml} - \sum_{k \in S} \sum_{l=1}^t b_k^{ml} \leq Q$ ,  $\forall m \in V$ ,  $t = 1, \dots, T$
- (11)  $\sum_{m \in V} \sum_{t=1}^T b_j^{mt} \leq C_j$ ,  $\forall j \in S$
- (12)  $\sum_{m \in V} \sum_{t=1}^T b_j^{mt} = D_j$ ,  $\forall j \in P$
- (13)  $\sum_{j \in P} \sum_{t=1}^T b_j^{mt} = \sum_{k \in S} \sum_{t=1}^T b_k^{mt}$ ,  $\forall m \in V$
- (14)  $x_{ij}^{mt} \in \{0,1\}$ ,  $\forall (i,j) \in A$ ,  $m \in V$ ,  $t = 1, \dots, T$
- (15)  $b_j^{mt} \geq 0$ ,  $\forall (i,j) \in A$ ,  $m \in V$ ,  $t = 1, \dots, T$

Objective function (1) is to minimize the duration of evacuation. Constraint (2) requires  $T_{evac}$  to be greater than or equal to the maximum cost incurred by any bus which is then minimized by the objective function (1). Therefore it is referred as the "min-max" objective. Constraint (3) is the flow-balance constraint for the demand node. It ensures that bus travelling to demand node  $j$  on trip  $t$  leaves node  $j$  on trip  $t+1$ . Constraint (4) is the flow-balance constraint for the shelter. It ensures that the last bus doesn't have to leave the shelter. Constraint (5) allows a bus to make at most one trip at a time. Constraint (6) denotes that the first trip of each bus starts from its yard. Constraint (7) doesn't allow buses to start from yard for later trips. Constraint (8) doesn't allow the last trip of bus to end at the demand node. Constraint (9) dictates that bus can pick up evacuees from node  $j$ , if it travels to node  $j$ . Constraint (10) is the bus capacity. Constraint (11) is the shelter capacity. Constraint (12) ensures that all the evacuees are picked up from demand node. Constraint (13) ensures that all the evacuees are delivered to the shelter. Constraints (14) and (15)

are logical binary and non-negativity restrictions on the  $x$  and  $b$  variables respectively.

## Solution techniques

### BEP as a network

The BEP network structure adds complexity to the solution. In addition to the demand nodes to which initially served from each yard, it indicates which shelter each vehicle should use considering shelter capacities and their location to the demand nodes. However, according to Bish (2011) even for the case where the shelters are un-capacitated (have sufficient capacities for all evacuees), sending each vehicle to its closest shelter from its last pick up node is not necessarily optimal. It is shown by the following example:

**Observation 1:** The route may not be optimal even though it passes through the nearest shelter. According to Bish (2011) even without shelter capacity constraints, it is not always optimal for the BEP to allocate each vehicle to the nearest shelter (i.e. the shelter closest to its pick up node). It is clearer by the following example:

**Example1:** For optimal solution a bus might not use the nearest shelter if it is scheduled to serve additional routes, even without capacity constraints. For instance let's consider the following network fragment. There are two pick up nodes (P1 and P2) and two shelters (S1 and S2). A bus has to pick up a full load of evacuees at node P1 and must pick up a full load of evacuees at node P2 also. The bus can transport the current load of evacuees to either shelter S1 or S2. In the following figure shelter S1 is closer than S2 from pick up node P1. If the closest shelters are used, the bus will take route P1-S1-P2-S2 which has the cost of 15 (5+7+3), whereas if the bus takes route P1-S2-P2-S2, the cost would be 14 (8+3+3) which is optimal solution.

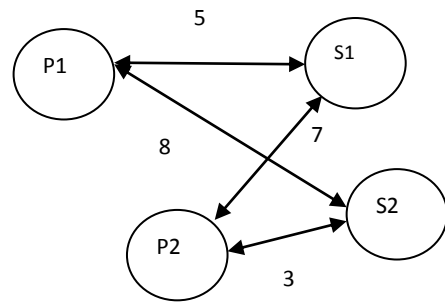


Figure 2: A network for the un-capacitated shelter example.

**Dynamic programming**

A Bellman’s equation, named after its discoverer, Richard Bellman, also known as a dynamic programming equation, is a necessary condition for optimality associated with the mathematical optimization method known as dynamic programming.

