OPSEARCH (Apr–June 2010) 47(2):143–157
DOI 10.1007/s12597-010-0020-3

THEORY AND METHODOLOGY

An experimental evaluation of heuristic algorithms for bus-depot matching problem of urban road transport systems

M. Mathirajan · C. V. Hariharakrishnan · V. Ramachandran

Accepted: 22 November 2009 / Published online: 15 October 2010

Abstract This research is motivated by a bus-depot matching problem observed in Urban Road Transport Systems (URTS). In URTS, buses are parked overnight at depots. Starting points of routes are usually different from depot locations. A bus has to cover the distance from its depot to the starting point of its route before being engaged on regular service. Likewise, buses usually do not provide service to the depot at the end of the service period. The distance travelled by a bus in a day from a depot to a starting terminus and/or from the ending or last terminus back to the depot without carrying passengers is known as 'dead kilometers'. The dead kilometers can be reduced by efficiently allocating the buses to depots. In the literature this problem is solved using mathematical model and heuristic algorithms. However, there is no detailed computational analysis to highlight the merits and demerits of various solution methodologies, so far addressed in the literature. In this study a set of heuristic algorithms are considered to make an efficient decisions in buses-depots matching problem. A computational experiment is carried out to understand the efficiency of the heuristic algorithms considered in this study for various large size problems in comparison with exact solutions. From the computational analysis, two out of the five heuristic algorithms considered in this study, resulted very close to exact solutions in most of the problem instances. All the heuristic algorithms considered in this study takes very meager computational time in Pentium IV for the large size problem of 30 depots and 5,031 buses considered in this study.

Keywords Bus-depot matching · Urban road transport systems · Dead kilometers · Heuristic algorithm · Experimental analysis

M. Mathirajan (✉) · V. Ramachandran
Department of Management Studies, Anna University Tiruchirappalli, Tiruchirappalli 620 024, India
e-mail: msdmathi@mgmt.iisc.ernet.inb
e-mail: drmuthu.mathirajan@gmail.com

C. V. Hariharakrishnan
Department of Mechanical Engineering, National Instituite of Technology Karnataka, Surathkal, Surathkal 575 025, India

1 Introduction

In most Indian cities, the Road Transport Services act as the backbone of Urban Road Transport System (URTS). These are generally managed by Road Transport Corporations or Undertakings controlled by the Government. Each such organization is provided with hundreds of buses, thousands of crew and personnel, and characterized by passenger demand arising at different points of time in several directions. Associated with these are the decision problems related to rationalization of routes, scheduling of vehicles, crew scheduling, allocation of vehicles to routes and such other problems. Fortunately, decisions in some of these areas can be made efficient using Operations Research Methods. The availability of Computers has hastened the development of better solutions to many complex and computationally intractable decision problems. In this paper, we describe one such area of decision making: Allocation of buses to depots without changing the existing route and schedule of each bus, where simple heuristic methods along with today's power of the desktop computer can be effectively used for substantial savings of computational time and effort with virtually no investment costs.

It is well known that each one of the buses in any Transport System is maintained and serviced by one of several depots. Due to the location of depots and terminus at different places it is essential that the buses traverse some 'Dead Kilometers' everyday. That is, every day, each bus starts from a particular depot and reaches the starting terminus of the trips it supposed to make in the whole day without carrying passengers. A 'trip' indicates a specified journey made by a bus with the intention of carrying passengers and collecting revenue for that purpose. Similarly in the end of the day, every bus returns back to the same depot (from where it started in the morning) from the last terminus of the trip without carrying passengers. The total distance covered, without carrying passengers, in this process is known as 'dead kilometers'.

Furthermore, depending on how the buses are allocated in depots, the dead kilometers for a bus can be in the 'forward' direction only (that is, from the depot to the starting terminus of the trip) or, in the 'backward' direction only (that is, from the ending terminus of the last trip to the depot) or, in both directions. For a bus, if the first trip originates and the last trip terminates at the same terminus, the dead kilometers in both directions would be the same and the total dead kilometers for the bus in the day is two times the distance between the depot and the terminus. If they are at different termini, the distances covered in each direction would be different and the overall dead kilometers for the bus are the cumulative sum of the two distances.

