Improving the connectivity of a bus system: A case study of Ho Chi Minh city

Van Son NGUYEN
Academy of Cryptography Techniques
141 Chien Thang street, Hanoi, Vietnam
sonnv188@gmail.com

Quang Dung PHAM
Hanoi University of Science and Technology
School of Information and Communication Technology, Hanoi, Vietnam
dungpq@soict.hust.edu.vn

Minh Hoang HA
FPT Technology Research Institute, FPT University, Hanoi, Vietnam
hoanghm@fpt.edu.vn

ABSTRACT

Public transportation modes like buses plays an important role in our real life. Designing a good bus system can bring advantages to both passengers and the society. One of key design criteria is the connectivity which can be measured by the minimum number of bus lines a passenger must take to travel from one station to another. This paper considers the bus system design problem with the objective of improving the connectivity of the system from the existing infrastructure. We defined the problem in the form of an optimization problem and propose an algorithm for the solution. Results obtained from the real instances derived from the bus system of Ho Chi Minh city are analysed and reported.

CCS Concepts

•**Applied computing** → **Transportation;**

Keywords

Optimization, Public Transportation Design, Bus System, Algorithms

1. INTRODUCTION

Cities have now become bigger and bigger in terms of surface and population. This phenomenon has caused a number of consequences: severe traffic congestion, noise, pollution, road accidents, etc. Public transport is considered as one of the most efficient solutions for these issues, since it allows more efficient movements across a city. However, public transport systems often face a not-easy-to-solve trade-off between the service quality and the cost. In other words, these systems have to provide an enough good service to attract passengers, while at the same time their price must be acceptable for the low income segments of the population. To achieve this, public systems need to have a good design, an efficient operational planning and a reasonable real-time control strategy.

Vietnam has now confronted with many issues from public transportation systems. The main public transport mode in cities of Vietnam is buses. In Ho Chi Minh city, although the number of passengers by buses has increased rapidly (8.4 times from 2002 to 2009), the bus system fulfils only 7.3 % of traveling demand of people [1]. Moreover, according to a report of Ho Chi Minh Department of Transport, the number of passengers, has decreased recently because of system's inconvenience (please cite here which report). One of the main reasons for this is due to the backward bus network which could not catch the growth speed of the city. A clear indication is that, people quite often have to take too many buses to travel between two points in the network. This is very time consuming and costly for passengers who currently have to pay a ticket for each bus line.

To solve the problem, the city should redesign its bus network. In the literature, the authors of [2] propose a bi-level optimization model to tackle the Transit Network Re-Design problem. In the upper level, network structures are determined based on a program which minimizes the weighted sum of use cost, operator cost, and unsatisfied demand, while in the lower-level a transit assignment problem is solved. A genetic algorithm is developed for the solution and numerical results show that the spatial equity increases the total travel cost in the network. For more information on bus network design problem, the readers are recommended to the review article of [3].

All the researches in the literature on the bus network design problem have been carried out with the hypothesis that demands between any two points are known (see [3] for more information). But this is not the case of Ho Chi Minh city where the demand estimation is an impossible task due to lack of resources. Therefore, results of Lampkin and Saalmans [4] and a number of applications as Whye-Liang Chia [5], C. E. Mandl [6] is applied to both the assignment of passengers or buses to routes is not feasible. Instead of allocating again, in a simpler manner, the city could add some more bus lines to reduce the number of corresponding stations a passenger passing through to travel from any origin to any destination. Opening new bus lines with new stations which require investment on land is very expensive [1]. Therefore, a more pertinent way is to reuse existing bus stations to construct new lines. Adding the new bus lines should respect several constraints. Firstly, because of the

SoICT '16, December 08-09, 2016, Ho Chi Minh City, Viet Nam

DOI: http://dx.doi.org/10.1145/3011077.3011094

limited resources, the number of added lines must be less than a given value. Secondly, the number of bus stops in a new line must be limited by a predefined parameter to avoid challenges related to operational management.

In this paper, we firstly measured the efficiency of a bus network by an indicator called connectivity level. It is defined as a minimum number of bus lines which an user has to take in order to travel between any pairs of stations in the network. We then studied the problem of redesigning the bus network by improving its the connectivity level. Our contributions are that we introduce a new bus network redesign problem when the demands are unknown and propose an algorithm to solve the problem. We also report and analyse computational results obtained on a real-world problem of Ho Chi Minh city.

