

A COMPUTER SYSTEM FOR BUILDING COMMUNITY BUS  
PICK-UP ROUTES

P Stampoultzis  
Manager  
Information Technology Branch

L Chambers  
Manager  
Operations Research and  
Information Branch  
Department of Transport  
Western Australia

ABSTRACT

*The paper describes a computer program developed at the Western Australian Department of Transport that solves the Travelling Salesman Problem. There are a number of community transport bodies that have highly variable pickup routes and the use of traditional algorithms, because of required computation times, cannot be used to define travel routes.*

*The program employs a heuristic originally developed by J J Bartholid III and L K Platzman of the School of Industrial and Systems Engineering, Georgia Institute of Technology. This heuristic solves the TSP mainly by the simple expedient of sorting a list of factors calculated from the defined space filling curve. The TSP solution so generated is generally 25% longer than optimal but can be reduced by human editing of the route generated.*

*Route additions are simple to incorporate and require calculations only for the new pickup points and their insertion into the sorted list. Deletions require removal from the list and no additional calculations are needed.*

*The system allows daily pickup routes to be developed and printed for each driver. The software also produces statistics that are useful in the management of the vehicle fleet being employed.*

## A Computer System for Building Community Bus Pickup Routes.

The paper describes a computer program being developed by the Western Australian Department of Transport that solves the Traveling Salesman Problem. There are a number of community transport bodies that have highly variable customer listings and the use of traditional algorithms, because of required computation times, cannot be used to develop pickup routes.

The program employs a heuristic originally developed by J.J. Bartholdi III and L.K. Platzman of the School of Industrial and Systems Engineering, Georgia Institute of Technology. This heuristic solves the TSP by the simple expedient of sorting a list of values calculated from a defined space filling curve. The TSP solution so generated is generally 15% longer than optimal but may be able to be reduced by manual editing of the generated route.

Route additions are simple to incorporate and require calculations only for the new pickup points and their insertion into the sorted list. Deletions require removal from the list with no additional calculations.

The system allows daily pickup routes to be developed and printed for each driver. The software also produces statistics that are useful in the management of the vehicle fleet.

Community transport agencies fulfil needed services in the community and also hold strategically important positions in the total transport portfolio. Given their importance in their market niche and the fact that all western cities, of even moderate size, have these agencies it is a pity that many of them suffer from resource shortages. The communities they service, generally disadvantaged groups, are not in a position to fund these agencies

at the levels necessary to provide wide ranging or complete needs satisfaction. If it becomes possible to offer a service at either a lower cost or greater efficiency then satisfaction can be increased for both those being serviced and those offering the service, either directly or indirectly via funding.

A number of journal articles have described the use of computer systems in assisting the development of fleet pickup routes. (Belardo et al 1985, Fisher et al 1982, Bodin and Berman 1979, Swersey and Ballard 1984, Beltrami and Bodin 1973, Avramovich et al 1982) The objective of this paper is to describe a Decision Support System (DSS) being developed to assist these agencies to manage their vehicle fleets more efficiently. It is hoped that the system, once fully developed, will provide 'good' solutions to fleet management problems as opposed to 'optimal' ones. It was also decided, based on work by Remus (1984) to maintain a graphical user interface. Hence all route development and editing, where possible, is presented in graphical form and modified using a mouse pointing device.

The provision of transport services for many agencies is one of the larger budget provisions and hence any savings could have commensurate effects on easing monetary constraints. The routes of transit vehicles are one of the major determinants of vehicle-hours, vehicle-miles, and required fleet size (Jacobs, Skinner and Lemer 1984) and as such the optimization of routes becomes a major requirement in the reduction of costs for community based transport bodies.

One problem of concern in optimizing fleet vehicle allocation is in first discovering the solution to the Traveling Salesman Problem. (TSP) Unfortunately many of the interesting routing problems are NP-complete, including the TSP. It has been conjectured that there exist no efficient algorithms for the solution of the group of NP-complete problems. As a consequence of this lack of efficient algorithms, attention has been devoted to the development of efficient heuristics that solve the various NP-complete problems, however, only approximately (Ibarra and Kim 1975 and Johnson 1974).

