A School Bus Routing And Scheduling Problem

Wolfgang Lindenberg*
Frantisek Fiala**

SCS-TR-5
July 1982

* Institut für Mathematik, Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD), Schloss Birlinghoven, Postfach 1240, D5205 St. Augustin 1, Federal Republic of Germany.

** Department of Mathematics and Statistics/School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada.

INTRODUCTION

This is a preliminary report on research work being done jointly during the last years on the school bus routing and scheduling problem. The first author describes here the methodology he developed for solving the problem. It serves as a basis for a computer system designed and implemented. A full report describing the system is in preparation and will appear soon.*)

In an empirical study of the economy of student transportation in rural regions of the Federal Republic of Germany [1] a conclusion was made that the main cost factor in using school buses was not so much the total number of kilometers the buses travel but the total number of buses used for the transportation. In other words: the smaller the number of buses needed, the lower the expected operating cost. If a change of existing bus routes leads to a reduction of buses required then it leads also to a lowering of the expected operating cost. For example in [1], a school district is described which is actually served by 22 buses and which could be served by only 14 buses (selected from the old bus fleet) if routes are appropriately modified.

These findings provided motivation for a detailed mathematical analysis of the problems related to the use of school buses in rural areas. Its results are presented in this report. The requirement of a minimum size bus fleet actually used is the only minimization criterion here. In all school bus routing systems developed previously, mainly in North America, the objective is the minimization of the total distance traveled by all buses and the fleet size plays a secondary role. Therefore there is a need for research aimed at development of methods for determination of the minimum size school bus fleet based on graph theoretical and combinatorial principles. The algorithms developed should serve as a basis for a computer system which can be written and implemented.

Such a system has been developed in cooperation between GMD and Carleton University. Most of the research leading to the design, analysis, implementation, and testing was done in and supported by GMD, part of it at Carleton University where it was supported by NSERC of Canada, Grant A8233.

---

*) The research described here originated from a previous cooperation between the Fachbereich Rechts- und Wirtschaftswissenschaften der Universitat Mainz and the Institut fur Mathematik (IMA) der GMD, St. Augustin.

This report contains a summary of the basic methodology used in the system and a description of the key points which must be taken into account in solving an optimization problem of this type. A full report containing all the details will follow.

TABLE OF CONTENTS

I. DESCRIPTION OF THE PROBLEM

Let L_0 be a location of a school and let $L_1, \ldots, L_n$ be all locations in which students attending the school located in L_0 live. The students are transported to the school by buses selected from a given bus fleet **C**.

We are looking for:

a) The **minimum number** N of buses from **C** needed for transportation of all students from L_i, $i=1, \ldots, n$, to the school.

b) A subset **B** of **C** containing N buses which can actually perform the transportation.

c) A set of routes and time tables for all buses in **B**.

We make the following assumptions:

P1: The classes start at the same time T_{init} for all students from L_i, $i=1, \ldots, n$.

P2: Every student must be transported to the school within a given maximal riding time T_{max}.

P3: The maximal waiting time, WT_{max}, between the arrival to the school and the beginning of classes cannot be violated for any student.

P4: Every location L_i, $i=1, \ldots, n$, has exactly one bus stop assigned to it.

P5: The bus fleet **C** is composed of buses with different bus capacities.

P6: Every bus from **C** can start its route at any location and at any time.

The maximal riding time T_{max} given in P2 corresponds to the maximal time a bus can spend on a route between the moment when the first student enters the bus and its arrival to the school. On the other side, WT_{max} given in P3 determines available time within which a bus can make **a second tour** starting and ending at the school. Assumption P4 corresponds well to the situation in **rural areas.** Assumption P5 allowing the use of **buses with different capacities** complicates the whole problem significantly and has not been treated often in the literature. Its handling requires the development of new combinatorial techniques which help to find a solution. By assumption P6 it is required that any bus from the bus fleet **C** can be available at any given location at any given time to start its route to the school. Note that we do not make any assumption about a location of a bus depot.

The school bus routing and scheduling problem treated here belongs to the class of problems for which no polynomial algorithm giving an exact solution is known. Therefore we restrict ourselves to a development of **heuristic methods.**

The underlining network formed by the locations and the connecting roads can be naturally associated with a labeled and weighted (undirected) graph $G=(V,E)$ in the following manner:

a) Vertices v_i from V correspond to locations L_i, $i=1, \ldots, n$, containing students to be transported.
b) Vertex v_0 from V corresponds to school location L_0.
c) (v_i,v_j) is an edge in E only if there is a direct road between L_i and L_j.
d) Every vertex v_i from V, $1 \leq i \leq n$, is labeled by the total number of students who should be transported from location L_i to the school.
e) Every edge (v_i,v_j) from E is weighted by the distance d_{ij} between locations L_i and L_j.

Assume that the set of available bus capacities $\mathbf{C}=\{c_1, \ldots, c_m\}$ is totally ordered:

(1) $$c_1 \geq c_2 \geq \ldots \geq c_j \geq \ldots \geq c_m.$$

Our problem consists in the determination of an appropriate subset $\mathbf{C'}=\{c'_1, \ldots, c'_N\} \subseteq \mathbf{C}$ such that we can transport all the students to the school with the buses from $\mathbf{C'}$ under the assumptions P1 - P4 and P6. In addition, the number N is minimal. As we proceed in determining the set $\mathbf{C'}$, we simultaneously generate time tables for all buses in $\mathbf{C'}$.

II. GENERAL NETWORK CONSIDERATIONS

Using assumption P3 and the average speed of the buses we can identify the set of locations around the school (and different from it) which can be reached in time $T < WT_{max}$. The corresponding set $\mathbf{V_0}$ of vertices in G generates a partition of $V-\{v_0\}$ into two disjoint subsets $\mathbf{V_0}$ and $\mathbf{V_1} = (V-\{v_0\})-\mathbf{V_0}$. We also say that $\mathbf{V_0}$ is the **inner district** of the school and $\mathbf{V_1}$ is the **outer district** of the school. Let $S = S(\mathbf{V_1})$ be the total number of students assigned to vertices of $\mathbf{V_1}$.

From (1) follows that if there exists a k such that

(2) $$\sum_{i=1}^{k-1} c_i < S \leq \sum_{i=1}^{k} c_i \quad , \quad k \leq m,$$

then there are at least k buses needed for the transportation. If

$$S > \sum_{i=1}^{m} c_i$$

then the problem has no solution.

The lower bound k easily determined from (2) is independent on the structure of the graph. But the actual number of buses needed depends on both the structure of the graph and the composition of the available bus fleet. This is illustrated by the following examples. The graphs of Fig. 1a and 1b correspond to a network with a school and 7 additional locations. The solid lines represent the shortest paths from nodes v_1 and v_5 to the school. The vertex labels (in brackets) give the numbers of students and the edge weights indicate the distance of two locations in multiples of some basic unit. We assume that during the maximal riding time T_{max} a bus can travel 100 distance units and that during the maximal waiting time WT_{max} it can travel less than 70 distance units. Since the total number of students to be transported is 120, the sum of the capacities of available buses must be at least 120.

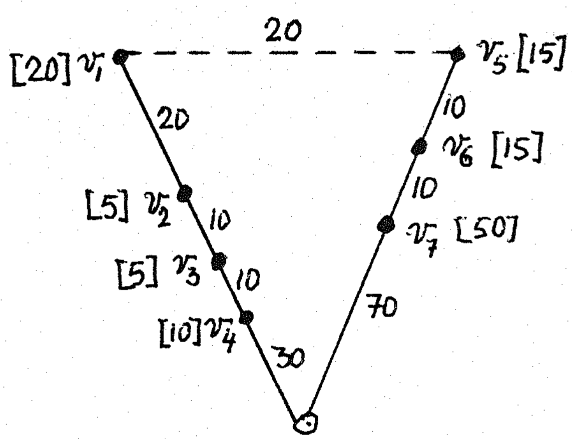


Fig. 1a

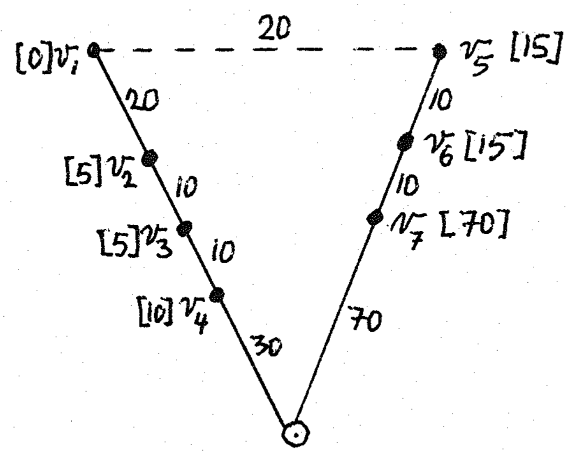


Fig. 1b

Example 1 (Fig. 1a).
Case 1: There are two buses available with capacities c_1 = 80 and c_2 = 40. The requirements can be satisfied using the following routes (numbers in the brackets indicate how many students were taken at a location):

Bus 1: v_5[15], v_6[15], v_7[50], School.

Bus 2: v_1[20], v_2[5], v_3[5], v_4[10], School.

Both buses can follow the direct routes from their starting points to the school.

Case 2: The available capacities are c_1 = 75, c_2 = 45. The following routes give a solution in this case:

Bus 1: v_5[10], v_6[15], v_7[50], School.

Bus 2: v_5[5], v_1[20], v_2[5], v_3[5], v_4[10], School.

In this case the first bus follows the shortest path from v_5 to the school while the second bus makes a "detour" at the beginning of its route. Notice also that the students at vertex v_5 had to be split in order to get a solution. We can split students at vertex v_6 and get another solution:

Bus 1: $v_5[15]$, $v_6[10]$, $v_7[50]$, School.

Bus 2: $v_6[5]$, $v_5[0]$, $v_1[20]$, $v_2[5]$, $v_3[5]$, $v_4[10]$, School.

The last solution has a disadvantage in that that the second bus must follow a longer route (still feasible). Following the longer route with a detour will be necessary in the next case.

Case 3: The available bus capacities are $c_1 = 64$, $c_2 = 56$. A solution (not unique) is:

Bus 1: $v_6[14]$, $v_7[50]$, School.

Bus 2: $v_6[1]$, $v_5[15]$, $v_1[20]$, $v_2[5]$, $v_3[5]$, $v_4[10]$, School

Example 2 (Fig. 1b). The difference between this network and the one depicted in Fig. 1a is only in the labeling of vertices v_1 and v_7, but the total number of students in the network is unchanged. We can see that with the bus capacities $c_1 = 80$ and $c_2 = 40$ (Case 1 of Exmple 1) as well as with capacities $c_1 = 75$ and $c_2 = 45$ (Case 2 of Example 1) the problem has a solution. But the problem has no solution with capacities $c_1 = 64$ and $c_2 = 56$ (Case 3 of Example 1).

We can notice two different types of routes in the previous examples:

a) **direct routes** along which the students are transported using the shortest paths to the school,

b) **routes with detours** along which at least one student must make a detour on his way to the school.

It will be useful to have a classification of the different types of detours which can arise in order to simplify our future discussions. For illustrative purposes we will consider a network with total of 40 students as depicted in Fig. 2.

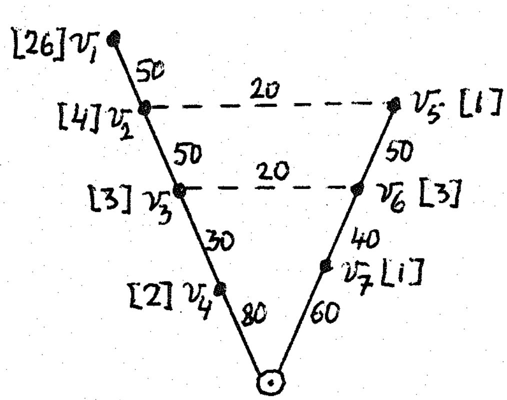


Fig. 2

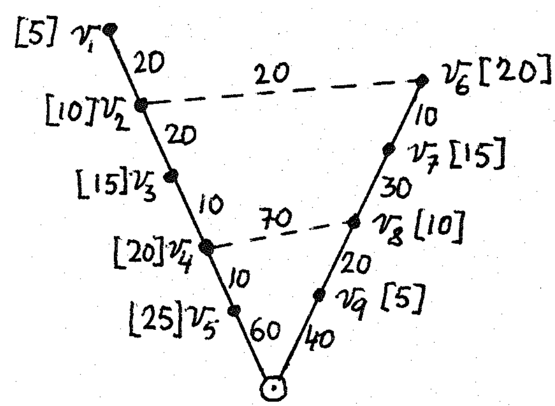


Fig. 3

Example 3 (Fig. 2). Assume that T_{max} corresponds to 210 distance units.

Case 1: $c_1 = 36$, $c_2 = 4$. The transportation can be accomplished using the following routes:

Bus 1: $v_1[26]$, $v_2[4]$, $v_3[2]$, $v_6[3]$, $v_7[1]$, School.

Bus 2: $v_5[1]$, $v_2[0]$, $v_3[1]$, $v_4[2]$, School.

Obviously, both buses must make detours in order to get a feasible solution.

Case 2: $c_1 = 31$, $c_2 = 9$. We have the following routes:

(3)
Bus 1: $v_1[26]$, $v_2[3]$, $v_3[0]$, $v_4[2]$, School.

Bus 2: $v_5[1]$, $v_2[1]$, $v_3[3]$, $v_6[3]$, $v_7[1]$, School.

We can see that the second bus must follow the route indicated in order for the problem to be solvable. The choice is only in the decision where it picks up students. Thus instead of (3) we can have:

(3')
Bus 1: $v_1[26]$, $v_2[2]$, $v_3[1]$, $v_4[2]$, School.

Bus 2: $v_5[1]$, $v_2[2]$, $v_3[2]$, $v_6[3]$, $v_7[1]$, School.