The remainder of the paper is organized as following. Section 2 describes some notations, defines the problem and analyses the connectivity level of a bus system. Our algorithm is presented in Section 3. Section 4 discusses the computational results and Section 5 summarizes our conclusions.

2. PROBLEM FORMULATION

The bus network redesign problem in this paper is defined as follows. Let $V = T \cup N$ be the set of bus stops where T is the set of terminals at which the bus starts and terminates its itinerary and N is the set of intermediate bus stations. Define the connectivity level (or *level* for short) of two points u and v the minimum number of bus lines a passenger must take to travel from station u to station v and vice-versa. We then defined the connectivity level (or simply *level*) of the bus network the maximum level of any pairs of its stations. The objective of the problem is to determine a limited number of new bus lines including existing stations so that (1) each is not too long, that includes no more than a predefined number of stations; (2) the level of the resulting system reduces to a given number γ and (3) the number of pairs of bus stations having maximal level is minimized. Before formulating the problem, we introduce some definitions and notations as follows.

2.1 Definition and notations

- $\mathcal{R} = \{\langle s, x_1, \ldots, x_k, t\rangle \mid s, t \in T \wedge x_1, \ldots, x_k \in V \wedge 0 \leq k \leq |V| - 2\}$ is the set of all possible designed bus lines. A bus line is a sequence of stops in which the starting and terminating stops are terminals and the remaining stops are intermediate bus stations. The length of a bus line br, denoted by $l(br)$, is the number of its bus stops of r.
- A bus system $\mathcal{B}$ is defined to be a tuple $\mathcal{B} = (T, N, \mathcal{L})$ where $\mathcal{L} \subseteq \mathcal{R}$
- Given $br \in \mathcal{R}$, we denote $s_{br}(u_i, u_j) = \langle u_i, u_{i+1}, ..., u_j\rangle$ a sub-line of the bus line where $i, j = \overline{1, k}, i < j$.
- $S(br_i, br_j)$: set of common stops of br_i, br_j, $\forall br_i, br_j \in \mathcal{R}, i \neq j$.
- $I(x, br)$: index of bus stop x on bus line br (start station of br is indexed by 1). If the stop x is not in the bus line br, then x, $I(x, br) = 0$ by convention.

Given a bus system $\mathcal{B} = (T, N, \mathcal{L})$, $V = T \cup N$

- A route from station u to station v of the bus system $\mathcal{B}$ ($\forall u, v \in V$) is a sequence $r = r(u, v) = \langle u = x_0, r_1, x_1, r_2, x_2, ..., r_m, x_m = v\rangle$ where $r_i \in \mathcal{L}, x_i \in S(r_i, r_{i+1})$, $I(x_i, r_{i+1}) < I(x_{i+1}, r_{i+1})$, $\forall i \in \overline{1, m-1}$. Intuitively, a route is a way that people travel from one stop to another (i.e., people start from stop x_0 and take bus line r_1 to stop x_1, and then take the bus line r_2 to the stop x_2, etc.). We have a notation $br(r)$ which is a sequence of stops according to r: $br(r) = s_{r_1}(x_0, x_1)$ :: $s_{r_2}(x_1, x_2)$:: ... :: $s_{r_m}(x_{m-1}, x_m)$.
  We denote $level(r) = m$ the level of route r which is the number of bus lines that people take to travel on the route r $level(r) = m$.
- The length of a route is defined as the number of stops on the route: $l_{r(u,v)} = (I(x_1, r_1) - I(u, r_1)) + (I(x_2, r_2) - I(x_1, r_2)) + ... + (I(v, r_m) - I(x_m, r_m)) + 1$.
- We denote $l(\mathcal{B}) = \max\{l(br) \mid br \in \mathcal{L}\}$.
- $R(u, v, \mathcal{B})$: set of possible routes from stop u to stop v in the bus system $\mathcal{B}$.
- $level(u, v, \mathcal{B}) = \min\{level(r) | \forall r \in R(u, v, \mathcal{B})\}$. Intuitively, the *level* of a pair of bus stops u and v is the minimum number of bus lines that people have to take in order to travel from u to v.
- We denote $r(u, v, \mathcal{B})$ is the route $r \in R(u, v, \mathcal{B})$ having $level(r)$ equal to $level(u, v, \mathcal{B})$.
- $level(\mathcal{B}) = \max\{level(u, v, \mathcal{B}) \mid u, v \in V\}$
- $o(\mathcal{B}) = \sharp\{(u, v) \in V \mid level(u, v, \mathcal{B}) = level(\mathcal{B})\}$