Worst-case analysis of heuristics for routing problems has focused on the TSP. It has been shown (Sahni and Gonzales 1976) that if the triangle inequality is not satisfied, the problem of finding an

## A COMPUTER SYSTEM FOR BUILDING COMMUNITY BUS PICKUP ROUTES

approximate solution for the TSP within any constant bound of the optimum is as difficult as finding an exact solution. One advantage of the proposed heuristic is that it is independent of the triangle inequality requirements since inter-node distances/costs are not employed. (Note: Since there are  $N^2$  of these distance/cost relationships in any set of nodes, systems that develop solutions based on these values require at least  $N^2$  computational effort.) The heuristic presented escapes both the inter-node distance/cost penalty and the potential triangle inequality problem.

In the bus routing problem confronting community transport operators the solution of the TSP is only a start, the true problem that requires solving is the  $k$ -person TSP, where  $k \geq 2$ . When one bus cannot handle the route developed then  $k$  buses must be dispatched with an attendant constraint that none of the buses has too large a task. (i.e. does not attempt to pickup more passengers than can be handled by the bus and/or does not extend its travel time excessively)

The buses in use by the organizations the system is being developed for are held at the homes of the drivers. There are a number of advantages and disadvantages that develop from this:

- The drivers have been found to take greater care of the vehicles if they are allowed to take them home. One can speculate as to the reasons but that goes beyond the scope of this paper.

- The buses can be allocated to routes that start close to their starting points thus reducing quite dramatically the required empty travel distances at the start of the tours. This however makes the task of bus allocation to routes an order of magnitude harder to solve.

The aspect of differing fleet vehicle locations at start of morning pickup and multiple destinations introduce a number of problems; however, the system requires that the problem be solved only once for the morning pickups or afternoon pickups, not both. The routes, if optimal for pickup and drop to destination in the morning, are optimal for the afternoon pickup. (Note: although the routes are optimal the bus allocation to routes may need to change.) An example of the problems to be solved is shown in figure 1.

In many situations the numbers of pickups eliminate the use of traditional TSP algorithms because of the large solution times inherent in these techniques. With highly stable pickup listings optimal TSP algorithms can be used for the rare occasion when the routes do need to be modified. However, for those with highly dynamic listings the solution to the TSP for generation of routes must be solved in a reasonable time. (i.e. significant delays are unacceptable since the work needs to be done in real time)

In this paper a modified TSP heuristic (Bartholdi and Platzman 1982) is used to solve the TSP in real time. These solved TSPs are then used as the raw data for a number of modules that allocate pickups to vehicles depending upon route allocation of pickup, vehicle seating capacity, vehicle costs, special pickup requirements, etc.

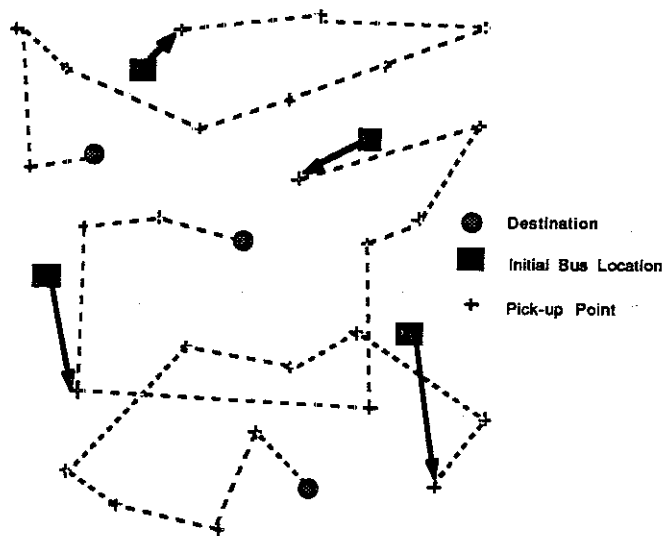


Figure 1

The data required for initial setup are data on each vehicle in the fleet, information on each individual and their pickup and set down points. It is also recognized that some individuals require differing set down points during the week. (i.e. work on Monday, Tuesday, and Friday and medical on Wednesday and Thursday.)