Before we describe the classification of detours we will consider one last example based on network from Fig. 3.

Example 4 (Fig. 3). The following bus capacities are available: $c_1 = 65$, $c_2 = 60$. T_{max} corresponds to 130 distance units. A solution is given by the routes:

Bus 1: $v_1[5]$, $v_2[0]$, $v_3[15]$, $v_4[20]$, $v_5[25]$, School.

Bus 2: $v_2[10]$, $v_6[20]$, $v_7[15]$, $v_8[10]$, $v_9[5]$, School.

We see that in this case the first bus does not take any students from v_2 in spite of the fact that it passes through the location. v_2 is served by the second bus which has to make a detour. If the first bus took all the students from v_2 then it would leave behind some students in locations v_3, v_4, or v_5. But it is easy to see that in that case the remaining available bus cannot transport all the remaining students. The same situation would occur if the first bus took only a single student from v_2.

This example illustrates that there can be situations in which a partially filled bus must pass through a location without taking a single student there. Otherwise it would lead to an increase of the required number of buses providing the transportation.

The few examples given here indicate the complexity of the problem at hand. They show first of all that even very small changes in the weights of the edges, the number of students at some locations, the bus capacities, and/or the structure of the network can lead to a change in the value of the objective function or the routes and schedules of the buses. Any efficient method which will solve the problem of minimizing the objective function subject to constraints of the described types must take all these facts into account. The following section outlines how these questions are handled in the computer system we have developed and implemented.

III. STRUCTURAL CHARACTERISTICS OF THE PROBLEM

We showed in the previous section that the buses must take different types of detours in order to get a good value for the number of buses used for the transportation. While a bus following the shortest path brings the students to the school in the shortest time, the bus following a detour carries some students around for longer time. For this reason it is natural to give the highest priority to direct routes. Whether the problem is solvable optimally with direct routes or routes with detours depends basically at the following:

a) the structure of graph G,
b) the weights of its edges and labels of its nodes **WL**,
c) the available bus capacities **C**.

The **triple** (G,**WL**,**C**) determines implicitly the value of the objective function and its corresponding routes and their types. Therefore an algorithm for a solution of the minimization problem, which is based on graph theoretical and combinatorial techniques, must fully utilize these facts. In order to get a good (heuristic) value of the objective function it is essential to use the information provided by the triple (G,**WL**,**C**) simultaneously as opposed to methods using the given information sequentially (i.e. the network is used first to generate the routes and only then are the buses considered and scheduled).

In order to satisfy the requirement of "simultaneity", which must be reflected in the developed computer system, we turn our attention first to questions of

- proper choice of different types of bus routes,
- proper choice of subgraphs G' of G,
- a covering of G' by buses of the set **C**.

We start with a classification of different route types.

a. Classification of Route Types

The simplest route type is the **direct** one. Here each student on the bus is transported to the school along the shortest path (see Fig. 4a). On the other hand, bus routes which start at a location where the first student enters the bus and end at the school which are not direct routes are called **routes with a detour** or simply **detours.** Since every bus uses at least for a part of its route a direct route to the school (if not sooner then after taking the last student it travels directly to the school) we can assign to each detour a part of a direct route DT. The longer the part DT of a route is the shorter the required detours on the route.

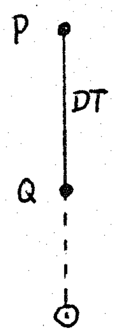


Fig. 4a

Direct route

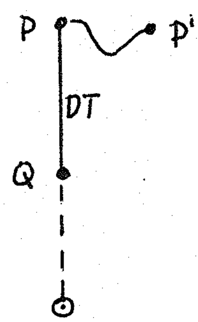


Fig. 4b

Type 1 detour

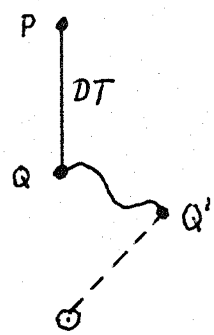


Fig. 4c

Type 2 detour

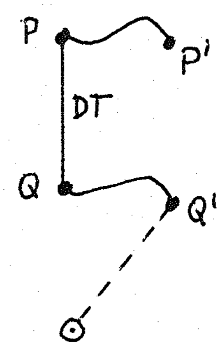


Fig. 4d

Type 3 detour

Type 1 detour (Fig. 4b) is characterized by the fact that the bus starts with a detour between points P' and P. Then it follows the shortest path to the school passing through point Q. Only those students entering the bus at points starting at P' and before P are inconvenienced by following a longer path to the school than necessary. We saw this route type in Example 1, Case 2 and 3, of the previous section.

Another type of detour, **type 2**, is characterized by the fact that the route starts at point P, follows a direct path to the school up to point Q and at Q it sidetracks to point Q'. It is continued along the shortest path from Q' to the school (Fig. 4c). This time all students entering the bus before point Q' make a longer trip than absolutely necessary, while the students entering between Q' and the school follow the shortest path. An example of this route type is given in Case 1, Example 3, Section II. Case 2 of the same example shows that the second bus makes a detour which is neither type 1 nor type 2. This detour type, called **type 3**, is composed of the previous two types in such a way that the part DT (following the direct route to the school) is preceded by a detour P'P and followed by a detour QQ' (Fig. 4d).

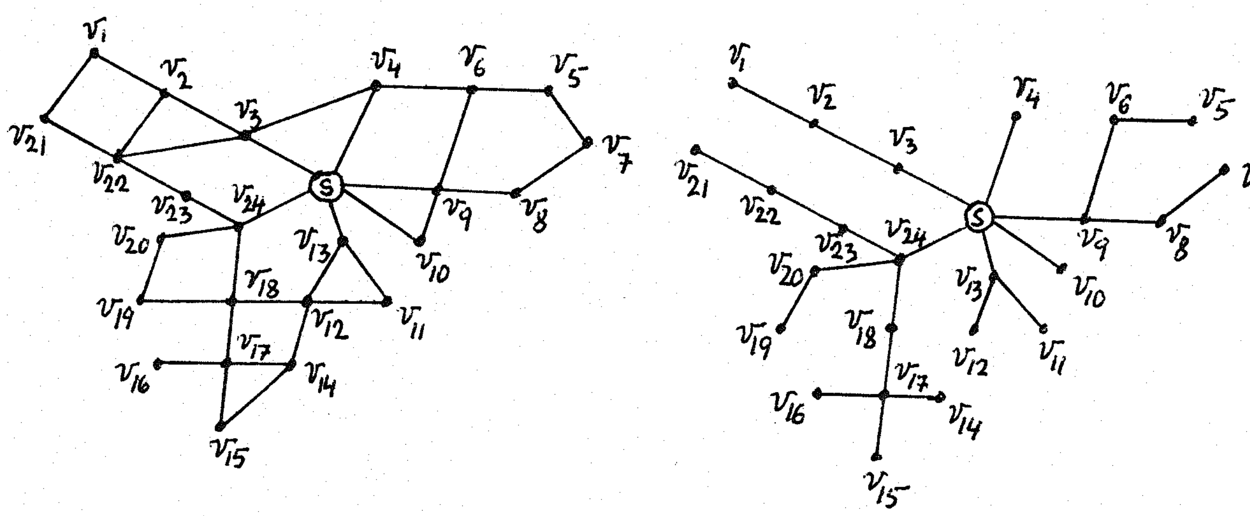
The route types illustrated by Fig. 4a - 4d are so called

open routes. From assumption P6 of Section I follows that we can consider as a bus route that part of the bus trip between the location in which the bus picks up the first student and the school. Only if a bus makes a second trip then it follows a **closed path** with the school as both the starting and ending point. As described in Section II, the second trip of a bus is possible only in the inner district of the school $\mathbf{V_0}$ and therefore all routes serving the outer district $\mathbf{V_1}$ of the school can be considered as open.

We will use the route types from Fig. 4a - 4d in the computer system developed. In order to obtain a good value of the objective function we will try to cover the outer district $\mathbf{V_1}$ (=V-$\mathbf{V_0}$) of graph G using a minimum number of routes of the above described types.

b. Shortest Path Tree

A common feature of all route types is the direct part DT. We need some information at hand which will allow us to recognize the possibility of a direct route. To that end we will generate a spanning tree **T** in network G of shortest paths from the school (root of the tree) to all other nodes. Note that such a tree is not necessarily unique. Every shortest path **A** in **T** from a leaf of the tree to the school represents a possible direct route from the leaf to the school (Fig. 5).



Network G representing locations and their connections

Shortest path tree **T** of G (with school as its root)

Fig. 5

We will use the shortest path tree to classify nodes of G different from the school. A node v V(G), $v \neq$ school, is called

- a **leaf** with respect to **T** if its degree in **T** is 1,
- an **inner point** if its degree in **T** is 2,
- a **branching point** if its degree in **T** is greater than 2.

In the case of tree **T** from Fig. 5 we have the following situation:

Leaves with respect to **T**: v_1, v_4, v_5, v_7, v_{10}, v_{11}, v_{12}, v_{14}, v_{15}, v_{16}, v_{19}, and v_{21}.

Branching points with respect to **T**: v_9, v_{13}, v_{17}, and v_{24}.

All other nodes are inner nodes with respect to **T**.

A branching point v_r from **T** implies a subtree **T**(v_r) of **T** in the following manner:

1. v_r is **root** of **T**(v_r).

2. $v_i \in$ **T**, $v_i \neq v_r$, is a node of **T**(v_r) exactly if

   (i) v_i is farther from the school than v_r,

   (ii) the shortest path (in **T**) from v_i to the school passes through v_r.

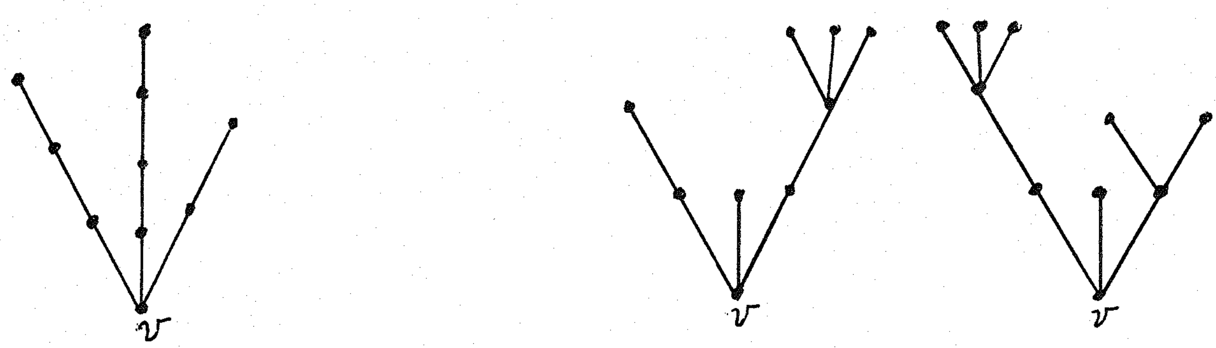
For example, the branching node v_9 in the above tree **T** implies a subtree **T**(v_9) defined by

$$\mathbf{T}(v_9) = \{v_i : i=5,6,7,8,9\}.$$

Similarly, v_{24} implies a subtree

$$\mathbf{T}(v_{24}) = \{v_i : i=14,\ldots,24\}.$$

In the following we will call every subtree T(v) implied by a branching point v in **T**, as defined above, the **fork with source v.**



Fork of the first degree

Forks of higher degrees

Fig. 6

If the fork $\mathbf{T}(v)$ does not contain any point which itself is a branching point with respect to $\mathbf{T}$ then $\mathbf{T}(v)$ is called a **fork of the first degree** (Fig. 6).

In the tree of Fig. 5 there are the following forks of the first degree:

$$\mathbf{T}(v_9) = \{v_9;\ v_5, v_6, v_7, v_8\},$$

$$\mathbf{T}(v_{13}) = \{v_{13};\ v_{11}, v_{12}\},$$

$$\mathbf{T}(v_{17}) = \{v_{17};\ v_{14}, v_{15}, v_{16}\}.$$

The fork defined by the branching point v_{24}:

$$\mathbf{T}(v_{24}) = \{v_{24};\ v_i\ :\ i=14,\ldots,23\}$$

is not of the first degree since it contains additional branching point v_{17}.

The forks play an important role in the development of methods for solving our transportation problem. For this reason it is useful to look more closely at the structure of forks of the first degree. Let v_i and v_j be two arbitrary nodes of a fork $\mathbf{T}(v)$ of the first degree that are different from the root v and do not belong to the same branch of $\mathbf{T}(v)$. If v_i and v_j are adjacent in G, i.e. there is an edge between them, than we call the two branches of $\mathbf{T}(v)$ containing v_i and v_j **neighbouring in** $\mathbf{T}(v)$. A fork is called **isolated** if all its branches are pairwise not neighbouring (Fig. 7).