An example is shown in Figure 1.
Where:

- $T = \{1, 7, 8, 10, 11, 12, 15, 16, 19, 20\}$.
- $N = \{2, 3, 4, 5, 6, 9, 13, 14, 17, 18, 21\}$.
- $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21\}$.
- $\mathcal{R} = \{br_1 = \langle 1, 2, 3, 4, 5, 6, 7\rangle, br_2 = \langle 8, 9, 2, 11\rangle, br_3 = \langle 12, 13, 4, 14, 15\rangle, br_4 = \langle 16, 3, 15, 17, 18, 19\rangle, br_5 = \langle 20, 18, 21, 10\rangle\}$. There, $s_{br_1}(2, 5) = \langle 2, 3, 4, 5\rangle, s_{br_3}(12, 14) = \langle 12, 13, 4, 14\rangle, ...$
  Given $\mathcal{B} = (T, N, \mathcal{L})$, where $\mathcal{L} \equiv \mathcal{R}$, $V = T \cup N$
- $S(br_1, br_2) = \{2\}$, $S(br_1, br_3) = \{4\}$, $S(br_1, br_5) = \{\emptyset\}$, ...

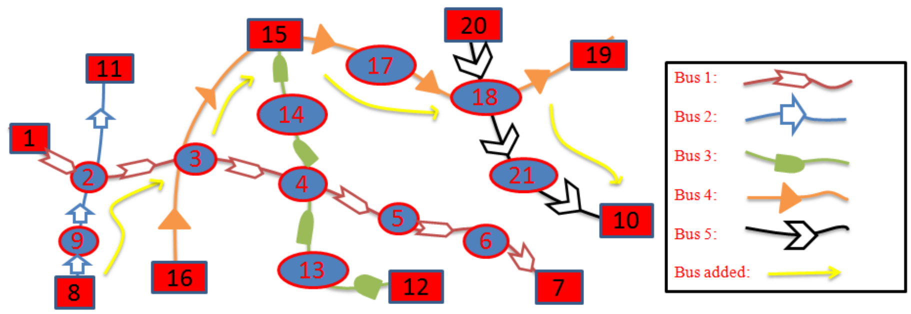

Figure 1: An example of the bus system.

- $I(1, br_1) = 1, I(4, br_1) = 4, I(14, br_3) = 4, I(19, br_4) = 6, \ldots$

- $R(1, 5, \mathcal{B}) = \{r^1\}$, where $r^1 = \langle 1, br_1, 5\rangle$. We have $level(r^1) = 1$. Therefore, $level(1, 5, \mathcal{B}) = 1$ and $r(1, 5, \mathcal{B}) = r^1 = \langle 1, br_1, 5\rangle$.

- $R(1, 17, \mathcal{B}) = \{r^1, r^2\}$ where $r^1 = \langle 1, br_1, 3, br_4, 17\rangle, r^2 = \langle 1, br_1, 4, br_3, 15, br_4, 17\rangle$. We have $level(r^1) = 2, level(r^2) = 3$. Then obviously, $level(1, 17, \mathcal{B}) = 2$, $r(1, 17, \mathcal{B}) = r^1 = \langle 1, br_1, 3, br_4, 17\rangle$.

- $R(8, 10, \mathcal{B}) = \{r^1, r^2\}$, where $r^1 = \langle 8, br_2, 2, br_1, 3, br_4, 18, br_5, 10\rangle$, $r^2 = \langle 8, br_2, 2, br_1, 4, br_3, 15, br_4, 18, br_5, 10\rangle$. We have $level(r^1) = 4, level(r^2) = 5$. Therefore, $level(8, 10, \mathcal{B}) = 4$ and $r(8, 10, \mathcal{B}) = \langle 8, br_2, 2, br_1, 3, br_4, 18, br_5, 10\rangle$.