## A COMPUTER SYSTEM FOR BUILDING COMMUNITY BUS PICKUP ROUTES

If the Vehicle Maintenance option is required then a wider range of vehicle data is needed over and above fleet number and seating capacity. Service kilometer intervals, registration date and cost, regular bus driver and regular route number serviced, driver hourly rate, fuel type and capacity, special fixtures and facilities available on the vehicle and regular service needs. Each time a vehicle is refueled or has maintenance performed the details are input to the system. Rough running accounts are maintained on both a per kilometer and per pickup basis, these numbers can be improved by the importation of data from the in-house accounting system. Vehicle availability is known by the system so that vehicles that are in for maintenance, repair, etc. are not allocated to routes and the same applies to drivers when unavailable.

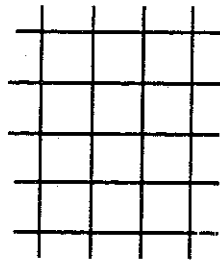
Once routes for each day are developed, an 0,1 Integer Linear Program (ILP) (Kuester and Mize 1973) assigns vehicles optimally to each route. The objective function is simply vehicle running costs. This objective function can be altered to suit any particular function desired and feasible.

Other output from the system will consist of impending vehicle maintenance, pickup listings for each vehicle, reports on liter/km for vehicles, average occupancy, driver and vehicle utilization, next maintenance requirements, warning of impending vehicle registrations, graphs of vehicle maintenance costs and utilization over time, selected output of members by suburb, destination and special requirements.

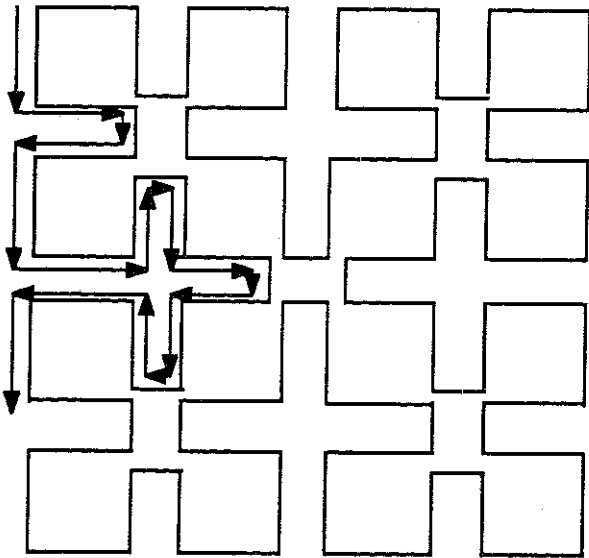
The system is written in Microsoft QuickBasic V4.0. Executable and source code are offered free of charge to selected, non-profit community transport agencies. The selection of QuickBasic was made because of the ease of programming and the low cost of the compiler. This low cost and ease of programming makes it possible for agencies to modify the system if and when they feel a need to do so. The system requires an IBM PC (or clone) XT, AT or 386 with a hard disk, CGA graphics and a Microsoft (or compatible) mouse as a minimum configuration.

The method of solving the TSP is now presented. If we consider the roads of a city to be orthogonal and regular we are presented with a relatively trivial problem for the solution of the TSP.

Consider the diagram below:



If this is a valid representation of the road network that confronts the route developer then figure 2 is an acceptable model of a possible pathway through the road system that covers every road section. If we now locate positions on this street system that correspond to the required pickup points for the organizations clients we have a relatively accurate map of the area of interest. (Note: The proxy road system can be made as dense as is necessary to reflect more accurately the true street network confronted. This is achieved by increasing the order of the system. See figure 3)