Forks of the first degree:

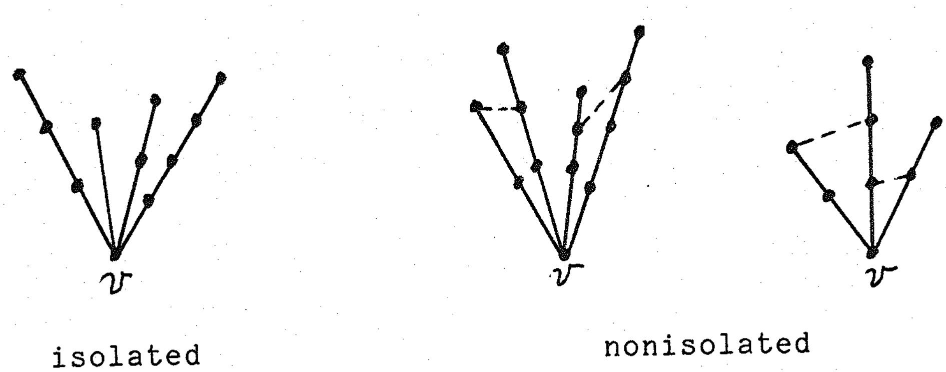
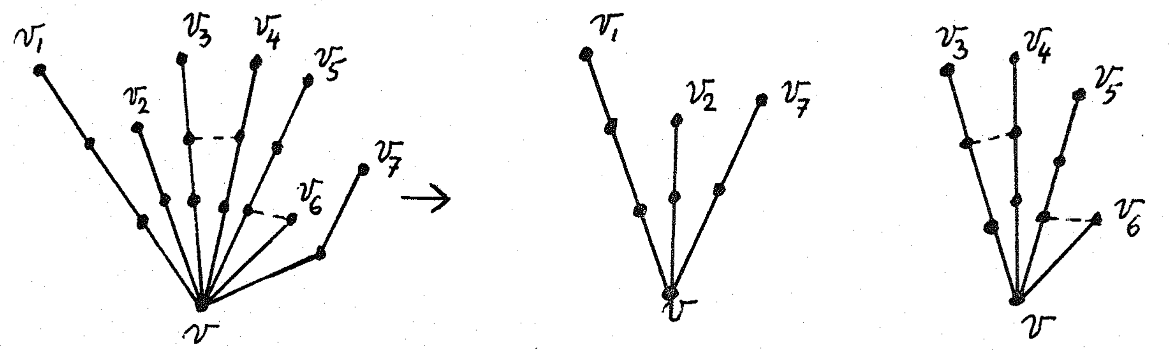


Fig. 7

Every fork of the first degree can be decomposed into its isolated and nonisolated components (Fig. 8).



Fork of first degree | Isolated component | Nonisolated component

Fig. 8

c. Decomposition of the Shortest Path Tree

An analysis similar to the one we applied to forks of the first degree can be applied to the whole shortest path tree **T** having the school as its root. First of all it can be established whether **T** can be decomposed into a set of subtrees $\mathbf{T}_1, \ldots, \mathbf{T}_r$ all of which have the school, call it v for a moment, as the root but otherwise are disjoint so that the following conditions hold:

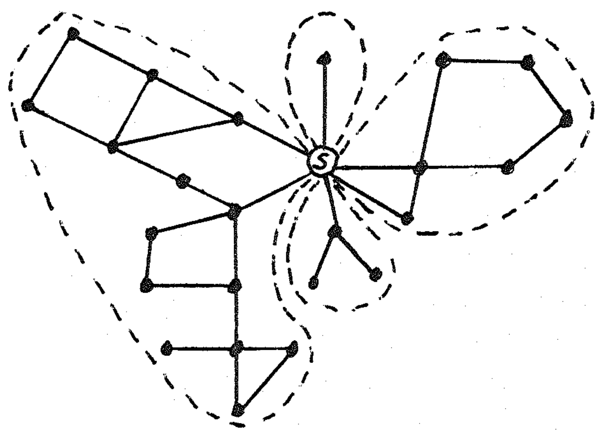
1. $V(\mathbf{T})-\{v\} = \bigcup_{i=1}^{r} (V(\mathbf{T}_i)-\{v\})$.

2. Every path in G between arbitrary nodes v_i in $V(\mathbf{T}_i)$ and v_j in $V(\mathbf{T}_j)$, $i \neq j$, either passes through the school or the time required for traveling from v_i through v_j (or from v_j through v_i) to the school is greater than the allowed upper limit ($i,j = 1, \ldots, r$).

3. Each subtree $\mathbf{T}_i$, $i = 1,\ldots,r$, contains at least one node from the outer district $\mathbf{V}_1$ of the school (see Section II).

4. The decomposition of **T** is maximal, i.e. there is no $\mathbf{T}_i$, $(1 \leq i \leq r)$ which can be itself decomposed.

As an example we can consider the network given in Fig. 9. It is a slight modification of the network from Fig. 5 and it has the same shortest path tree. While the network from Fig. 5 has only a trivial decomposition (i.e. it is itself the only component), the network from Fig. 9 has a decomposition into four components as indicated.



Maximal decomposition of a network into isolated components

Fig. 9

d. Composition of the Bus Fleet

So far we considered primarily structural properties of the network. Now we will look at some configurations of the bus fleet which will be important in the development of our algorithm.

Let $\mathbf{C} = \{c_i : i=1,\dots,n;\ c_i \geq c_j \text{ if } i<j\}$ be the ordered set of all available bus capacities. Using relation (2) of Section II we can determine the minimal number k of buses needed to transport S students. But this does not say anything about the number of different possible configurations of k buses selected from the available bus fleet which can provide the service. We can determine the overall **smallest** bus capacity which may be used (if only k buses are used for the transport) in the following manner: We substitute in (2) the term c_k by an element $c_j \in \mathbf{C}$, $j>k$, so that

$$(3) \quad S \leq c_j + \sum_{i=1}^{k-1} c_i, \text{ and if } j<n \text{ then } S > c_{j+1} + \sum_{i=1}^{k-1} c_i$$

Given a sufficiently large set **C** of capacities available for transportation of S students then there exists always

- a smallest number k of buses needed and
- a smallest capacity c_j from **C** which may be used so that the k buses are sufficient for the service.*)

*) Thus if we find out during the route generation process that a bus with the smallest capacity c_j cannot be completely filled and that a smaller bus capacity $c < c_j$ is sufficient then it implies that the total number of buses used for the transportation becomes greater than k.

Note that this is true in general only if other constraints of the problem (such as maximal riding or waiting time) are not binding. But let us ignore for a moment the other constraints and consider only the capacities.

Let

$$\mathbf{B} = \{c_{r1},\ldots,c_{rk} : c_{ri} \in \mathbf{C},\ i=1,\ldots,k;\ c_{ri} \geq c_{rj} \text{ if } i<j\}$$

be a k-element subset of capacities of **C** ordered in nonincreasing order. Then **B** is called a **representation of** S **of length k** if the following conditions are satisfied:

1. The k buses from **B** can transport at least S students, i.e.
$$S \leq \sum_{i=1}^{k} c_{ri}$$
2. Transportation of S students using only k-1 buses from **B** is not possible, i.e.
$$S > \sum_{\substack{i=1 \\ i \neq j}}^{k} c_{ri} \quad \text{for every } j,\ 1 \leq j \leq k.$$
3. No element $c_{ri} \in \mathbf{B}$ can be replaced by a smaller element of **C** without destroying property 1, i.e.
$$S > \sum_{i=1}^{k} c_{ri} - c_{rj} + c \quad \text{for each } c \in \mathbf{C-B} \text{ and } c<c_{rj}.$$

The difference **D** between the total capacity of representation **B** and the number of students S is called **defect** of the representation:

$$\mathbf{D} = \sum_{i=1}^{k} c_{ri} - S.$$

In general, there exists more than one representation of a given length, as the following example illustrates. Let

$$\mathbf{C} = \{36,34,28,26,22,18,17,12,10\}$$

be the available bus fleet and S = 52. The following are all representations of specified lengths:

<u>Length=2:</u>

(36,17), **D** = 1
(34,18), **D** = 0
(28,26), **D** = 2

<u>Length=3:</u>

(34,12,10), **D** = 4
(26,17,10), **D** = 1
(22,18,12), **D** = 0

<u>Length=4:</u>

(18,17,12,10), **D** = 5

Note that in this case the smallest capacity is 17 and k = 2.

The defect of a representation **B** is always nonnegative, otherwise **B** would not be a representation. On the other hand, if we replace some capacity in a representation by a larger one, then the first condition of the definition of a representation remains still valid, i.e. we can transport all S students using the k buses.

IV. SOLUTION STRATEGIES

In this section we describe basic ideas of solution strategy used in the development of our algorithm. We introduce them under a few suggestive headings.

a. Determination of the Minimum Number of Buses Required

Since we do not know the exact number of buses required in advance we assume that the input includes a description of a bus fleet having more buses of different capacities than necessarily required. This situation occurs also when we are looking for an improvement of a currently used transportation system by trying to reduce the number of buses actually used.

Generally, we can say that a greater number of available buses implies a greater number of bus representations (see Section III.d) and therefore a greater number of different choices for a solution. Since we are looking for an optimal solution, among possibly many others, we must be careful in our choices right from the very beginning.

The first piece of information which is readily available to us is the estimate following from (2), Section II. After the decomposition of the network into the **inner** and **outer districts** with respect to the school we apply (2) to the total number of students in both districts and the larger value obtained gives us a lower bound on the necessary number **MIN** of buses for the transportation.

b. Decomposition of the Network

A possible correction of number MIN found above can be required as a result of the decomposition of our network into isolated districts T_i, $i=1,...,r$, as described in Section III.c. This decomposition implies a disjoint decomposition of the outer district $\mathbf{V_1}$ into subdistricts $\mathbf{T'_i}$, where

$$\mathbf{T'_i} = \mathbf{T_i} \cap \mathbf{V_1}, \quad i = 1, \ldots, r.$$

From assumptions P2 and P3 of Section I follows that a transition from $\mathbf{T'_i}$ to $\mathbf{T'_j}$ is not possible*) and therefore the buses serving $\mathbf{T'_i}$ cannot in any way serve also $\mathbf{T'_j}$. Thus the districts $\mathbf{T'_i}$ and $\mathbf{T'_j}$ will be handled independently.

We will develop subroutines which will:

i) show whether the decomposition of $\mathbf{V_1}$ implies increase of the number of buses above MIN;

ii) generate representations for the districts $\mathbf{T'_i}$ so that the total number of all buses used to serve the outer district will be minimal.

The following example illustrates the point.

Example. Let $C = \{50,50,50,10,10,10\}$ be the available bus fleet and $S(\mathbf{V_1}) = 150$. This implies MIN = 3, since the three buses with capacity 50 are capable of transporting 150 students. If the district $\mathbf{V_1}$ is decomposed into

$$\mathbf{V_1} = \mathbf{T'_1} \cup \mathbf{T'_2} \cup \mathbf{T'_3} \tag{4}$$

with the corresponding numbers of students

$$S(\mathbf{T'_1}) = 70, \quad S(\mathbf{T'_2}) = 50, \quad S(\mathbf{T'_3}) = 30$$

then the required number of buses is increased to 5. The individual districts can be served by following bus fleets:

$\mathbf{T'_1}$: (50,10,10)
$\mathbf{T'_2}$: (50)
$\mathbf{T'_3}$: (50)

It is also obvious that every subdistrict $\mathbf{T'_i}$, $i=1,2,3$, has a representation of the minimal length l_i. Unfortunately, it is not known a priori whether this representation will be sufficient for serving $\mathbf{T'_i}$. In order to find an answer we will proceed now to a description of a heuristic strategy which will be used in our algorithm.

c. Additional Strategy Considerations

One can see the complexity of the problem from the previous

*) Note that a bus going from $\mathbf{T'_i}$ to $\mathbf{T'_j}$ must pass through the school and while there it can leave all the students collected before there. In view of the maximal waiting time allowed, WT_{max}, the bus is not able to reach district $\mathbf{T'_j}$ in the time period available before beginning of classes.

description and the illustrative examples given. For that reason we will not try to develop an exact algorithm which will give the true solution. Our goal will be rather to develop practical heuristics based on the structural properties of the problem which will be used in our calculations. We will also avoid use of backtracking in cases when a good bus route determined in a local district is later found to be a bad one. Backtracking methods are justified only when we are completely sure about the good quality of the solution found. But this is the case only for exact methods.

We will proceed in such a manner that a bus route determined as a good one locally will be accepted as satisfactory and will become a part of our final solution. The bus assigned to serve this route is no longer available for service elsewhere. That means that if we determine a bus route and a bus serving it then

i) the available bus fleet size is reduced by the bus serving the route,

ii) network G will be reduced by removing nodes containing no more students waiting for transportation,

iii) the labeling of nodes which were served only partly by the bus route will be modified.

We explained in Section III that the simultaneous consideration of the structure of network G, the node and edge labeling **WL**, and the available bus fleet **C** plays a key role in determining the quality of the results obtained. The more complicated the structure of G is, the more complicated and diverse are the questions of bus route determination. In order to keep under control the total number of different possibilities considered, it is useful to consider not the whole district $\mathbf{T'}_i$, $i=1,\dots,r$, at the same time, but only some of its well defined subdistricts and use an iterative algorithm for their analysis. These subdistricts should be sufficiently well structured within network G to allow identification of different alternatives.

We can use the **forks of the first degree**, introduced in Section III.b, as subdistricts satisfying these conditions. They are easily identifiable using the shortest path spanning tree **T** (Section III.b) and they represent simple but important structural properties of network G. We proceed now to a description of our strategy for algorithmical handling of forks of the first degree.

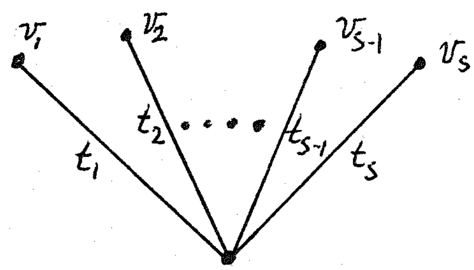
d. Algorithms for Handling Forks of the First Degree

Let **F** (=**T**(v)) be an arbitrary fork of the first degree of **T** with root v. In our algorithm we associate with **F** some other nodes of G, first of all the nodes lying on the shortest path from v to the school. Unless there is a specific reason other nodes of G will not be considered when handling **F** (including those of the neighbouring forks). For example in the network of Fig. 5 nodes v_{14}, v_{15}, v_{16}, and v_{17} form a fork of the first degree with root v_{17}. In considering this fork we will consider also nodes v_{18} and v_{24} but not node v_{12} which is a neighbour of v_{14} in G but which belongs to the fork $F_1 = \{v_{13}; v_{11}, v_{12}\}$.