Dynamic programming breaks a multi-period planning problem into simpler steps at different points in time. Therefore, it requires keeping track of how the decision situation is evolving over time (Wikipedia).

According to Pedregal (2004) in order to complete a whole process and to reach a desired state, a system has to move successively through a number of different steps. Each one of these actions has an associated cost. To reach the desired state from the given initial state we have to determine the optimal global strategy with the least cost.

**Variables:**

$t$ : denotes the variable indicating the successive stages in which a decision must be made about where to lead the system

$$t = t_i, i = 0, 1, \dots, n;$$

$x$ : denotes the variable describing the state of the system. At each step  $i$ , we should have

$$x \in A_i$$

$A_i$ : denotes the finite set of feasible states when  $t = t_i$ . The cost associated with the passage from  $x \in A_i$  to  $y \in A_{i+1}$  is denoted by  $c(i, x, y)$

The main aim of the present work is to reach the final desired state  $(t_n, x_n)$  from the initial state  $(t_0, x_0)$  with the least cost by determining the optimal strategy. This is a typical situation of dynamic programming.

We know the optimal path starting from  $(t_0, x_0)$  and going to  $(t_j, x)$  for each  $x \in A_i$  where  $0 < j < n$ .

Let  $S(t_j, x)$  denotes the cost associated with such an optimal strategy ending at  $(t_j, x)$ .

The problem  $\min_{x \in A_j} [S(t_j, x) + c(j, x, y)]$  which is fundamental law or property of dynamic programming through which optimal cost from  $(t_0, x_0)$  to  $(t_n, x_n)$  can be found out in the most rational way.

**Proposition 1:** Pedregal (2004) has given following proposition to understand the fundamental property of dynamic programming.

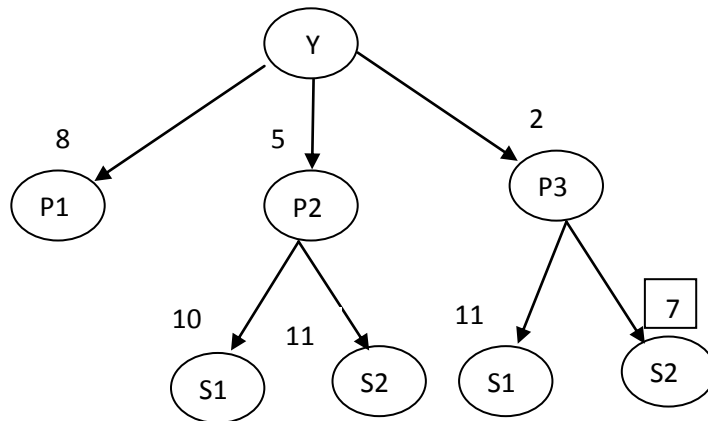
If the  $S(t_j, x)$  denotes the optimal cost from  $(t_0, x_0)$  to  $(t_j, x)$ , then we must have

$$S(t_{j+1}, y) = \min_{x \in A_j} [S(t_j, x) + c(j, x, y)]$$

We have applied Bellman’s equation to find the shortest distance for BEP. Example has been given below:

**Example 2**

Let us consider the network fragment with one yard {Y}, three pick up nodes {P1, P2, P3} and two shelter nodes {S1, S2}.



**Figure 3: BEP type network (Scenario YP3S2Vv).**

In this example we have three stages  $t_0, t_1, t_2$  with associated sets of feasible states.

For each of the pickup nodes {P1, P2, P3}, we can find a unique path from yard Y, so that it must be optimal, and  $S(t_1, P1) = 8, S(t_1, P2) = 5, S(t_1, P3) = 2$ .

For the each of the shelter nodes {S1, S2}, we can find the optimal cost based on the fundamental property of dynamic programming.

$$S(t_{j+1}, y) = \min_{x \in A_j} [S(t_j, x) + c(j, x, y)].$$

We are searching for the minimum of following:

$$\begin{array}{lll} 8+5 & 5+5 & 2+9 \\ 8+7 & 5+6 & 2+5 \end{array}$$

Then,

$$S(t_2, S1) = \min \{8+5, 5+5, 2+9\} = 10$$

$$S(t_2, S2) = \min \{8+7, 5+6, 2+5\} = 7$$

Consequently the shortest distance (minimum cost) is 7 and the corresponding route is Y-P3-S2.