- $R(8, 21, \mathcal{B}) = \{r^1, r^2\}$, where $r^1 = \langle 8, br_2, 2, br_1, 3, br_4, 18, br_5, 21\rangle$, $r^2 = \langle 8, br_2, 2, br_1, 4, br_3, 15, br_4, 18, br_5, 21\rangle$. We have $level(r^1) = 4, level(r^2) = 5$. Therefore, $level(8, 10, \mathcal{B}) = 4$ and $r(8, 21, \mathcal{B}) = r^1 = \langle 8, br_2, 2, br_1, 3, br_4, 18, br_5, 21\rangle$.

- $level(\mathcal{B}) = 4$, $o(\mathcal{B}) = 4$.

2.2 Connectivity analysis

Users have several requirements for the services they would choose such as fast, convenient and comfortable. Designing a good bus system can bring advantages to passengers. As mentioned above, in this research, we tried to improve the performance of an existing bus system by adding new bus lines to reduce its connectivity. In this paper, the connectivity is represented by *level*. If the *level* is high, the passenger may need many bus lines to travel and the connectivity of the system is considered as *weak*. On the other hand, if the *level* is low, the passenger has to take less bus to travel and the connectivity of the system is *strong*.
In the experiments, we collected data and analyzed the connectivity of the Ho Chi Minh bus system. Data is shown in Table 1.

Total bus lines ($\lvert\mathcal{L}\rvert$)	119
Total stations ($\lvert V\rvert$)	2115
Number of terminal stations ($\lvert T\rvert$)	92
The length of the longest bus line	75
The length of the shortest the bus line	8

Table 1: Data of the bus system of Ho Chi Minh city

We applied a Breadth-First-Search algorithm for computing the minimal number of bus lines necessary to travel from one station to another. Figure 2 represents the number of pairs of stations at different levels in the current bus system of Ho Chi Minh city. We observed from the analysis that the level of the bus system of Ho Chi Minh city is currently equal to 6 and there are 13,031 pairs of two bus stops having level greater or equal to 4.

2.3 Problem formulation

The objective of this paper is to improve the connectivity of the existing bus system. To achieve this goal without perturbing the system too much, our approach is to add some other bus lines to reduce the connectivity level of the bus system. Obviously, the more bus lines added to the system, the more reduction of the level we get. However, we cannot add too many lines due to high operational cost. Therefore, the number of bus lines added should be limited by an upper bound. We formulated the problem as follows. Given a bus system $\mathcal{B} = (T, N, \mathcal{L})$ and two constants λ and γ, find a subset of bus lines $X \subseteq \mathcal{R} \setminus \mathcal{L}$ such that

- $|X| \leq \lambda$
- $level(\mathcal{BX}) \leq \gamma$
- $l(\mathcal{B}) \leq \beta$
- $o(\mathcal{BX})$ is minimized

where $\mathcal{BX} = (T, N, \mathcal{L} \cup X)$. Intuitively, we search to add no more than λ new bus lines to the system so that its level is less than or equal to γ and the number of pairs of bus stops having the maximal level is minimal.

3. PROPOSED ALGORITHM

The goal is to find a set of no more than λ new bus lines to add to the bus system in order to improve the connectivity. The number of such possibilities is exponentially increased with the number of bus stops. We cannot perform an exhaustive search to achieve the goal. Therefore, we apply the search on specific possibilities. Suppose that, the level of the current bus system $\mathcal{B}$ is m. Let $r = \langle x_0, r_1, x_1, \ldots, r_m, x_m\rangle$ be a route with $level(r) = m$. Clearly, if we add a bus line that starts from some terminal, and some tops, and then all the stops of r, and terminates at some terminal, then the *level* of r and many other pairs of stops of r becomes 1. This is the key idea of our proposed algorithm. Given two routes:

$r(u_1, v_1) = \langle u_1, r_1, x_1, r_2, x_2, ..., r_m, v_1\rangle \in R(u_1, v_1, \mathcal{B})$,
$r(u_2, v_2) = \langle u_2, r_1, x_1, r_2, x_2, ..., r_m, v_2\rangle \in R(u_2, v_2, \mathcal{B})$

We denote $r(u_1, v_1) \ll r(u_2, v_2)$ if $I(u_2, r_1) \leq I(u_1, r_1)$ and $I(v_1, r_m) \leq I(v_2, r_m)$. Intuitively, if there is a bus travelling along $r(u_2, v_2)$, then all bus tops of route $r(u_1, v_1)$ can also be serviced by this bus in a direct way (only one bus line). We say "$r(u_1, v_1)$ is dominated by $r(u_2, v_2)$".