It follows from the discussion in Section III.b that **F** is decomposed into **isolated** and **nonisolated** components (see Fig. 8). If a bus serves a nonisolated component then it can pass through the root of the fork **only once.** On the contrary a bus serving an isolated component must repeatedly return to the root if it serves more than one branch of the fork. Therefore routing along an isolated component can be very ineffective since some branches must be traversed twice in opposite directions and some nodes visited twice. For that reason, we will determine first if a branch of an isolated component is connected to some branches of other forks (of the first or higher degrees). If that is the case then such a branch is not considered any longer isolated and we look for possible detours when routing it.

Assume that an isolated component of a fork contains s branches, none of them being a neighbour of branches of other forks. Denote by t_i the travel time needed to traverse branch i of the component. We can assume without loss of generality that

$$t_1 \leq t_2 \leq \ldots \leq t_s.$$



The minimal time needed for a bus to traverse all of the branches of the component and arrive finally at v (for the last time) is

$$T_1 = t_s + 2 \sum_{j=1}^{s-1} t_j.$$

If t is the travel time needed between v and the school then the bus can serve the whole isolated component only if

$$T_1 \leq T_{max} - t.$$

On the contrary, if

$$T_1 > T_{max} - t$$

then there are **at least** two buses needed for serving the isolated component. For exactly two buses we obtain the following time bound T_2 for traversing the whole component

$$T_2 = t_s + t_{s-1} + \sum_{j=1}^{s-2} t_j = T_1 - t_{s-1}.$$

If $T_2 \leq 2(T_{max}-t)$, then it may be possible to serve the whole isolated component with two buses. Otherwise at least one more bus is needed and we get the following value T_3 (a lower bound for the total time):

$$T_3 = t_s + t_{s-1} + t_{s-2} + 2 \sum_{j=1}^{s-3} t_j = T_1 - t_{s-1} - t_{s-2}.$$

If $T_3 > 3(T_{max}-t)$ then at least one more bus is needed. In general, there will be at least k (<s) buses needed if

$$T_k = \sum_{i=0}^{k-1} t_{s-i} + 2 \sum_{j=1}^{s-k} t_j \leq k(T_{max}-t),$$

(5)

$$T_e > e(T_{max}-t) \text{ for all } e,\ 1 \leq e < k.$$

We can see that in considering the isolated component of a fork of the first degree, relation (5) gives us a lower bound k for the minimal number of buses based only on the structural properties of network G and weights of its edges. Whether this number is also sufficient depends not only on the network itself but also on the way how bus routes are determined, how the total number of students is distributed among the nodes of the network, and on the available bus capacities.

Consider now one of the buses. If it starts its route with traversing the longest branch (once only) then for its total travel time the following must hold:

$$\sum_{j=1}^{s-1} 2t_j.x_j \leq T_{max} - t_s - t,$$

(6)

$$\text{where} \quad x_j = \begin{cases} 1, & \text{if bus traverses branch } j, \\ 0, & \text{otherwise.} \end{cases}$$

Since we try to optimize, the difference between the right and left hand sides of (6) should be minimal. This will occur if the left hand side of (6) is maximum:

(7) $$\max \sum_{j=1}^{s-1} 2t_j.x_j.$$

The problem of maximizing (7) subject to (6) is a **knapsack problem.** Thus we can find routes for buses within an isolated component of a fork by solving knapsack problems.

The number k defined above is only a lower bound of the number of buses needed. Whether k buses are sufficient depends also on the distribution of the students among the nodes of the network. For example, if we consider the fork in Fig. 10 and if the allowed driving time along the branches of the fork is T = 200 units then looking **only at the time restriction** the fork can be served by two buses using the following routes: v_6, v, v_1, v, v_4, v and v_5, v, v_2, v, v_3, v. If the available bus fleet contains capacities $c_1 = 40$, $c_2 = 29$, and $c_3 = 25$, then the fork

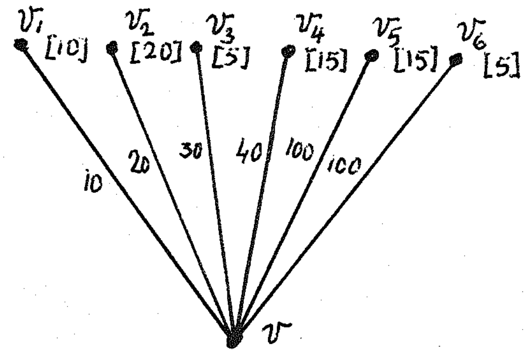


Fig. 10

cannot be served by two buses. It can be served by two buses if capacity c_2 is replaced by a larger one.

The algorithmic treatement of an isolated component will consist from the following steps applied iteratively:

1. Determine the total number of students in the isolated component and the length of a minimal bus representation.
2. Set up the knapsack problem and solve it.
3. Determine the number of students along the route identified in 2.
4. Find an appropriate bus capacity to serve the whole route or a part of it.
5. If an unused capacity remains try to fill the bus on its way to school or find a detour at the beginning of the route (through nodes of another fork), if possible.
6. Reduce the network by removing nodes completely served.
7. Reduce the bus fleet by removing the bus used.
8. If the isolated component has not been served fully, return to step 2, otherwise we are finished with the routing of the component.

Information found in step 1 can be used to evaluating how well we succeeded in routing the component. If the number of buses used for serving the component is larger than the length of a minimal bus representation, then it means that it is not possible to reach the absolutely minimal number of buses MIN for serving the whole network. The number MIN will also be exceeded if the appropriate bus capacity found in step 4 is smaller than the previously determined minimal bus capacity (see Section III.d).

The nonisolated components of forks of the first degree will be handled in a different manner. Fig. 11 shows that these components can be classified into those allowing a transition from any branch to any other branch without passing through the root v of the fork (Fig. 11a) and those where it is not possible

(Fig. 11b). In the latter case there exists a splitting of the whole component into nonisolated subcomponents which are **pairwise isolated**, i.e. we can move from a node in one subcomponent to a node in another subcomponent only through the root v of the fork of the first degree while visiting only nodes of the fork.

Nonisolated components

a. Without splitting   b. With splitting

Fig. 11

Since the nonisolated components of a fork of the first degree offer more possibilities for detours than the isolated ones, it is desirable to have some information available which can be used to initialize the routing process. This information should contain first of all some knowledge about topological properties of the examined district of the given network. It can be important whether the routing process starts with an **internal** or a **boundary** node of the district. In the example of Fig. 12, the nodes of the fork can be visited by a bus in time T = 70 units if it starts its trip in one of the "boundary" nodes v_1 or v_3 (i.e. nodes lying at the "outer" branches of the fork). If the bus started its route at the "inner" node v_2 then it would not be possible to serve all the nodes within the available time (assuming this time is less than 90 time units).

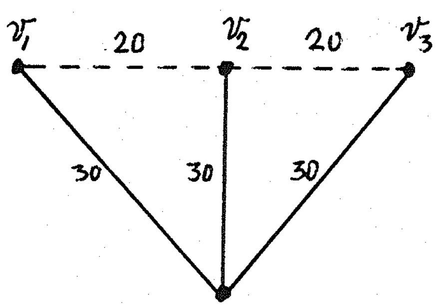


Fig. 12

We introduce now a concept of a distance measure for branches of a fork of the first degree based on the distance of nodes which will allow us to make subsequent description more precise.

Definition. Let B_i and B_j be two branches of a fork of the first degree with root v. The minimum of all distances between an arbitrary node of B_i and arbitrary node B_j (both of them distinct from v) is called **distance** of B_i and B_j and is denoted by $d(B_i,B_j)$.

Our handling of nonislated component **K** of a fork of the first degree (no matter whether it is split or not - see Fig. 11) is based on the following steps:

1. Determine the **total number of students** S in **K**, not counting the students at root v, and the number S' of all students from **K**, including those in v.
2. Determine a minimal representation of S and bus configurations with **minimal defect** (Section III.d).
3. Determine distances of branches of **K**.
4. Determine appropriate routes for the buses of the representation.

Step 2 uses information obtained in step 1 and generates a minimal subset **B** of the available bus fleet which can be used for transportation of S students. **B** has the property that its defect, i.e. the number of extra seats (above the required number S) is minimal. These bus capacities are used in step 4 for determination of routes in such a way that the time restrictions are satisfied and all S students are served, if possible. The following describes how this is actually done.

The route generated should start with one of the outer branches of the fork, if possible, as we illustrated in the example of Fig. 12. Of course, it can happen that an outer branch does not exist, as shown in Fig. 13. Nevertheless, if outer branches are available then they are characterized as the pair B_i and B_j of branches of the nonisolated component with maximum distance $d(B_i,B_j)$. If there are no outer branches then we can select the pair of branches having maximum distance. In any case we select out of the two branches determined the longest one and denote it by B. Let S_B be the total number of students along B (not including the root v).

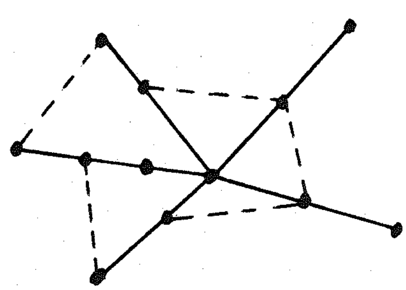


Fig.13

Since $S_B < S$ we can determine the minimal number k such that k buses chosen from the set **B** found in step 2 are sufficient (with respect to their capacities) for transportation of S_B students. We can also find a representation $\mathbf{B}_B$ of S_B of length k (from **B**) so that its defect $\mathbf{D}_B$ is minimal, i.e. the set $\mathbf{B}_B \subset \mathbf{B}$ of k buses is sufficient for the transportation of S_B students and $\mathbf{D}_B \geq 0$ seats will not be used. If $\mathbf{D}_B = 0$ then the transportation can follow the **direct route** along branch B.

If $\mathbf{D}_B > 0$ then the situation looks different. In this case not all buses from $\mathbf{B}_B$ can be completely filled along branch B. Without loss of generality we can assume that only one bus $C \in \mathbf{B}_B$ is not completely filled along branch B while the remaining buses from $\mathbf{B}_B$ are fully loaded. The main emphasis incorporated into our approach to route generation is:

a) use detours through other nodes of the currently analyzed fork (different from its root v) to pick up additional $\mathbf{D}_B$ students; if this is not possible then
b) fill the remaining empty seats of C by students along the direct way from root v to the school.

Thus our strategy consists in trying to eliminate the calculated defect $\mathbf{D}_B$. If $\mathbf{D}_B$ is eliminated using only nodes of the fork (case a) above) then this is the best possible result, at least temporarily. Of course, it may prove later to be a poor result.*) Since we do not use backtracking, as mentioned earlier, we accept such a solution as the best possible and include the route found into the final solution. If we must use case b) then a **conflict situation** can arise. It can be recognized in that the necessary minimal number of buses for routing the current fork must be increased. The situation occurs if $\mathbf{D}_B > \mathbf{D}$ and if it is not possible to pick up at least $\mathbf{D}' = \mathbf{D}_B - \mathbf{D}$ students along a detour through nodes of the fork outside of branch B and different from root v. Whether there are some possibilities to get rid of this conflict situation, and what they are, will be discussed in the following when we describe our routing methods.

The following example will illustrate this situation. Let the given values be

$$S = 115, \quad S_B = 55, \quad \mathbf{C} = (50,40,30,10).$$

Then the representation B of S with minimal length is

$$\mathbf{B} = (50,40,30); \quad \mathbf{D} = 5.$$

The representation $\mathbf{B}_B$ ($\subset \mathbf{B}$) of S_B with minimal length is

$$\mathbf{B}_B = (40,30); \quad \mathbf{D}_B = 15.$$

We have $\mathbf{D}_B > \mathbf{D}$. It is easy to see that unless there can be at least $\mathbf{D}' = 15 - 5 = 10$ students picked up at nodes (different from root v and outside branch B) inside the fork considered then there must be four buses used for routing the fork, not only three.

---

*) This can happen, for example, if bus capacities used might be better used later in the process but are no longer available.

e. Route Finding Methods

Let us assume that the defects **D** and D_B of the representations **B** and B_B, respectively, as described in the previous paragraph, satisfy the relation $0 < D_B \leq D$. We will describe now the methods used to find the appropriate routes.

Since $D_B > 0$ therefore at least one bus of the representation B_B should make a detour, if possible, in order to decrease D_B. Without loss of generality we may assume that it is the smallest bus from B_B. This way only a minimal number of students will be required to make a detour.

We will use the three types of detours described in Section III.a (Fig. 4a-4d) as possible detours. First we will try to reduce completely the defect D_B using a detour of type 1. If it is not possible we will look for a detour of type 2. If we fail even then we will look for a detour of type 3.

Before we proceed further let us remind ourselves of a situation which can complicate the process and which was demonstrated in example 4 of Section II (Fig. 3). A difficulty can arise from the fact that there exist nodes on a branch from which there is no detour possible because of the time restrictions. In example 4, v_1, v_3, and v_5 are such nodes. In general, we will call nodes of a branch **critical** if they cannot be points of detours of type 2 or 3.

The number of possible detours from branches containing critical points is reduced in comparison to branches without critical points. Therefore handling branches with critical points requires special consideration. For this reason we start first with handling branches without critical points.

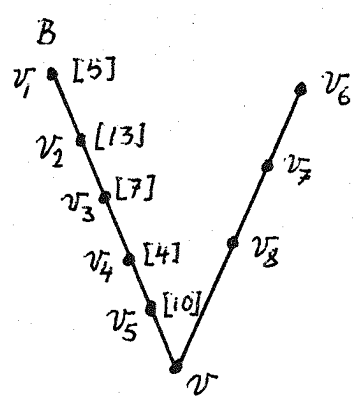


Fig. 14

If C is a bus with the smallest capacity c from representation B_B then it follows from our assumptions that all other buses from B_B different from C can be filled completely along a direct route (see Fig. 4a). For the bus with capacity c remains only $k = c - D_B$ students for transportation along branch B. The additional D_B students should be found, if possible, along a detour through other branches ($\neq B$) of the fork considered. In doing so it is important to watch the nodes of B from which the k students are picked up in order to get a solution. That leads us to definition of a k-block.