**Figure 2.**  
Second order network of Street Design.

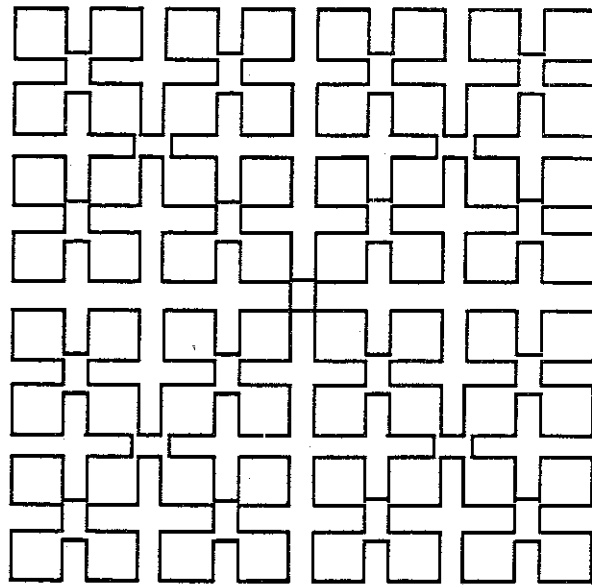


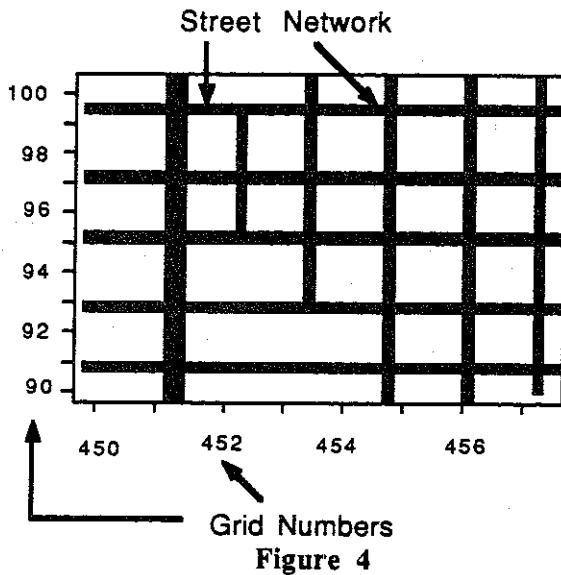
Figure 3  
Third order network of street design

A calculation of the distance from some arbitrary starting point to every pickup point gives a proxy for the travel distance to each pickup point. If these pickup points are now ordered by increasing magnitude of travel distance from the arbitrary origin we have a tour for pickup sequence.

The optimality of these tours is obviously open to question and the resultant tours need to be tested. However it is possible to develop an efficient starting point with ease. A quick look at the Perth road directory, at the densest parts of the city, will give a relatively confident upper bound on the numbers of streets in each orthogonal axis. I selected an upper bound of fifty streets to one map sheet and then multiplied this number by the numbers of map sheets in each direction then selected the largest value. (There are 16 maps by 50 streets = 800 streets at a maximum in any orthogonal direction) This approximation defines the order of our network for street design. The numbers of intersections should be the lowest power of 2 that gives a result greater than 800, which is 2<sup>10</sup>. Given the order we can now start the process of defining the locations of pickup points for the network.



A map of the area of interest is partitioned, by pen or pencil, into a grid that matches the value  $2^{10} = 1024$  found in the previous step in the longest axis and the shorter axis is partitioned employing the same scale (See figure 4).



The location that is used by the system is the closest cross-referenced grid point for the pickup. These values are input into the program, for each pickup, and are then employed to calculate the travel distance proxy for use by the TSP module.

You will note that the partitioning of the road network does not need to be regular. It is quite acceptable to employ differing scales on each axis; this has no effect on the accuracy or calculation times for the programs.

The calculation of the proxy for travel distance is linear with  $N$ . All that is now required is that the resultant distances be sorted. This can be achieved with a sort that will also operate with complexity  $N$ . This gives a heuristic for the solution of the TSP in  $O(N)$ .

The heuristic is accurate to within 15% for generalized TSP problems (Bartholdi and Platzman 1982). Solution times for small problems are not significantly different for any other easily available heuristic or algorithm. The system comes to the fore when there are large problems. A tour for 100 points can be

## A COMPUTER SYSTEM FOR BUILDING COMMUNITY BUS PICKUP ROUTES

developed from scratch within seconds. (see table 1) The code is still in the early stages of development and improvements should see these times decrease even further.