One of the phases of our algorithms is to find a set of routes that can be candidates to establish a new bus line. These are routes having maximum level among all possible routes. This phase is depicted in Algorithm 1. Note that in this algorithm, lines 7–11 ignore routes dominated by routes in the computed candidate (this is to avoid the redundancy).

Algorithm 1: FindCandidateRoutes($\mathcal{B} = (T, N, \mathcal{L}), P$)

Input: A bus system $\mathcal{B} = (T, N, \mathcal{L})$ and a set P of pairs of bus stops ($P \subseteq \{(u, v)|u, v \in T \cup N \wedge u \neq v\}$)
Output: L_R: Set of candidate routes for establishing new bus lines.

1 **foreach** $(u, v) \in P$ **do**
2   Compute $level(u, v, \mathcal{B})$ and a route $r(u, v, \mathcal{B})$ using Breadth-First-Search algorithm;
3 $l_{max} \leftarrow max\{level(u, v, \mathcal{B})|(u, v) \in P\}$;
4 **if** $l_{max} \leq \gamma$ **then**
5   **return** $\emptyset$;
6 $L_R \leftarrow \{r(u, v, \mathcal{B})|(u, v) \in P \wedge level(u, v, \mathcal{B}) = l_{max}\}$;
7 $TR \leftarrow \emptyset$;
8 **foreach** $r(u, v, \mathcal{B}) \in L_R$ **do**
9    $RI(r(u, v, \mathcal{B})) \leftarrow \{r'(u, v, \mathcal{B})|r'(u, v, \mathcal{B}) \in L_R, r'(u, v, \mathcal{B}) \ll r(u, v, \mathcal{B})\}$;
10    $TR \leftarrow TR \cup RI$;
11 $L_R \leftarrow L_R \setminus TR$;
12 **return** L_R;

In the example described in Figure 1, we have $level(\mathcal{B}) = 4$ and $L_R = \{r(8, 10, \mathcal{B}),\ r(8, 21, \mathcal{B}),\ r(9, 10, \mathcal{B}),\ r(9, 21, \mathcal{B})\}$. However, $RI(r(8, 10, \mathcal{B})) = \{r(8, 21, \mathcal{B}), r(9, 10, \mathcal{B}), r(9, 21, \mathcal{B})\}$. Hence, we have output $L_R = \{r(8, 10, \mathcal{B})\}$.

A bus line must be started and terminated with stops in the terminal set T, and should not be too long. So, the output of this phase will be extended and post-processed as follows:

- Terminal points of the routes: If the origin or destination of chosen routes is not the same as the system's terminal points, we will find the nearest terminal points of the system and then update the routes. An example is shown in Figure 3 and the algorithm is presented in Algorithm 2, Algorithm 3.

- Length of the routes: If the length of an added route is too long (i.e., greater or equal to $2 \times l(\mathcal{B})$), it will be cut into two different routes and the terminals points will be updated. This is depicted at line 8 to 17 in the Algorithm 4.

Algorithm 2: ExtendStartingPoint($r(u, v), L$)

Input: $r(u, v) = \langle u, r_1, x_1, r_2, x_2, ..., r_m, v\rangle$
Output: $r(a, b)$: Extended route with starting terminal

length $\leftarrow +\infty$;
$r(a, b) \leftarrow r(u, v)$;
foreach $br = \langle s, u_1, u_2, ..., u_k, t\rangle \in L$ **do**
   if $I(u, br) > 0$ *and* $I(u, br) <$ *length* **then**
      $r(a, b) \leftarrow \langle s, br, u, r_1, x_1, r_2, x_2, ..., r_m, v\rangle$;
      length $\leftarrow I(u, br)$;
return $r(a, b)$;