Definition. Let B be a branch of a fork with root v and let q be an arbitrary node from B. A set M of consecutive nodes from B is called a **k-block defined by q** if it has the following properties:

1. q belongs to M.
2. The distance of every node of M from the root v is not less than the distance of q and v.
3. The total number of students in nodes of M is at least k.
4. The set $M - \{q\}$ contains less then k students.

As an example consider branch B of the fork in Fig. 14.

The following table contains k-blocks for different values of k and different nodes on branch B.

q \ k	5	8	11	15	22	30
v_5	$\{v_5\}$	$\{v_5\}$	$\{v_4, v_5\}$	$\{v_3, v_4, v_5\}$	∅	$\{v_2, v_3, v_4, v_5\}$
v_4	∅	$\{v_3, v_4\}$	$\{v_3, v_4\}$	∅	$\{v_2, v_3, v_4\}$	∅
v_3	$\{v_3\}$	∅	∅	$\{v_2, v_3\}$	$\{v_1, v_2, v_3\}$	∅
v_2	$\{v_2\}$	$\{v_2\}$	$\{v_2\}$	$\{v_1, v_2\}$	∅	∅
v_1	$\{v_1\}$	∅	∅	∅	∅	∅

A k-block represents a sequence of consecutive nodes of a branch and it contains at least a given number of students. Node q defining the block determines the location of the k-block along

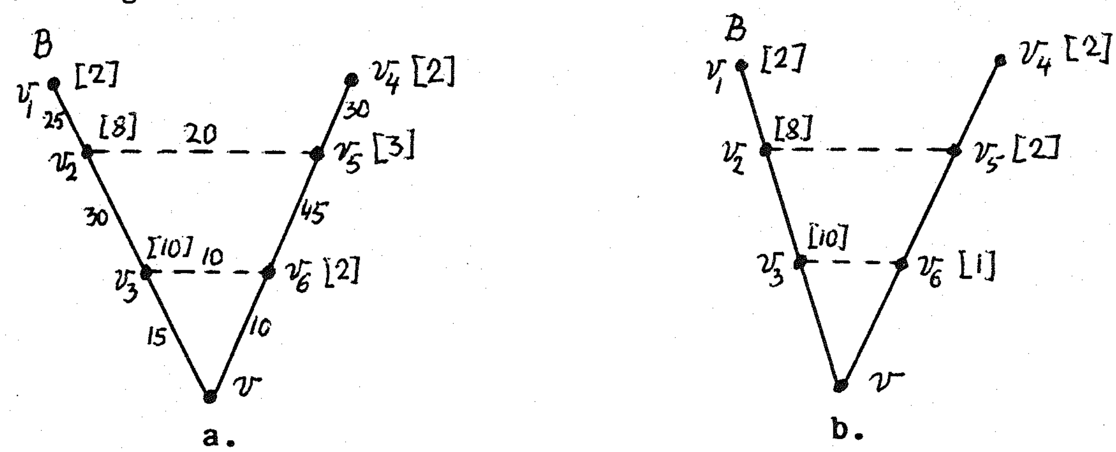


Fig. 15

the branch. This location is often the determining factor in an optimal routing of a detour, as illustrated by the following example.

Example. Let k, D_B, and T be given as follows: k = 10, D_B = 5, T = 100 (time available for a detour).

The two forks in Fig. 15 differ only in the number of students at nodes v_5 and v_6. Branch B contains k-blocks $\{v_1, v_2\}$ and $\{v_3\}$. If we consider network of Fig. 15a then there is only one possible detour for a bus with capacity c = 15 in order to pick up 5 more students from nodes outside branch B, namely v_1, v_2, v_5, v_6. For network of Fig. 15b the only possible detour for the same bus capacity is v_4, v_5, v_6, v_3.

As soon as we find out that a detour is needed in order to use best a bus capacity (when routing along a branch B of a fork) we determine a specific k-block of branch B, namely the k-block KBl which is defined by the node of B closest to the school (or to root v of the fork).

If p is the node of KBl most distant from the school and if the driving time from p to the school is t then time $T = T_{max} - t$ is available for a detour. The possible candidate nodes for a starting point of a detour are those nodes of the fork, not lying on branch B, which are reachable from p within the available time T. Denote the set of these nodes by M1. Using a tree search algorithm we will try to determine a subset $\mathbf{M} = \{v_i \in M1: i \in I\}$ of M1 having the following properties:

1. If we add to **M** node p then the nodes of **M** can be ordered into an open path W with one end point at p, which can be traversed in time less than or equal to T.
2. The total number of students at the nodes of **M** - {p} is at least D_B.
3. The length of W cannot be shortened by eliminating a point (different from p) from W without destroying property 2.

If these properties are satisfied then our problem is solved, i.e. using a detour we can fill completely the bus considered.

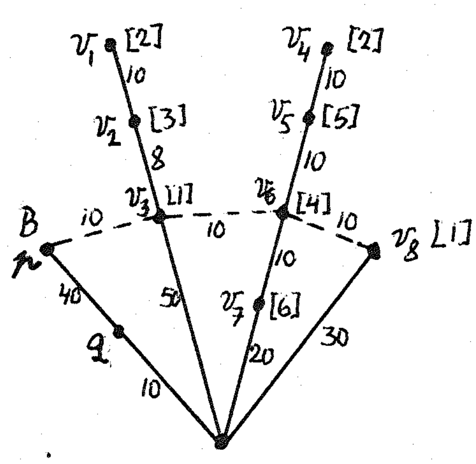


Fig. 16

Example. Consider the fork of Fig. 16.

If $T = 40$ then $M1 = \{v_i: i=1,\ldots,8\}$ and the possible paths are:

$$W_1 = (v_1, v_2, v_3, p)$$

$$W_2 = (v_4, v_5, v_6, v_3, p)$$

$$W_3 = (v_7, v_6, v_3, p)$$

$$W_4 = (v_8, v_6, v_3, p)$$

$$W_5 = (v_2, v_6, v_3, p)$$

If $\mathbf{D}_B = 7$ then W_1 and W_4 do not satisfy property 2. From the remaining paths only W_3 and W_5 satisfy property 3. Property 3 is not satisfied for W_2, but path W'_2 W_2, where $W'_2 = (v_5, v_6, v_3, p)$ satisfies all three conditions. Which one of the three paths W'_2, W_3, W_5 will be found as the first one and thus become a part of a solution depends on the algorithm used.

As a further point we can notice that a detour found can be improved upon with respect to its length. The example above has path $W_5 = (v_2, v_6, v_3, p)$ of length 38 while the path $W'_5 = (v_6, v_2, v_3, p)$ has length only 36.

We already mentioned earlier (Section IV.d) that a conflict situation can arise as a consequence of a decision made previously, without really knowing what possible consequences it could have later. When we look for detours we also select exactly one of perhaps several possibilities in the hope that we made a good choice. If as a consequence of our choice we run into trouble later we will not return and examine whether an alternative choice could have avoided the trouble.

If our attempt to find a detour does not lead to a desired result then there is no detour of the described type along which at least $\mathbf{D}_B$ students can be picked up. Since the detour finding process is deterministic, we obtain some useful information even if we do not succeed in determining a detour. This information can be used later in our decisions. First of all we find a detour along which a maximal number of students $\mathbf{D}_m$, where $\mathbf{D}_m < \mathbf{D}_B$, can be picked up together with the required minimal time for its traversal. All this is available for subsequent steps. Whether this information will actually be used depends on the outcome of our search for another type of a detour which follows the traversal of a k-block as opposed to the above discussed type which preceeded a k-block. More specifically, if q is the node of a k-block closest to the school, then the bus should divert from branch B at q to some other nodes of the fork (not belonging to B) in order to collect additional $\mathbf{D}_B$ students (see Fig. 4c). If t_B is the time needed to traverse the k-block and t' the time required to travel from v to the school then $T = T_{max} - t_B - t'$ is the time available for a detour from node q to node v (root of the fork). If we denote by $t(v_i, v_j)$ the minimal required driving time between nodes v_i and v_j then every node l of the fork located inside the ellipse defined by the inequality

$$t(q,l) + t(l,v) \leq T$$

is a potential detour location. From l we can reach all nodes l_i of the fork satisfying the following inequality

$$t(l,l_i) + t(l_i,v) \leq T' \text{ , where } T' = T - t(q,l).$$

We will use again a tree search algorithm for determining a detour along which at least $\mathbf{D}_B$ students can be picked up between q and v. If we do not succeed in finding such a detour then we identify a detour along which a maximal number of students $\mathbf{D'}_m$, where $\mathbf{D'}_m < \mathbf{D}_B$, can be picked up and which can be traversed in minimal time.

Example. Consider the network from Fig. 17 with T = 100.

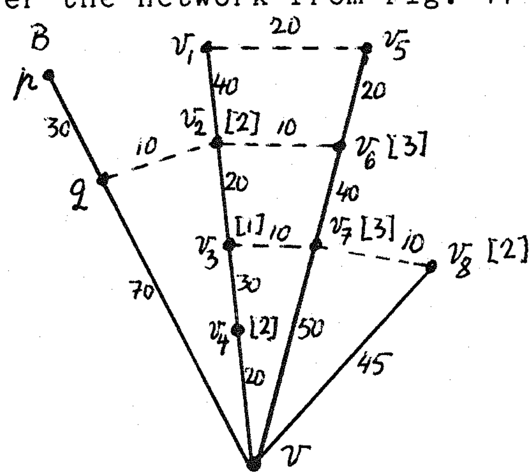


Fig. 17

Only nodes v_2, v_3, v_4, v_7, and v_8 can be considered for a detour starting from node q. The possible detours are

$$W_1 = (q,v_2,v_3,v_4,v)$$

$$W_2 = (q,v_2,v_3,v_7,v)$$

$$W_3 = (q,v_2,v_3,v_7,v_8,v)$$

with corresponding numbers of students $S_1 = 5$, $S_2 = 6$, and $S_3 = 8$, respectively. If $\mathbf{D}_B = 9$ then we will store for possible later use the result which comes closest to $\mathbf{D}_B$, i.e. W_3 with $\mathbf{D'}_m = 8$ and time to traverse the detour T' = 95.

If we cannot find any detour of type 1 or 2 through nodes of the fork lying outside of branch B along which at least $\mathbf{D}_B$ students can be picked up then we start looking for detours of type 3 which are combinations of detours of the previous two types (see Fig. 4d). In this case, a detour starts outside branch

B, reaches node p (a part of a detour of type 1), serves nodes of the k-block on branch B between nodes p and q, then leaves branch B at node q to follow a detour (a part of a detour of type 2) to root v of the fork and from there directly to the school.

Example. Consider the network from Fig. 18 with available bus capacities $c_1 = 30$, $c_2 = 10$.

If the time available for routing through the fork is T = 125 then the problem is solvable only when the route (v_2, p, q, v_3, v) is used. It is just a detour of type 3.

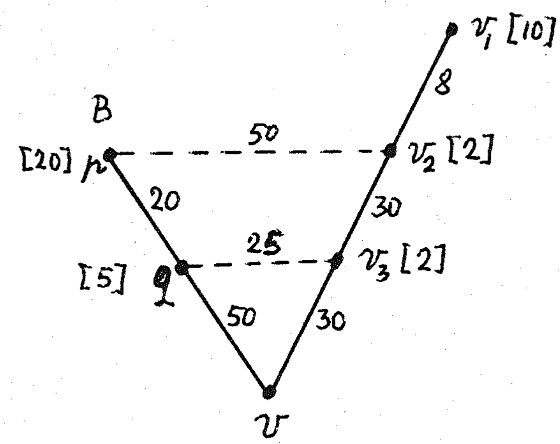


Fig. 18

If our attempt to find a detour of type 3 fails, i.e. we cannot find D_B students on a detour, it does not mean that we stop searching for other detours. In fact, by shifting the k-block on branch B away from the fork root v we can succeed in finding an appropriate detour. The following example illustrates this point.

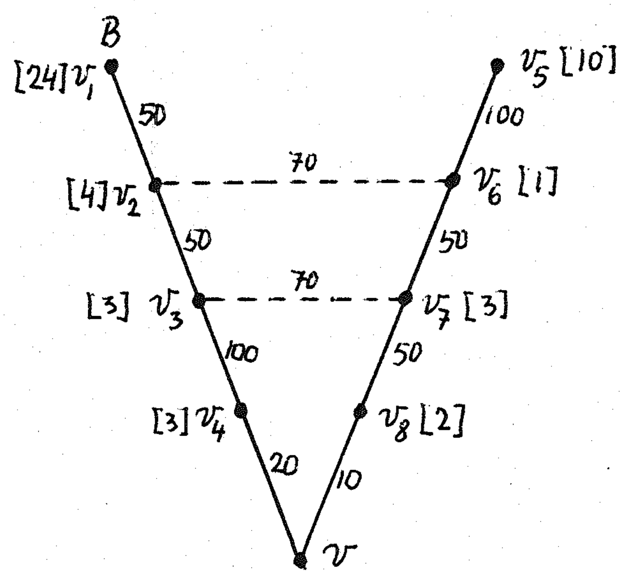


Fig. 19

Example. Consider the network from Fig. 19. The total number of students in the whole fork is $S = 50$. Assume that the available bus fleet is $\mathbf{C} = (30,10,10)$ and the time allowed for routing along the fork (up to node v) is $T = 250$.

Branch B can be served by the two capacities $c_1 = 30$ and $c_2 = 10$. Thus $\mathbf{D}_B = 6$ and for a k-block on branch B we get $k = 4$. The k-block closest to the school is $\{v_3,v_4\}$. Unfortunately, it cannot be used to find a detour completely filling the bus with capacity 10. The situation looks different if we use a more distant k-block $\{v_2,v_3\}$. In this case we can find a route (v_6,v_2,v_3,v_7,v_8,v) containing a detour of type 3, and the whole fork can be served completely by the buses of fleet **C**.