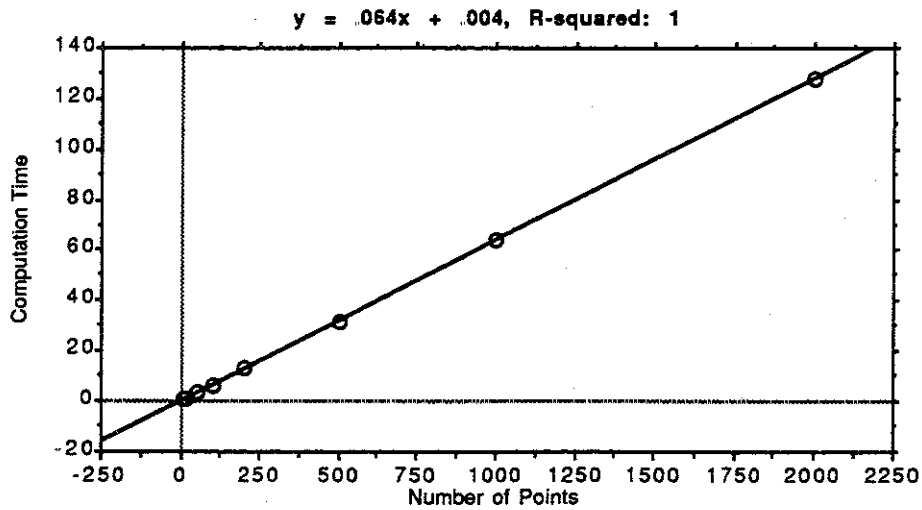
When a new pickup point needs to be included into the tour only the proxy for distance for that point needs to be calculated and slotted into the relevant position in the list. It is this simple insertion procedure that makes this heuristic of great value to organizations with highly dynamic listings. The need to solve the tour from scratch is not required. The ease of programming of this particular algorithm makes it of value to organizations that lack both technical expertise, from an Operations Research perspective, lack the funds to purchase expensive computer hardware and equally expensive route developing software systems.

Below are computation data from test runs of the heuristic. The hardware used was a COMPAQ 386 with numeric coprocessor. As can be seen the computation times are of  $O(N)$  as shown overleaf (graph 1). Each point requires 0.064 seconds to solve and the system requires 0.004 seconds set up time for array dimensioning and parameter initialization.

Computation Time (secs) vs Number of Pickups	
# of Pickups	Calculation Times (Secs)
10	0.66
20	1.26
50	3.19
100	6.42
200	12.80
500	32.02
1000	64.05
2000	128.09

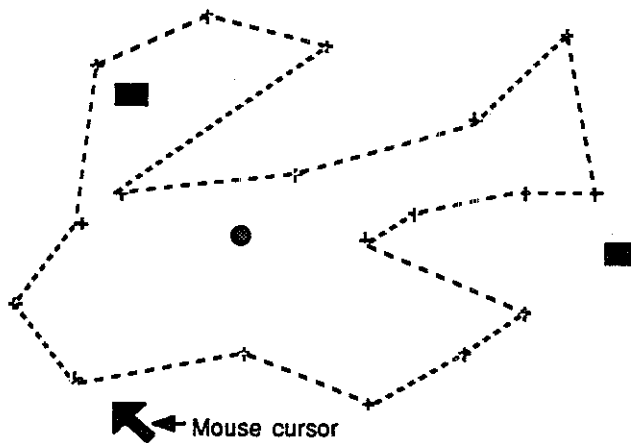
Table 1

The major implication of the speed of calculation is that it becomes possible for routes to be developed, on a daily basis, for highly dynamic and changable pickup listings. Even the cheapest IBM PC system could be employed to perform the required number crunching within a reasonable time.



Graph 1

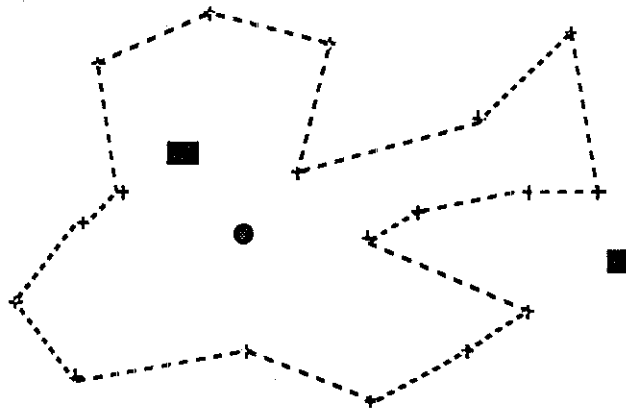
Part of the graphical interface is presented below:



Hypothesised initial solution

The above is a screen image of a hypothesized tour of pickups for one destination. The tour has been developed by the heuristic. If required or desired the tour can be edited to something maybe resembling the next diagram.