Algorithm 3: ExtendTerminatingPoint($r(u, v), L$)

Input: $r(u, v) = \langle u, r_1, x_1, r_2, x_2, ..., r_m, v\rangle$
Output: $r(a, b)$: Extended route with ending terminal

length $\leftarrow +\infty$;
$r(a, b) \leftarrow r(u, v)$;
foreach $br = \langle s, u_1, u_2, ..., u_k, t\rangle \in L$ **do**
   if $I(v, br) > 0$ *and* $I(t, br) - I(v, br) <$ *length* **then**
      $r(a, b) \leftarrow \langle u, r_1, x_1, r_2, x_2, ..., r_m, v, br, t\rangle$;
      length $\leftarrow I(t, br) - I(v, br)$;
return $r(a, b)$;

The overall algorithm is depicted in Algorithm 5 in which line 2 initializes a stack Ω to store bus systems and a set of pairs of two bus stops which will be considered as candidates for establishing a new bus line during the search. At each step, line 6 retrieves the bus system(L^i) and the set of pairs(P^i). Lines 7–8 compute a set of candidate bus lines based on the pairs of P^i. Lines 9–17 explore all the candidates and try to add a new bus line. Lines 12–14 update the best solution if one found.

4. EXPERIMENTS

The Ho Chi Minh bus map provided by [7] was used in this research. By means of Google Map, we collected all bus lines and all stations with names, latitudes, longitudes, etc. Analyzing the system's connectivity, finding and choosing bus lines were automatically implemented in Java. The experiments ware conducted on a machine $Intel(R)Core(TM)i3-4005UCPU@1.70GHz(4CPUs), \sim 1.7GHz$ and RAM is $4GB$.

We carried out experiments with different values of parameters λ and γ. The value of β is set to $2 \times l(\mathcal{B})$. The obtained results are presented in Table (2). The column $level(\mathcal{BX})$ represents the level of the resulting bus system while the column $o(\mathcal{BX})$ reports the number of stations with the level equal to $level(\mathcal{BX})$.

The Table 2 shows that with $\lambda \leq 3$ and $\gamma = 4$, our algorithm cannot find any solution. When γ is set to 5, the number of stations having the maximal level is significantly reduced from 2434 to 90 if we add two new lines instead of one to the existing system. However, if we add three new lines, the number of stations having the maximal level is slightly reduced to 78. Thus, in the case where we want to reduce the level of the system from 6 to 5, adding 2 new lines should be the most reasonable solution. And finally, the experiments show that our algorithm is able to find and add 4 new bus lines so that the level of the bus system is re-

Algorithm 4: UpdateRoutes($L_R, B = (T, N, L)$)

Input: A set of routes L_R and a bus system B
Output: L_R: List routes with reduced lengths

$TS \leftarrow \emptyset$;
foreach $r(u, v) = \langle u, r_1, x_1, r_2, x_2, ..., r_m, v\rangle \in L_R$ **do**
  $r(a, b) \leftarrow r(u, v)$;
  if $u \notin T$ **then**
    $r(a, b) \leftarrow$ ExtendStartingPoint($r(a, b), L$);
  if $v \notin T$ **then**
    $r(a, b) \leftarrow$ ExtendTerminatingPoint($r(a, b), L$);
  $TS \leftarrow TS \cup \{r(a, b)\}$;
  if $l_{r(a,b)} \geq 2 \times l(B)$ **then**
    $k \leftarrow m/2$;
    $r(a_1, b_1) \leftarrow \langle a, r_1, x_1, r_2, x_2, ..., r_k, x_k\rangle$;
    $r(a_2, b_2) \leftarrow \langle x_{k+1}, r_{k+2}, x_{k+2}, ..., r_m, b\rangle$;
    if $x_k \notin T$ **then**
      $r(a_1, b_1) \leftarrow$ ExtendTerminatingPoint($r(a_1, b_1), L$);
    if $x_{k+1} \notin T$ **then**
      $r(a_2, b_2) \leftarrow$ ExtendStartingPoint($r(a_2, b_2), L$);
    $TS \leftarrow TS \setminus \{r(a, b)\} \cup \{r(a_1, b_1), r(a_2, b_2)\}$;
return TS;