Thus if we cannot find an appropriate detour when routing nodes of branch B then we will shift the k-block farther away from root v, if possible, and continue our search for a detour of one of the three types remembering always the best detour found so far with its corresponding $\mathbf{D}_m$. If we cannot continue in this manner any longer and we have not filled the bus fully within the fork then we take the best detour found with the maximal value $\mathbf{D}_m$ $(<\mathbf{D}_B)$ and try to fill the remaining available seats in the bus on its way from v to the school.

If the branch contains critical points then we determine appropriate routes using a modified strategy. It takes into account the relationship between the number of students at the critical points and the number of students along the whole branch.

A similar modified strategy is used when it is established that the minimal bus capacity c_j, as defined in Section III.d, must be replaced by a smaller capacity c $(<c_j)$ due to the fact that there are not enough students along a particular route. In that case we start with the most distant (from the school) location p within the fork and determine a different route passing through other nodes of the network so that there are as many students as possible along the new route. If even then the number of students is so small that a capacity $c < c_j$ must be used then it means that the value of the objective function will increase as a consequence of the current situation.*)

This basically completes our description of the strategy used in handling a fork. As soon as branch B of a fork is fully served then it is deleted from the network and all other relevant information is updated. After that we proceed to analyze a fork still remaining in the reduced network by applying to it the same strategy until the whole outer district is served.

There are some fork configurations for which our method will not discover optimal detours in spite of the fact that they

---

*) In order to prove whether this situation could have been avoided we must resort to enumeration of all possibilities - a choice that we do not want to consider for practical reasons.

exist. The following example illustrates this.

Example. Consider the network of Fig. 20 which is only a small modification of network from Fig. 18 (exchange of the lengths of edges (v_2,v_3) and (q,v_3)).

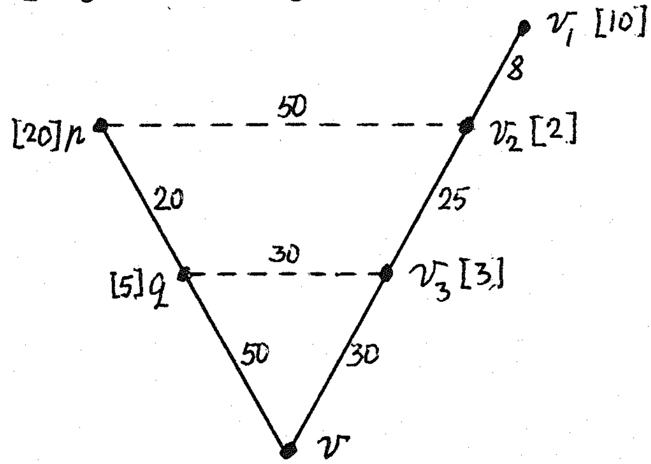


Fig. 20

Under the same conditions as for Fig. 18 it is not possible to extend the k-block {p,q} of branch B using one of the three detour types. On the other hand, route (q,p,v_2,v_3,v) represents a solution which would be discovered by our algorithm if the nodes $\{v_2,v_3\}$ of the other branch served as a k-block for a detour (with appropriately modified student distribution within the network).

If a detour is discovered then it does not mean that the obtained route is really the shortest one possible for routing that set of nodes. A reordering of the nodes can lead to a shorter route as we saw it already in the example of Fig. 16 with paths W_5 and W'_5.

The strategy described here for handling forks of the first degree is applicable only to the forks with their root in the outer district $\mathbf{V_1}$. We will change the strategy if the reduced network contains still some nodes in $\mathbf{V_1}$ which belong to forks with roots in the inner district $\mathbf{V_0}$. In this case we start with an appropriate node $p \in \mathbf{V_1}$ and determine a route passing through a node $q \in \mathbf{V_1}$ (not necessarily different from p) with the following properties:

1. All nodes of the initial section (p,q) of the route belong to $\mathbf{V_1}$.
2. The whole route (p,q,School) can be served within time T_{max}.
3. The total number of students S along the initial section (p,q) is maximal.

The number S is then used for determination of an appropriate bus capacity which will serve the route found or its part. This method is applied iteratively until the whole outer district $\mathbf{V_1}$ is fully served.

To conclude our description of the methods used for route generation, we turn our attention to serving the inner district $\mathbf{V}_0$ (see Section III.a). We will assume that the total number of students in the outer district $\mathbf{V}_1$ is greater than the corresponding number in the inner district $\mathbf{V}_0$ (we assume that this is the case in most practical situations). If we ignore the topological structure of $\mathbf{V}_0$ then we can say that in general we have more buses available for serving $\mathbf{V}_0$ then necessary. Since these buses are used for the second time, after completing routes through the outer district, their **second routes are closed** and the school is both their starting and terminal node. We want to minimize the total number of buses used; therefore our main concern now is to find out whether the buses which served the outer district $\mathbf{V}_1$ can also serve the inner district $\mathbf{V}_0$. If this is possible then we can use as a secondary requirement the condition that the routes generated are the shortest ones. Of course, this is not necessary since it cannot in any case change the value of our objective function.

On the other hand, if the total number of students in $\mathbf{V}_0$, $S(\mathbf{V}_0)$, is greater than $S(\mathbf{V}_1)$ then the service of $\mathbf{V}_0$ can be accomplished in two different ways. First, we can ignore the topological properties of $\mathbf{V}_1$ and find a solution for $\mathbf{V}_0$ and after that try to solve the problem for $\mathbf{V}_1$ using the bus capacities which are provided by the solution for district $\mathbf{V}_0$. The second approach is to analyze and serve the outer district $\mathbf{V}_1$ first and use the buses serving $\mathbf{V}_1$ for service in the inner district $\mathbf{V}_0$ *) and, if necessary, include further appropriate bus capacities.

We have seen that routing of the outer district $\mathbf{V}_1$ is a very complex task requiring special attention. For this reason we have chosen in our computer system the second alternative described above. No matter what the relationship between the total number of students in $\mathbf{V}_1$ and $\mathbf{V}_0$ is, we start with the routing of the outer district $\mathbf{V}_1$. In this manner we obtain a totally ordered bus fleet

$$\mathbf{C}' = \{c'_1,\dots,c'_s : c'_i \in \mathbf{C}\}, \text{ where } c'_i \geq c'_j \text{ for } i < j \text{ and } i,j=1,\dots s,$$

which is now fully available for serving the inner district $\mathbf{V}_0$.

In order to determine a closed route for a bus within $\mathbf{V}_0$ we start from the school and go to its nearest neighbour l ($l \in \mathbf{V}_0$). We denote the time needed for this trip by $t(sch,l)$. Then we try to add other nodes if possible. As mentioned earlier, any added node must lie within an ellipse. In general, if $l \in \mathbf{V}_0$ is the end node on a partly formed route then another node $l_i \in \mathbf{V}_0$ can be added to the same route only if

*) It is sometimes necesary, as we know, to increase the minimal number of buses MIN, calculated at the outset for $\mathbf{V}_1$. With this enlarged bus fleet it can be now possible to serve the whole $\mathbf{V}_0$.

$$(8) \qquad t(l,l_i) + t(l_i,sch) \leq WT_{max} - t',$$

where WT_{max} is the upper bound on the total time of any route within $\mathbf{V_0}$ (determined earlier) and t' is the time spent when going from the school to l.

The generation of a closed route requires successive application of (8) while observing the capacity restriction for the route formed. By an iterative application of this procedure we can completely cover district $\mathbf{V_0}$ by a set of closed routes $\mathbf{R_1} = \{R_1,\ldots,R_q\}$. The number q obtained is not necessarily the minimal number of closed routes, q_{min}, which can be determined for $\mathbf{V_0}$.

Example. Consider the network of Fig. 21 with $WT_{max} = 35$ and the assumption that all students can be picked up by a single bus.

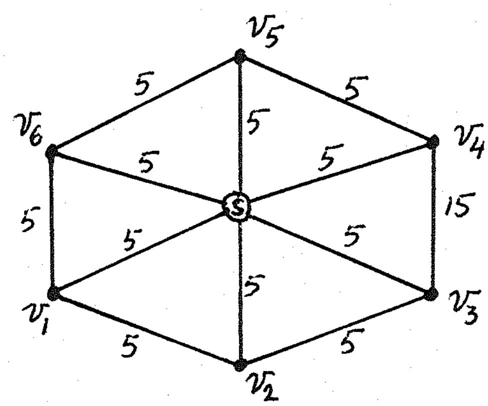


Fig. 21

A covering of $\mathbf{V_0}$ by closed routes depends on the order in which nodes are added to the partially formed route. If, for example, we travel first to node v_1, then to v_2, etc., then we obtain two closed routes:

$$R_1 = (sch,v_1,v_2,v_3,v_4,sch) \text{ and}$$

$$R_2 = (sch,v_5,v_6,sch).$$

On the other hand, if v_4 is the first node visited then there is a possibility of obtaining one single route

$$R = (sch,v_4,v_5,v_6,v_1,v_2,v_3,sch)$$

which is, of course, the most favorable result we can get, showing that $q_{min} = 1$. The decision whether knowledge of q_{min} is desirable or not depends on both the total number of students of the inner district $\mathbf{V_0}$, $S(\mathbf{V_0})$,*) and the capacities of the buses in the original available bus fleet **C**. If p is the minimal number of buses from **C** which is sufficient for transporting $S(\mathbf{V_0})$

students (ignoring the time restriction), then a lower bound s_0 on the actual number of buses needed to serve $\mathbf{V}_0$ is given by

$$s_0 = \max(q_{min}, p). \tag{9}$$

If s_1 is the calculated number of buses which serve the outer district $\mathbf{V}_1$ then we can see from (9) that these s_1 buses are not sufficient for transporting $S(\mathbf{V}_0)$ students of district $\mathbf{V}_0$ if $s_1 < p$. On the contrary, if $s_1 \geq p$ then there is a possibility of serving district $\mathbf{V}_0$ with s_1 buses if $s_1 \geq q_{min}$.

If $S_{\mathbf{C}'}$ is the total sum of the capacities of bus fleet $\mathbf{C}'$ which serves the outer district $\mathbf{V}_1$, then a comparison of $S_{\mathbf{C}'}$ with $S(\mathbf{V}_0)$ provides additional information. If $S_{\mathbf{C}'} \geq S(\mathbf{V}_0)$ then (ignoring any time restrictions) there is a sufficient number of buses available. If $S_{\mathbf{C}'} < S(\mathbf{V}_0)$, that does not necessarily mean that the total number of buses must be increased. Among other things, it could be possible to replace capacities c'_i $\mathbf{C}'$ by unused capacities c_i $\mathbf{C}$ such that $c_i > c'_i$ and thus increase the overall transportation capacity without increasing the total number of buses. Such a substitution of buses by buses with larger capacities does not have any influence on the routes previously determined for $\mathbf{V}_1$. It only increases the number of unused seats in the buses.

While the determination of value p in (9) is easy and quick, the determination of the exact value q_{min} is a very difficult combinatorial problem. For that reason we will avoid its calculation in our computer system.

Thus we start the generation of routes for $\mathbf{V}_0$ by determining a closed route R_1 using repeatedly (8) and calculating simultaneously the number of students $S(R_1)$. As we already know route R_1 does not necessarily have the shortest possible length among all the routes passing through the same points (i.e. it is not necessarily an optimal solution to a traveling salesman problem). An attempt to improve its length would have sense only if its shortening had as a result the possibility of including additional nodes from $\mathbf{V}_0$ into it. Whether we can obtain an improvement of the value of our objective function in this way cannot be said in advance since the value depends also on the distribution of students among the nodes of the network and available bus capacities. Since the time required to find an exact solution to a traveling salesman problem is high even for a small number of nodes visited, we will not pursue possible improvements of the routes found which may not influence the value of the objective function.

---

*) Note that while serving the outer district $\mathbf{V}_1$ it is possible that some students from $\mathbf{V}_0$ are also served by a bus which is not fully loaded in the outer district. In the following we denote by $S(\mathbf{V}_0)$ that number of students which is left in the inner district $\mathbf{V}_0$ after district $\mathbf{V}_1$ has been fully served.

Now we face the problem of assigning an appropriate available bus capacity to the route R_1 containing $S(R_1)$ students. In the first place, the buses from set $\mathbf{C'}$ should be considered but we may replace a bus from $\mathbf{C'}$ by a bus from the complement of $\mathbf{C'}$ (= $\mathbf{C}$ - $\mathbf{C'}$) in order to find a bus with sufficiently large capacity. For example, if $\mathbf{C} = \{60,50,40,30,20\}$, $\mathbf{C'} = \{50,30,20\}$, and $S(R_1) = 60$ then the transportation can be accomplished either by the bus from $\mathbf{C'}$ with capacity $c'_1 = 50$ and 10 students will be left behind to be picked up by another bus or by the bus from complement of $\mathbf{C'}$ with capacity $c_1 = 60$. If we decide for the latter possibility which allows serving the entire route R_1 then the set $\mathbf{C'}$ must be modified to $\{60,30,20\}$ and its complement to $\{50,40\}$. It is difficult to say in advance which one of the two possibilities should be chosen and what consequences follow from the choice without some additional work.

The inequality

$$S_{\mathbf{C'}} < S(\mathbf{V_0}) \tag{10}$$

can be used to help us in our decision how to proceed. If (10) is not satisfied, i.e. if $S_{\mathbf{C'}} \geq S(\mathbf{V_0})$, then we can chose the first option. If (10) is satisfied then we see immediately that the bus fleet $\mathbf{C'}$ serving the outer district $\mathbf{V_1}$ is not sufficient for serving the inner district $\mathbf{V_0}$. Nevertheless, there is still a possibility that by replacing some buses in $\mathbf{C'}$ by larger capacity buses from complement of $\mathbf{C'}$, if possible, the size of the new $\mathbf{C'}$ will remain unchanged and $\mathbf{C'}$ will be sufficient for serving $\mathbf{V_0}$. This exchange technique will be used systematically in our search for appropriate second routes serving the inner district $\mathbf{V_0}$. If such exchanges are not possible while (10) is valid then the size of bus fleet $\mathbf{C'}$ must be increased in order to find a solution to the given problem.