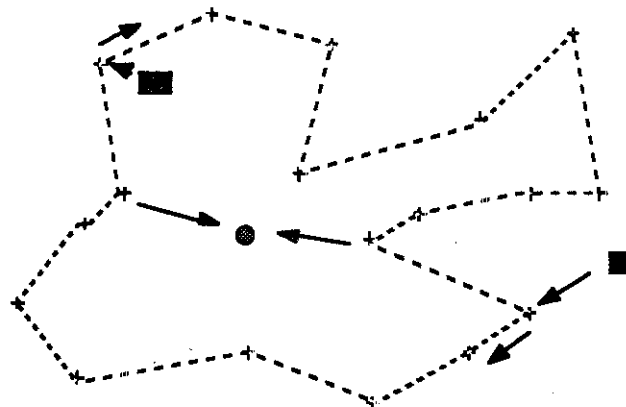
## A COMPUTER SYSTEM FOR BUILDING COMMUNITY BUS PICKUP ROUTES



Improved solution after manual editing

The editing is accomplished by selection of relevant points with the mouse cursor. By manipulation of the mouse buttons it is possible to break the tour and re-establish links where desired.

The next screen picture represents the ILP allocation of buses to the tour. In fact the pickups defined for each bus will be hi-lighted in different colors and these bus pickup schedules can also be edited. All editing is accomplished by the use of the mouse. No keyboard input is required to develop the tours and bus allocations.



Vehicle inclusion by ILP

The system is being designed to be as simple to use as possible while losing none of its potential value. It is hoped that the system will prove to be of use to the organizations that it is aimed towards and that it assists in reducing present and future capital investment in bus purchases and also in the reduction of day-to-day costs.

#### REFERENCES

- Avramovich D, Cook T M, Langston G D and Sutherland F (1982) A Decision Support System for Fleet Management: A Linear Programming Approach, *Interfaces*, June 1982, pp 1-8.
- Bartholdi III J J, Platzman L K (1982) An  $O(N \log N)$  Planar Traveling Salesman Heuristic based on Spacefilling Curves, *Operations Research Letters*, Vol 1 Nos 4, September 1982, pp 121-125.
- Belardo S, Duchessi P and Seagle J P (1985) Microcomputer Graphics in Support of Vehicle Fleet Routing, *Interfaces*, Vol 15 Nos. 6, November-December 1985, pp 84-92.
- Beltrami E J and Bodin L D (1973) Networks and Vehicle Routing for Municipal Waste Collection, *Networks*, Vol. 4, No. 1, 1973, pp 65-94.
- Bodin L D and Berman L (1979) Routing and Scheduling of School Buses by Computer, *Transportation Science*, Vol. 13, No. 2, May, 1979, pp 113-129.
- Christofides N, (1976) Worst-case Analysis of a new Heuristic for the Traveling Salesman Problem, *Management Science Research Report No. 388*, Carnegie-Mellon University, Pittsburgh, PA, 1976.
- Fisher L M, Greenfield A J, Jakimur R, and Lester III J T (1982) A Computerized Vehicle Routing Application, *Interfaces*, August 1982, pp 42-51.
- Ibarra O H, and Kim C E (1975), Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems, *Journal Association Computing Machines*, Vol 22, 1975, pp 463-468.

Jacobs M, Skinner R E, and Lemer A C (1984) Transit Project Planning Guidance: Estimation of Transit Supply Parameters, U.S. Department of Transport, Research and Special Programs Administration Transportation Systems Center, Oct 1984, pp 33.

Johnson D S, (1974) Approximation Algorithms for Combinatorial Problems, Journal of Computer Systems Sciences, Vol 9, 1974, pp 256-278.

Kuester J L and Mize J H (1973) Optimization Techniques with FORTRAN, McGraw Hill, 1973, pp 91-104.

Remus W (1984) An Empirical Investigation of the Impact of Graphical and Tabular Data Presentations on Decision Making, Management Science, Vol 30 Nos. 5, May 1984, pp 533-542.

Sahni S K and Gonzalez T (1976), P-complete Approximation Problems, Journal of Computing Machines, Vol 23, 1976, pp 555-565.

Swersey A J and Ballard W (1984) Scheduling School Buses, Management Science, Vol 30 Nos. 7, July 1984, pp 844-853.