Algorithm 5: ImproveConnectivity($\mathcal{B} = (T, N, \mathcal{L})$)

Input: Existing bus system $\mathcal{B} = (T, N, \mathcal{L})$
Output: List of new bus lines added to the system

$P \leftarrow \{(u, v) | \forall u, v \in T \cup N\}$;
Stack $\Omega \leftarrow \{(\mathcal{L}, P)\}$;
$L^* \leftarrow \mathcal{L}$;
$f^* \leftarrow o(\mathcal{B})$;
$l_{min} \leftarrow \gamma$;
while $\Omega \neq \emptyset$ **do**
  $(L^i, P^i) \leftarrow$ Pop(Ω);
  $L_R \leftarrow$ FindCandidateRoutes($(T, N, L^i), P^i$);
  $L_R \leftarrow$ UpdateRoutes($L_R, (T, N, L^i)$);
  foreach $r \in L_R$ **do**
    $L' \leftarrow L^i \cup \{br(r)\}$;
    $B' \leftarrow (T, N, L')$;
    if $level(B') \leq l_{min} \vee (level(B') == l_{min} \wedge o(B') < f^*)$ **then**
      $L^* \leftarrow L'$;
      $f^* \leftarrow o(B')$;
      $l_{min} \leftarrow level(B')$;
    if $|L'| - |\mathcal{L}| < \lambda$ **then**
      $P' \leftarrow \{(u, v) \mid (u, v) \in P^i \wedge level(u, v, (T, N, L^i)) \geq \gamma)\}$;
      Push(L', P', Ω);
return L^*;

duced to 4 and the number of pairs of two bus stops having level 4 is 20,591. Figure 4 represents the number of pairs of stations at different levels in the updated bus system of Ho Chi Minh city.

λ	γ	$level(\mathcal{BX})$	$o(\mathcal{BX})$
$\leqslant 3$	4	–	–
1	5	5	2434
2	5	5	90
3	5	5	78
4	4	4	20,591

Table 2: Results with different values of parameters

5. CONCLUSION

We addressed in this paper a problem of redesigning the bus system taking into account the connectivity of the bus system. The connectivity is measured as the minimum number of bus lines people have to take to travel between any pairs of bus stations in the network. We formulated the problem as an optimization problem and proposed an algorithm for solving it. The key idea of the proposed algorithm is to select and add some new bus lines based on the routes among pairs of two stops where people have to take the largest number of bus lines to travel. We considered the case study derived from the bus system of Ho Chi Minh city with 2115 stations and 119 bus lines and the connectivity equal to 6. Our experiments show that we can add 4 new bus lines to reduce the connectivity to 4 and the number of pairs of bus stops having level 4 is 20,591. For future works, we continue to develop algorithms improving the connectivity of the bus system taking into account both adding new bus lines and removing redundant existing bus lines.

6. ACKNOWLEDGMENTS

This paper was sponsored by Vietnamese National Foundation for Science and Technology Development (project FWO.102.2013.04).

7. REFERENCES

[1] Trinh Viet Dung, 2011. Van tai hanh khach cong cong tai thanh pho Ho Chi Minh, Tap chi khoa hoc xa hoi so 6 (154) - 2011.

[2] Fan, W., Machemehl, R., 2011. Bi-level optimization model for public transportation network redesign problem. Transportation Research Record 2263, pp 151–162.

[3] Ibarra-Rojas, O.J., Delgado, F., Giesen, R., MuÃśoz, J.C., 2015. Planning, operation, and control of bus transport systems: a literature review. Transportation Research Part B 77, pp 38–75.

[4] W. Lampkin and P.D. Saalmans, The design of routes, service frequencies and schedules for a municipal bus undertaking: A case study, Operational Res. Quart. 18 (1967) 375-397.

[5] A. Whye-Liang Chia, Identifying service improvements in urban bus operations, MS Thesis, Massachusetts Institute of Technology, Cambridge, MA (1979).

[6] C. E. Mandl. Evaluation and optimization of urban public transportation networks. European J. Oper. Res. 5:396–404, 1980.