V. REMARKS ABOUT THE COMPUTER SYSTEM

In the description given we tried to explain the basic viewpoints we used in solving our version of the school bus routing and scheduling problem. It seems simple in its formulation but is in fact a very difficult combinatorial optimization problem. The analysis done provides a basis for a heuristic solution of the problem given.

The present version of the computer system we developed represents a basic solution method for our problem. Since the system has a modular structure, it can be expanded to incorporate additional techniques for dealing with additional requirements which can arise in practice. Such extensions might deal, for example, with questions of interconnection and simultaneous handling of several school districts or different starting times for different grades in a single school. In the cases mentioned new techniques must be developed for handling the additional requirements. Another question is the practical applicability of the schedules generated by the system. For example, one of our assumptions was that every bus used had to be available at a certain location and a certain time to start its route. One can imagine a situation when this assumption is not always practical, e.g. when a line bus used for public transport is at a certain time converted into a school bus. Additional constraints may be formulated, implemented, and added to the system we developed.

VI. AN EXAMPLE

We will now illustrate the algorithm developed using an example with real data. Description of the network, numbers of students, bus capacities, time restrictions are taken from an existing school district. They are given as follows:

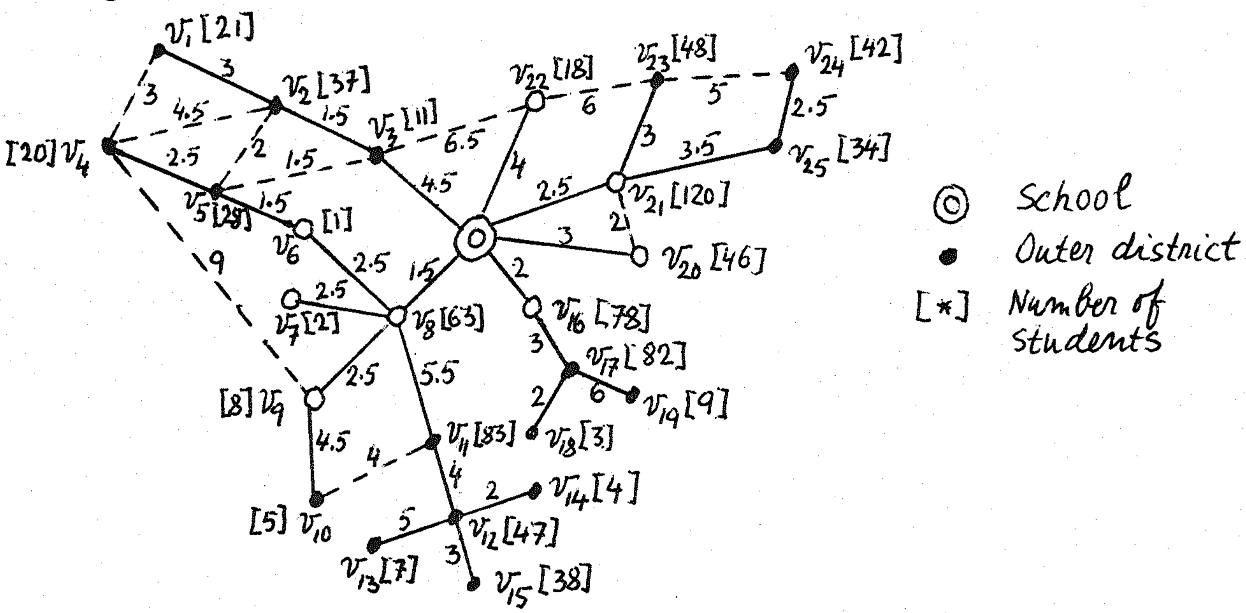


Fig. 22a: Network of a school district

Average speed of a bus: v = 32km/h.
Maximal riding time allowed: T_{max} = 30min. ≙ 16km.
Maximal waiting time allowed: WT_{max} = 15min. ≙ 8km.
Radius of $\mathbf{V_0}$ = 4km.
Available bus fleet **C** = {c_i : i=1,...,22} is given by **C** = {95,80,76,75,74,73,69,66,66,30,27,27,26,26,22,21,21,14,9,8,8,8}.
Total number of students in the district = 855.
Number of students in $\mathbf{V_0}$ = 336.
Number of students in $\mathbf{V_1}$ = 519.
Network G is depicted in Fig. 22a; the solid lines represent the

tree of shortest paths from the school to all other nodes, the dashed lines are the other edges of the network. The distances are indicated in kilometers, the number of students at each location is given in square brackets.

Using the available bus fleet **C** we can determine the (absolutely) minimal number of buses, MIN, necessary for transportation of all students. Since

$$\sum_{i=1}^{7} c_i = 542 \ (>519) \text{ and } \sum_{i=1}^{6} c_i = 473 \ (<519),$$

the number is 7.

The minimal bus representation for 519 is given by

$$\mathbf{C} = \{95,76,74,73,69,66,66\}, \qquad \mathbf{D} = 0.$$

We note that the minimal capacity is equal to 66.

Subgraph $\mathbf{T}_1$ generated by the set of nodes $\{v_{16},v_{17},v_{18},v_{19}\}$ is an isolated component of G (see Sectin III.c) since every path from a node in $\mathbf{T}_1$ to any other node not in $\mathbf{T}_1$ passes through the school. Note that after the first arrival to the school, a bus may start its second route and therefore such a bus cannot serve any more nodes from the outer district $\mathbf{V}_1$ due to the time restrictions. Thus, those buses from **C** which serve nodes from $\mathbf{T}_1 - \mathbf{V}_0 = \{v_{17},v_{18},v_{19}\}$, containing 94 students, can serve only nodes from the inner district $\mathbf{V}_0$ after serving the specified nodes in the outer district $\mathbf{V}_1$. Consequently, the outer district $\mathbf{V}_1$ is decomposed into two disjoint components with total number of students being 94 and 519 - 94 = 425.

Now we can determine whether the decomposition of 519 into 94 and 425 implies an increase in the necessary number of buses. First we can see that the minimum number of buses from **C** needed for transportation of 94 students is 1 and the corresponding representation (see Section III.b) **B** of 94 in **C** is **B** = {95}. To serve 425 students using buses from the complement of **B** (= **C-B**) we need at least 6 buses since

$$\sum_{i=2}^{7} c_i = 447 \ (>425) \text{ and } \sum_{i=2}^{6} c_i = 378 \ (<425).$$

Thus the decomposition of 519 into 94 and 425 does not necessarily lead to an increase of the required number of buses. In order to answer the question whether the bus with capacity $c_1 = 95$ can really serve all nodes of $\mathbf{T}_1-\mathbf{V}_0$ within the allowed riding time we can use the following analysis. $\mathbf{T}_1-\mathbf{V}_0$ consists of (exactly) one fork of the first degree with branching point v_{17}. Both branches are not only mutually isolated but they are also isolated from the rest of the network (see Section III.b). The solution of the corresponding knapsack problem gives us the following route:

Route #1 Bus capacity c = 95

Location	Number of students
v_{19}	9
v_{18}	3
v_{17}	82
v_{16}	1
School	

In the solution we left out the exact departure times at the individual locations which depend on both the class starting time and the location distances from the school along the route specified.

After finding a route and the corresponding bus we reduce both the given network G and the available bus fleet **C**. First, we remove nodes v_{17}, v_{18}, v_{19} from G, and decrease the number of students at node v_{16} by 1. Then we eliminate capacity $c_1 = 95$ from the bus fleet **C**.

Nodes v_8, v_{12}, v_{21}, and the school are branching nodes in the remaining network, therefore they are roots of forks. Using our strategy we select the most distant node from the outer district which is a root of a fork of the first degree. In this case v_{12} is the root of the following fork of the first degree

$$\mathbf{F} = \{v_{12};\ v_{13}, v_{14}, v_{15}\}.$$

The total number of students within this fork is 49 if we do not count the students at the root and it is 96 if we count them. We have again a completely isolated fork and therefore we will use the knapsack problem to find a route. Because the distance of the root v_{12} from the school is 11km and a route can be at most 16km long, it is obvious that the nodes v_{13}, v_{14}, and v_{15} must be served by at least three buses. None of the buses can serve any two of the leaves v_{13}, v_{14}, v_{15} and therefore the candidates for the routes are the shortest paths from v_{13}, v_{14}, v_{15} to the school. In addition none of the buses can make a detour through some nodes of the outer district $\mathbf{V}_1$ on their way to the school. Only v_{11} will be traveled through by all the buses.

This fact plays a substantial role in our algorithmic strategy. We will emphasize it by a discussion of two alternatives.

The first alternative: Since every bus used to serve the fork can be used also for serving the root of the fork, i.e. node v_{12}, we show whether there is at least one bus representation for 96 (= the total number of students in the fork including the root) of length 3. Representations with this property exist and are listed below:

$$\mathbf{B}_1 = \{80, 8, 8\}, \qquad \mathbf{D} = 0$$
$$\mathbf{B}_2 = \{74, 14, 8\}, \qquad \mathbf{D} = 0$$
$$\mathbf{B}_3 = \{66, 22, 8\}, \qquad \mathbf{D} = 0$$
$$\mathbf{B}_4 = \{73, 14, 9\}, \qquad \mathbf{D} = 0$$
$$\mathbf{B}_5 = \{66, 21, 9\}, \qquad \mathbf{D} = 0$$

Any of the above representations can be used in our fork since for any representation we can use the corresponding buses in such a way that all nodes v_{13}, v_{14}, v_{15} will be completely served.*) Since for all the representations listed the minimal bus capacities are smaller than 66, therefore the value of the objective function will become greater than 6 no matter which representation is chosen. E.g., if we choose B_1 then we can have the following routes:

a) <u>c = 80</u>		b) <u>c = 8</u>		c) <u>c = 8</u>	
v_{15}	38	v_{13}	7	v_{14}	4
v_{12}	42	v_{12}	1	v_{12}	4
School		School		School	

<u>The second alternative:</u> In this case we will consider all students in the outer district $\mathbf{V_1}$ that can be served by the buses used to serve the fork. In our case nodes v_{11}, v_{12}, v_{13}, v_{14}, and v_{15}, with total number of students equal to 179, will be considered. Representations of 179 (using the bus fleet reduced by capaciy c_1 = 95) are of minimal length three:

$$\mathbf{B}_1 = \{80,73,26\}, \quad \mathbf{D} = 0$$
$$\mathbf{B}_2 = \{80,69,30\}, \quad \mathbf{D} = 0$$
$$\mathbf{B}_3 = \{76,73,30\}, \quad \mathbf{D} = 0$$

Here we can also see that the objective value will increase independently on the choice made. If we select representation $\mathbf{B}_1$ then we have the following routes:

a) <u>c = 80</u>		b) <u>c = 73</u>		c) <u>c = 26</u>	
v_{15}	38	v_{13}	7	v_{14}	4
v_{12}	42	v_{12}	5	v_{11}	22
School		v_{11}	61	School	
		School			

In order to decide which one of the two alternatives should be selected we will make the following comparisons.

If we use the first alternative we will serve 96 students from $\mathbf{V_1}$ out of the total 425 (= 519 - 94). For the transport of the remaining 329 students we have available the following capacities:

$$\mathbf{C} = \{76,75,74,73,69,66,66,30,27,27,26,22,21,21,14,9,8\}.$$

For the transport of 329 students we need at least 5 buses. That means that instead of the original 7 buses we require now at least 9 buses. If we use the second alternative then we will

*) Note that this is not always the case. For example, if the number of students at v_{13} were 10 and at v_{14} only 1 then the representation $\mathbf{B}_1$ would not be applicable but all other representations listed would be applicable.

transport 179 students from the original 425 and therefore there will be 246 students left in $\mathbf{V_1}$ requiring transportation. The following bus fleet is available:

$$\mathbf{C} = \{76,75,74,69,66,66,30,27,27,26,22,21,21,14,9,8,8,8\},$$

so we will need at least 4 buses, i.e. altogether at least 8 buses will be needed.

We can see that the second alternative provides a better choice. A reason for that is the following: We have already determined that for the transportation of 425 students from $\mathbf{V_1}$ using the fleet

$$\mathbf{C} = \{80,76,75,74,73,69,66,66,30,27,27,26,26,22,21,21,14,9,8,8,8\}$$

there are at least 6 buses needed, the smallest of which must have capacity c = 66 (the corresponding bus representation is $\mathbf{B} = \{75,74,73,69,66,66\}$). That means that as soon as a bus with capacity $c < 66$ is used then the transportation of 425 students is not possible any more with only 6 buses but at least 7 buses must be used. The smaller a capacity used ($c < 66$), the larger the probability that more buses than absolutely necessary must be used. And this is just the case with the first alternative when the smallest capacity used is c = 8. While using the second alternative, the smallest capacity used is c = 26 or c = 30. Thus the second alternative is preferable since it can lead to a solution which is closer to the optimum.

It follows from the previous considerations that a good strategy requires use of larger bus capacities as opposed to smaller ones, any time it is possible. This means that in the previous example we will use the second alternative. Of course we cannot say whether one of the three representations listed is better than the two remaining ones and which one is it. If a conflict situation arises later, leading to an increase of the necesary number of buses, it does not mean that this situation would occur also if one of the two remaining representations was used.*) If we use representation $\mathbf{B_1}$ then we obtain the following routes (as mentioned earlier):

<u>Route #2: c = 80</u>		<u>Route #3: c = 73</u>		<u>Route #4: c = 26</u>	
v_{15}	38	v_{13}	7	v_{14}	4
v_{12}	42	v_{12}	5	v_{11}	22
School		v_{11}	61	School	
		School			

If we remove the nodes served from the network we obtain the

*) This situation is similar to that of a chess player who knows that when he moves in a certain manner his opponent will not be able to give him check-mate in the next move but that this can happen in one of the following moves.

reduced network of Fig. 22b.