### Branch and Bound

Branch and bound method is an algorithm design paradigm for discrete and combinatorial optimization problems, as well as general real valued problems. A branch and bound algorithm consists of a systematic enumeration of candidate solutions by means of state space search: the set of candidate solutions is thought of as forming a rooted tree with the full set at the root. The algorithm explores *branches* of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower estimated *bounds* on the optimal solution, and is discarded if it cannot produce a better solution than the best one found so far by the algorithm (Wikipedia).

The algorithm depends on the efficient estimation of the lower and upper bounds of a region / branch of the search space and approaches exhaustive enumeration as the size ( $n$ -dimensional volume) of the region tends to zero (Wikipedia).

The method was first proposed by Land and Doig in 1960 for discrete programming, and has become the most commonly used tool for solving NP-hard optimization problems (Land and Doig 1960, Jens 1999). The name "branch and bound" first occurred in the work of Little *et al.* on the traveling salesman problem (John *et al.* 1963, Egon & Paolo 1983).

According to Papadimitriou and Steiglitz (2006) the branch and bound method is a way in which we try to construct a proof that a solution is optimal based on successive partitioning of the solution space. The branch refers to the partitioning process and bound refers to lower bounds that are used to construct a proof of optimality without exhaustive search. Like Bell's equation this method is also applied to find out the shortest distance for BEP.

In a general context two things were needed to develop the tree in the branch and bound algorithm for ILP (Integer Linear Programming).

1. *Branching* A set of solutions, which is represented by a node, can be partitioned into mutually exclusive sets. Each subset in the partition is represented by a child of the original node.
2. *Lower bounding* An algorithm is available for calculating a lower bound on the cost of any solution in a given subset.

The set active set is used to hold the live nodes at any point; the variable U is used to hold the cost of the best complete solution at any given time (U is an upper bound on the optimal cost). Notice that the branching process needs not to produce only two children of a given node, as in the ILP version, but any finite number.

**Algorithm** (Papadimitriou & Stieglitz 2006)

**begin**

Active set:=0; (**comment:** "0" is the original problem)

U:=∞;

Current best:=anything;

**while** active set is not empty **do**

**begin**

choose a branching node, node  $k \in$  active set;

remove node k from active set;

generate the children of node k, child  $i, i = 1, \dots, n_k,$

and the corresponding lower bounds,  $z_i;$

**for**  $i = 1, \dots, n_k$  **do**

**begin**

if  $z_i \geq U$  **then** kill child  $i$

**else if** child  $i$  is a complete solution **then**

$U := z_i,$  current best: = child  $i$

**else** add child  $i$  to active set

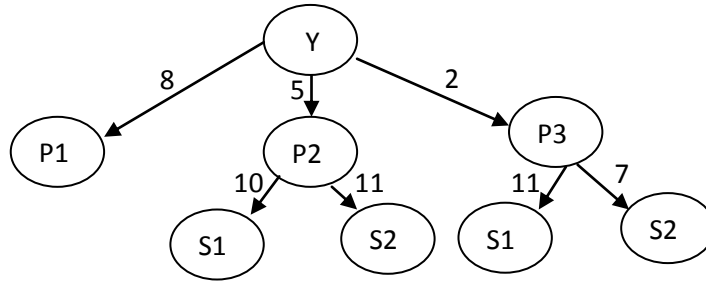
**end**

**end**

**end**

To determine the optimal solution of above mentioned example 2 by branch and bound, following figure has been used. The number given adjacent to the node is the total distance.

Let us consider one yard {Y}, three pick up nodes {P1, P2, P3} and two shelter nodes {S1, S2} in the following tree search.



**Figure4: A shortest path problem and its solution by branch-and-bound.**

In order to solve the figure 3 by branch and bound we branch by choosing the next arc with which continue the path. The figure 4 shows a snapshot of a search tree that results when we branch from a node with the lowest lower bound at any point. The total distance of the following paths are given below:

Y-P1 is 8, Y-P2-S1 is 5+5(10), Y-P2-S2 is 5+6 (11), Y-P3-S1 is 2+9 (11) and Y-P3-S2 is 2+5 (7). The nodes other than the lowest one are killed. So the optimal solution is 7 and the corresponding path is Y-P3-S2.