This study does not include other categories of trips made without passengers such as those covered between two termini to meet peak hour demands or changed route schedules or those providing intermediary linkage to chartered trips in a day. As these categories of dead kilometers are unavoidable and are independent of the depot allocations, they are termed as 'Operational Dead Kilometers' and are kept outside the purview of the problem in this study. Furthermore, this study does not attempt to change the existing route, schedule of each bus (that is, this study is not related to routing and scheduling of each bus) and location of depots instead efficiently matching the buses to depots with an objective of minimizing the overall dead kilometers.

For many major URTS in India, this 'dead kilometers' is of the order of several thousand kilometers each day. Each dead kilometer covered by a bus implies a loss to the URTS as it is not only non-revenue yielding, but results in higher fuel consumption and increased operational costs. Particularly for many major URTS in India, the annual implicit loss in terms of fuel, maintenance and other costs due to such 'dead kilometers' runs to several lakhs of Rupees.

It is obvious that dead kilometer cannot be totally eliminated as it is not possible to have all the starting or ending termini in front of the depots, nor it is possible to create as many depots as the number of termini. Furthermore, the dead kilometer depends on the location of the depots and the distances of the starting and ending termini of various bus routes. While some dead kilometers are inevitable for any organization, the question is, for any URTS what could be the minimum possible dead kilometers? And how to allocate each of the buses to one of the available depots, which minimizes the over all dead kilometers? This decision problem can be analyzed using a rational and scientific approach.

In the following section, we review previous closely related work on bus-depot matching problem without changing the existing route and schedule of each bus. Section 3 presents a list of heuristic algorithms considered in this study for the proposed computational analysis. We then present in section 4 the computational experiments carried out to compare the performance of the heuristic algorithms with the exact solution. Finally, we conclude the paper in section 5.

2 Previous closely related work

The dead kilometers problem was formulated with two objectives: minimizing the cumulative distance traveled by all buses from the depots to the starting points of their routes (primary objective) and minimizing the maximum distance traveled by individual buses from the depots to the starting points of their respective route (secondary objective) problem by Prakash et al. [5] and Sharma and Prakash [7]. This formulation has (a) an implementation difficulty in deciding which bus should get allocated to which depot, (b) the implicit assumption that every bus starts and ends in the same terminus and this is not true in practice, and (c) been demonstrated with their solution process on a tiny instance without any computational experiments to guarantee to obtain exact solution (and or non-dominated solution) for real life size problems, which would have two digits number of depots and four digits number of buses in any URTS in India.

Van der Perree and Van Oudheusden [10] proposed a mixed integer programming model for optimizing both number of maintenance facilities (i.e., depots) and the overall dead kilometers. Due to the computational difficulties, they proposed hierarchical optimization in which first the researcher proposed the formulation to solve the problem of minimizing the number of maintenance facilities and subsequently proposed the formulation to solve the problem of optimally allocating the buses to maintenance facilities while minimizing the overall dead kilometers. The proposed models are demonstrated empirically using the real life data collected from Bangkok Mass Transit Authority. As the process of hierarchical optimization have two different models for each of the specific decision problems, there can be several solutions for the integrated problem on "Minimizing number of depots and the overall dead kilometers by optimally allocating the number of buses to the depots".

A model for the allocation of buses to depots with a single objective of minimizing the overall dead kilometers was modeled as a Transportation problem [8,11] and as a (0-1) integer linear programming problem [6]. Though all the model ends with the same total optimal dead kilometers, the model (please see Appendix 1) given by Raghavendra and Mathirajan [6] provides the tactical details on which bus gets allocated to which depot and obtaining this implementation detail is not possible in the other models proposed by [8,11]. This tactical detail is very essential for physical implementation in assigning the buses to depots optimally as per the optimal solution obtained from the model.

The mathematical models proposed in the literature for the bus-depot matching problem has only two constraints namely (i) capacity of the depots, and (ii) each bus should get allocated to only one depot [6]; or (i) capacity of the depots, and (ii) the number of buses required at the starting terminus [8,11]. Based on the nature of the constraints involved in both mathematical models, the coefficient matrix of these models follow the unimodular property [4] and it is possible to solve any large scale bus-depot matching problem as an LP (Linear Programming) problem and obtain (0-1) integer solution. A mathematical programming model has many other aspects besides the coefficient matrix, although it is central. Finally, there is a possibility that these models do not follow the unimodular property when we have to incorporate some additional real life constraints such as conditional constraint, co-requisite constraint, r out of n routes/buses should be allocated to a specific depot (r<n), etc., that might be required for certain buses and their allocation to certain depots.