The available bus fleet is now

$$C = \{76,75,74,69,66,66,30,27,27,26,22,21,21,14,9,8,8,8\}.$$

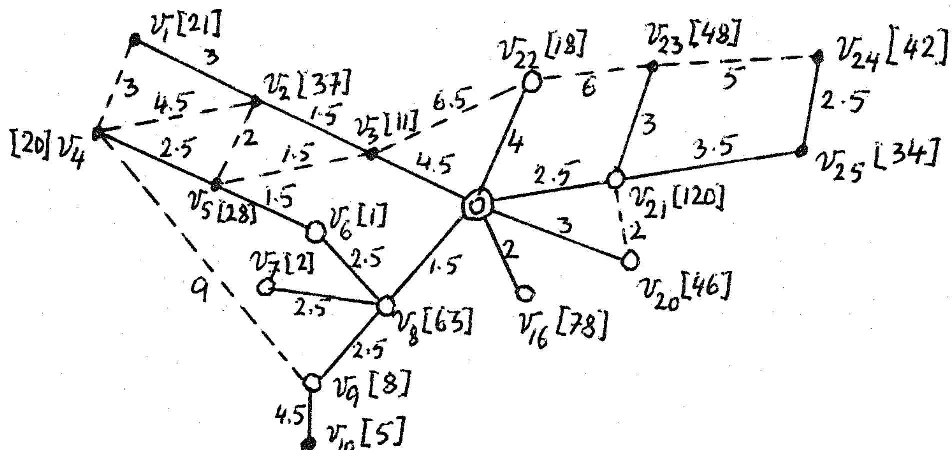


Fig. 22b: Reduced network after route #4.

All roots of the forks of the first degree now lie in the inner district $\mathbf{V_0}$. Thus we will generate a route through nodes of $\mathbf{V_1}$ with the property that the total number of students along the route is high enough to fill an available bus. The route passing through nodes v_4, v_1, v_2, and v_3 contains S = 89 students, which is sufficient to fill the largest still available bus with capacity c = 76.*) We get thus

Route #5:	c = 76	
	v_4	20
	v_1	21
	v_2	35
	School	

The reduced network is given in Fig. 22c and the reduced bus

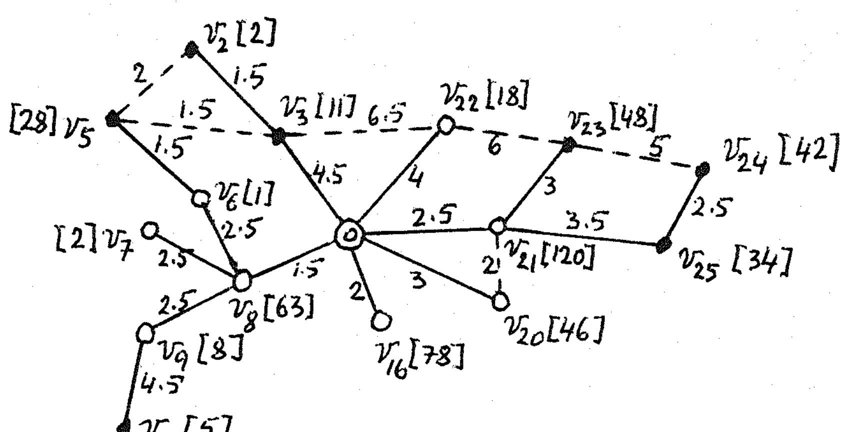


Fig. 22c

*) The fact that we can fill the largest available bus by students from $\mathbf{V_1}$ is, of course, the best we can hope for.

fleet is

C = {75,74,69,66,66,30,27,27,26,22,21,21,14,9,8,8,8}.

These buses are now available for serving the remaining 170 students still left in the outer district $\mathbf{V_1}$.

Using the same approach we will find as the next route the one passing through nodes v_{24}, v_{25}. It contains 76 students, 75 of them can be transported by the largest available bus with capacity c = 75.

<u>Route #6:</u>	<u>c = 75</u>	
	v_{24}	42
	v_{25}	33
	School	

Fig. 22d shows the reduced network. The bus fleet becomes

C = {74,69,66,66,30,27,27,26,22,21,21,14,9,8,8,8}.

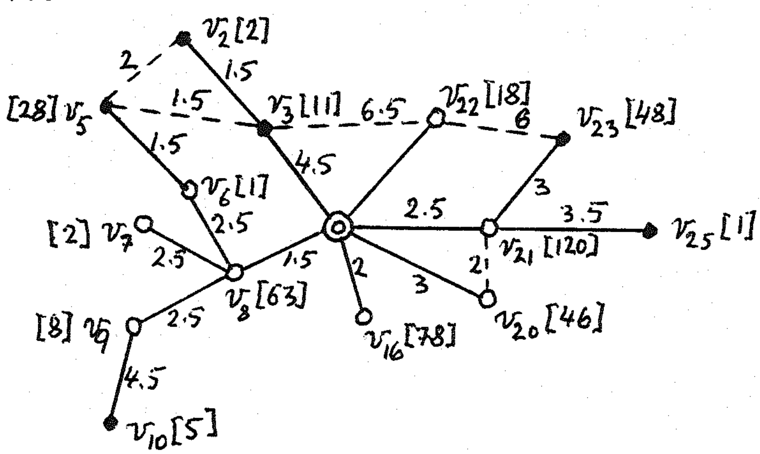


Fig. 22d

Next we find out that the route through nodes v_{25} and v_{23} contains most students from $\mathbf{V_1}$: 49. We can serve these students with the smallest bus capacity greater than or equal to 49, i.e. c = 66. The 17 free seats can be filled, if possible, by students from nodes in the inner district $\mathbf{V_0}$, that are visited by the bus on its way to the school. We obtain the following route:

<u>Route #7:</u>	<u>c = 66</u>	
	v_{25}	1
	v_{23}	48
	v_{21}	17
	School	

The reduced network is shown in Fig. 22e and the available bus fleet becomes

C = {74,69,66,30,27,27,26,22,21,21,14,9,8,8,8}.

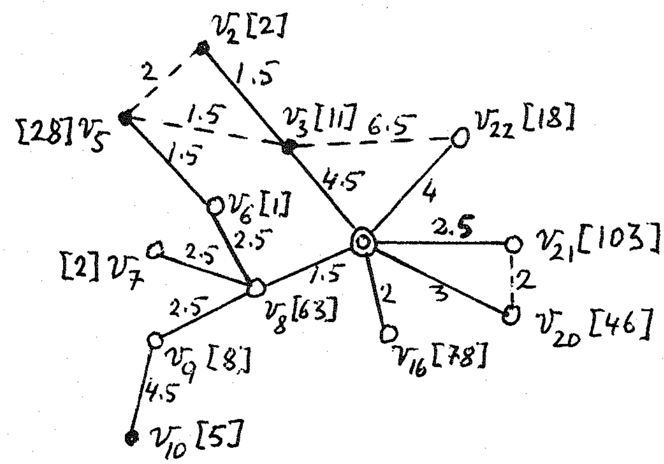


Fig. 22e

The time restriction does not allow service of the remaining nodes in $\mathbf{V_1}$, i.e. v_2, v_3, v_5 and v_{10}, by a single bus. As a suitable bus route we can get

Route #8:	c = 66	Route #9:	c = 8
v_3	11	v_{10}	5
v_2	2	v_9	3
v_5	28	School	
v_6	1		
v_8	24		
School			

The outer district is now fully served by the 9 routes found. All of the remaining nodes waiting for service are in the inner district $\mathbf{V_0}$. The reduced network is given in Fig. 22f. We will take the 9 buses used for serving the outer district as the available bus fleet, i.e.

$$C = \{95, 80, 76, 75, 73, 66, 26, 8\}.$$

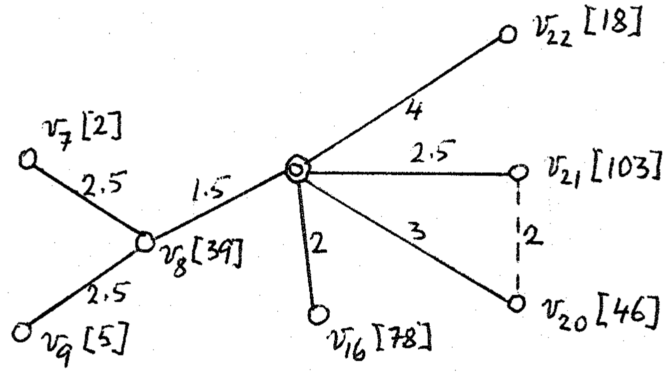


Fig. 22f

We start the generation of a closed route by considering the node closest to the school, i.e. v_8. On the basis of condition (8), the only nodes reachable are v_7 or v_9. If we decide to visit

v_7 then we cannot visit any other node. Thus we get a route with 41 students that can be served by a bus with capacity 66.

Closed route #1: c = 66 *)

School
v_8 39
v_7 2
School

In a similar manner we obtain:

Closed route #2: c = 8

School
v_9 5
School

Now v_{16} is the unserved node closest to the school. We can serve it using capacity c = 80 and get:

Closed route #3: c = 80

School
v_{16} 78
School

Continuing in a similar manner we get:

Closed route #4: c = 95

School
v_{21} 95
School

Closed route #5: c = 66

School
v_{21} 8
v_{20} 46
School

Closed route #6: c = 26

School
v_{22} 18
School

This completes the route and schedule generation for the whole network.

The strategy we developed and used in our computer system gives us as a result 9 buses for serving the whole network. 6 buses make a second tour through the inner district. For every bus used its route and timetable is printed out.

The results obtained represent one of many possible solutions. We can now try to modify and improve the routes without changing the value of the objective function. We can

*) Note that from the combinatorial point of view it does not matter in which direction the close route is traversed. Of course, to provide a better service to the students, the opposite direction is more convenient.

introduce other practical view points which can make the final result more suitable. For example, we can see that route #9 does not have to pick up students from node v_9 which is more suitable for students in v_9. One other possible change was described in footnote on p. 47. An important question arises when the final result generated requires more buses than MIN - the value determined in Section IVa which is based only on the available bus capacities and the total number of students in $\mathbf{V_1}$. The question is whether the value of the objective function found by the algorithm is really optimal or not and if not, whether it could be improved. In our discussion of the two alternatives we saw that it was better to use larger capacities whenever it was possible. Since our computer program is designed in such a manner that a small capacity is used, if the local topological properties require it, we can try to improve the objective value by an appropriate modification of the available bus fleet and running the program again with this changed fleet. For example, we can eliminate from the bus fleet some of the smallest capacities.*) In the example discussed in this section we could replace the original bus fleet

$$\mathbf{C}=\{95,80,76,75,74,73,69,66,66,30,27,27,26,26,22,21,21,14,9,8,8,8\}$$

by $\mathbf{C_1} \subseteq \mathbf{C}$; where

$$\mathbf{C_1} = \{95,80,76,75,74,73,69,66,66,30,27,27,26,26,22,21,21,14,9\}$$

or

$$\mathbf{C_2} = \{95,80,76,75,74,73,69,66,66,30,27,27,26,26,22,21,21\}$$

or

$$\mathbf{C_3} = \{95,80,76,75,74,73,69,66,66,30,27,27,26,26,22\}$$

etc.

Thus the program can be used to decide whether it is possible to find feasible solution which uses only the "larger" available buses. Iterative application of the coputer system can be used for finding out whether the suboptimal solution found during the first run of the program can be improved upon.

*) Of course, the remaining bus fleet must contain at least MIN buses.

REFERENCES

[1] Wittmann, P.: Der Schuelerverkehr in laendlichen Regionen. Verkehrswissenschaftliche Studien 33, Vandenhoek & Ruprecht, Goettingen, 1980.

[2] Newton, R.: Bus Routing in a Multi-School System. Ph.D. Thesis, 1970, State University of New York at Buffalo.

Carleton University
School of Computer Science

Bibliography of SCS Reports

SCS-TR-1 THE DESIGN OF CP-6 PASCAL
Jime des Rivieres and Wilf R. LaLonde, June 1982.

SCS-TR-2 SINGLE PRODUCTION ELIMINATION IN LR(1) PARSERRS:A SYNTHESIS
Wilf R. LaLonde, June 1982.

SCS-TR-3 A FLEXIBLE COMPILER STRUCTURE THAT ALLOWS DYNAMIC PHASE ORDERING
Wilf R. LaLonde and Jim des Rivieres, June 1982.

SCS-TR-4 A PRACTICAL LONGEST COMMON SUBSEQUENCE ALGORITHM FOR TEXT COLLATION
Jim des Rivieres, June 1982.

SCS-TR-5 A SCHOOL BUS ROUTING AND SCHEDULING PROBLEM
Wolfgang Lindenberg, Frantisek Fiala, July 1982.

SCS-TR-6 ROUTING WITHOUT ROUTING TABLES
Nicola Santoro, Ramez Khatib, July 1982.

SCS-TR-7 CONCURRENCY CONTROL IN LARGE COMPUTER NETWORKS
Nicola Santoro, Hasan Hural, July 1982.

SCS-TR-8 ORDER STATISTICS ON DISTRIBUTED SETS
Nicola Santoro, Jeffrey B. Sidney, July 1982.

SCS-TR-9 OLIGARCHICAL CONTROL OF DISTRIBUTED PROCESSING SYSTEMS
Moshe Krieger, Nicola Santoro, August 1982.

SCS-TR-10 COMMUNICATION BANDS FOR SELECTION IN A DISTRIBUTED SET
Nicola Santoro, Jeffrey B. Sidney, September 1982.