[7] The Center Of Management And Public Transport. http://www.buyttphcm.com.vn.

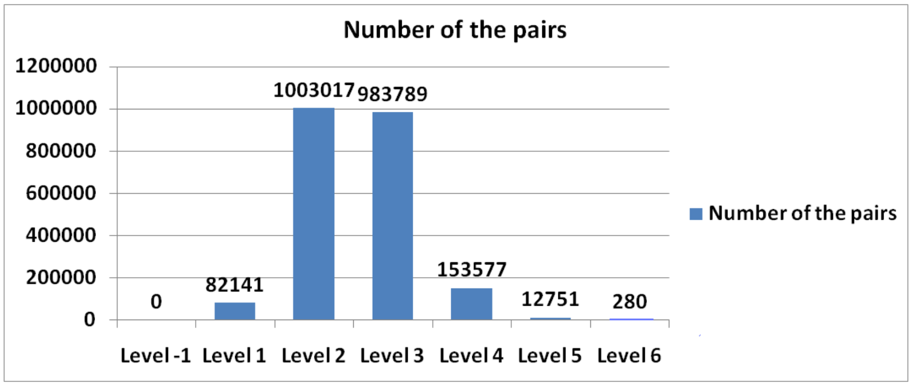

Figure 2: Number of pairs of stations at different levels.

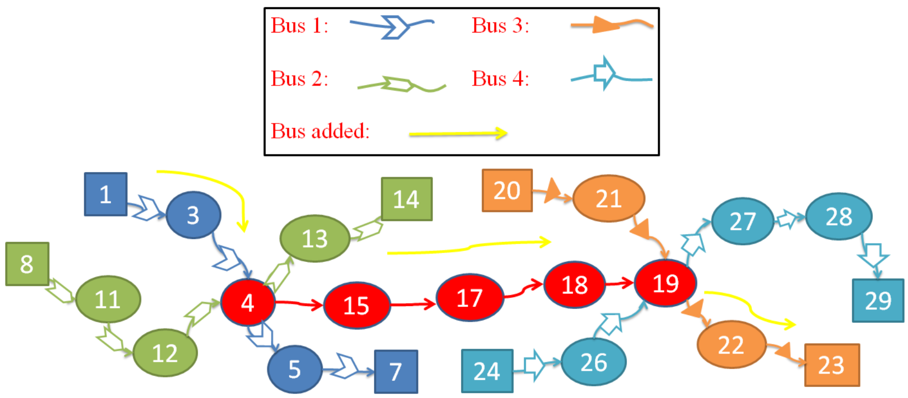

Figure 3: An example which describes how to find the nearest terminal points of the system and update the output routes of phase 1.

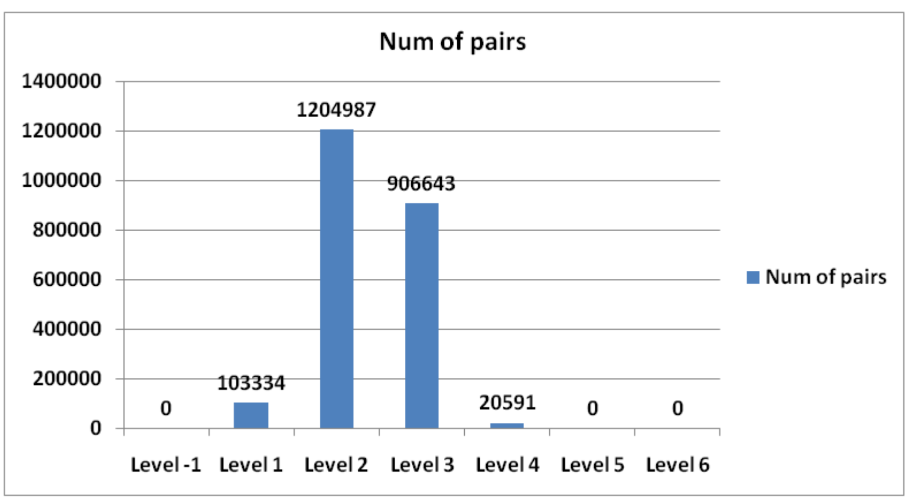

Figure 4: Number of pairs of stations at each level in the updated system.