#### **A case study for Kathmandu valley**

Pyakurel *et al.* (2016) have done a case study on transit dependent evacuation planning for Kathmandu valley. The objective of this study was to formulate a mathematical model of the densely populated metropolitan capital city, Kathmandu and implement an evacuation plan using available efficient software. They applied two models and solution algorithms to evacuate a core part of the capital city Kathmandu of Nepal, where a large part of population is transit dependent. They applied branch and bound algorithm for bus based evacuation planning (BBEP) and tabu search for robust bus based evacuation planning (RBBEP). They calculated and compared minimum, average and maximum evacuation times obtained by both the algorithms. The study showed that the results obtained by branch and bound algorithms for the BBEP with perfect information are always better than that of tabu search for the RBBEP with uncertainty in the case of minimum, average and maximum evacuation times. In this study they found that the choice of number of sources and sinks have not played significant role in both approaches.

This study motivated to conduct a number of researches like multi-depot multi-model evacuation planning with car and bus based evacuation and

with contra flow. The authors have also recommended other case studies in Kathmandu valley with increased number of parameters such as buses, sources, sinks and depots.

#### **Conclusion**

In bus based evacuation evacuees who do not travel on their own due to age, sickness, handicapped, children, tourists or who do not have their own car are gathered at few collection points, where they are brought on buses to take them to safe region during emergency. Bus based evacuation planning (BBEP) is more essential for developing country like Nepal because most of the people in Nepal cannot afford their own car and are bus-based. BEP is difficult to solve because it is interrelated with construction of routes, assignment of routes to the multiple vehicles and selection of a shelter for each route. There have been only few studies regarding bus based evacuation. We have applied branch and bound and Bellman's equation which is related to dynamic programming with some modification to solve the problem. For further study we can extend above mentioned model by using bus accommodated with wheelchair, considering pickup time and shelters with special facilities.

#### **ACKNOWLEDGMENTS:**

The research of Shree Ram Khadka was supported by the European Commission in the framework of Erasmus Mundus and within the project cLINK and Kantipur Engineering College.

#### **REFERENCES**

- Bish, D. R.. 2011. Planning for a bus-based evacuation. *OR Spectrum*, **33**(3):629-654. doi: 10.1007/s00291-011-0256-1.
- Egon, B. and Paolo, T. 1983. Branch and bound methods for the traveling salesman problem.

- Report. Carnegie Mellon University. Graduate School of Industrial Administration.
- Goerigk, M., Grun, B. and Hebler, P. 2013. Branch and bound algorithm for the bus evacuation problem. *Computers & Operations Research* **40** (12):3010-3020.
- Goerigk, M. and Grun, B. 2014. A robust evacuation model with delayed scenario information. or *Spectrum*, **36** (4):923-948.
- Jens, C. 1999. Branch and bound algorithms-principles and examples. Technical report. University of Copenhagen.
- John. D. C., Murty, K. G., Sweeney, D. W. and Karel, C. 1963. An algorithm for the traveling salesman problem. *Operations Research* **11**(6): 972– 989. doi:10.1287/opre.11.6.972.
- Land H. A. & Doig A. G. 1960. An automatic method of solving discrete programming problems. *Econometrica* **28**(3): 497–520. doi:10.2307/1910129.
- Liman, T. 2006. Lessons from Katrina and Rita: what major disasters can teach transportation planners, *Journal of Transportation Engineering* (<http://scitation.aip.org/teo>), **132**: 11-18.
- Papadimitriou, C. H. and Steiglitz, K. 2006. Combinatorial Optimization: Algorithms and Complexity. *Journal of Universal Computer Science* **12**(5):437-438.
- Pedregal, P. 2004. Introduction to optimization: Texts in applied mathematics. New York: Springer. 178-179.
- Pyakurel, U., Goerigk, M., Dhamala, T.N. and Hamacher, H. W. 2016. Transit Dependent Evacuation Planning for Kathmandu Valley: *International Journal of Operational Res. Nepal* **5**(1):49-73.
- Wikipedia, the free encyclopedia. July 12, 2016. [https://en.wikipedia.org/wiki/Branch\\_and\\_bound](https://en.wikipedia.org/wiki/Branch_and_bound).