When the mathematical model for the buses-depots matching problem failing to have the unimodular property we need to solve the problem using (0-1) integer linear programming methods to obtain 0–1 solution. With this the decision making problem becomes computationally intractable one, as the number of 0-1 variables in the real-life problem runs to several thousands.

Due to computational difficulties in obtaining exact solution a few researchers proposed heuristic algorithms for efficient allocation of buses to depots [2,12]. Heuristic algorithms proposed in the earlier studies were demonstrated on a particular case study data only and they did not carry out any extensive computational experiments to evaluate the consistency of quality of the solution. This has motivated us to conduct an appropriate research with extensive experimental analysis to conclude about the performance of the earlier heuristic algorithms in addition to the standard heuristic for the transportation problem: Vogel Approximation method (VAM) in efficiently allocating buses to depots. In addition, to the substantiation of a particular heuristic the experimental analysis enable the researchers (i) to evaluate the several alternate procedures in an attempt to confirm the relative standings of the heuristics which give near optimal solutions and (ii) to provide greater confidence in interpreting their efficiency.

3 Heuristic algorithms for bus-depot matching problem

Mathirajan and Meenakshi [3] empirically indicated that the heuristic algorithm VAM applying on non-dimensional[1] matrix (called as Total Opportunity Matrix in that paper)

[1] Non-dimensionalizing, also known as scaling or normalizing, should be a preliminary step to developing any model [1, 9]. The process of non-dimensionalizing (a) simplify the equations by reducing the number of variables, (b) analyze the behavior of the system, regardless of the units used to measure the variables (examples: aspects ration, and Reynolds number), and (c) rescale the parameters and variables so that all computed quantities are of relatively similar magnitudes.

instead of applying to original transportation cost matrix leads to very near optimal solution. This has motivated us, in this paper, to try out the heuristic algorithm: VAM applying to two types of non-dimensional matrices instead of applying to original dead kilometers matrix, as bus-depot matching problem can be viewed as transportation problem.

A numerical example for bus-depot matching problem with three depots and 12 buses system, presented in Appendix 2, is shown as a transportation problem in Table 1. We have also considered the earlier ranking algorithm [12] for efficient allocation of buses to depots, as this is some what close to the real life practice method, applying to original dead kilometers matrix as well as to two types of non-dimensional matrices.

Accordingly, we have considered five heuristic algorithms: Ranking Algorithm (RA) applied on original dead kilometers (DK) matrix, called as RA-DK; Ranking algorithm applied on first type non-dimensional matrix 1 [called as Total Opportunity Dead Kilometers Matrix (TODK)], called as RA-TODK; Ranking Algorithm applied on second type non-dimensional matrix [called as Ratio Opportunity Dead Kilometers], called as RA-RODK; Vogel Approximation Method (VAM) applied on TODK, called as VAM-TODK; and VAM applied on RODK, called as VAM-RODK in the proposed computational analysis to observe their performances in comparison with exact solution. For facilitating the readers, the pseudo code for these algorithms are given here.

Ranking Algorithm applying on to original Dead Kilometers (RA-DK)

Step 1: Route-wise (Schedule-wise), calculate the “dead kilometers”: $\mathrm{DK}(\mathrm{s},\mathrm{d}) = [\mathrm{FDK}(\mathrm{s},\mathrm{d}) + \mathrm{BDK}(\mathrm{s},\mathrm{d})]$ for all s=1, 2, …N—Schedules or Routes; d=1, 2 ...M-Depots; FDK(s,d)—Forward dead kilometer; and BDK(s,d)—Backward dead kilometer.

Table 1 The Transportation tableau for the numerical example given in Appendix 2

Bus/Schedule/Route	Dead Kilometers due to the allocation of Bus to Depot			Bus-Allocation (Supply Quantity)
	D1	D2	D3	
B1	18.2	19.0	18.5	1
B2	10.0	14.0	8.0	1
B3	22.2	14.8	17.3	1
B4	14.4	9.4	16.6	1
B5	14.4	9.4	16.6	1
B6	18.8	8.0	15.2	1
B7	18.2	19.0	18.5	1
B8	15.2	16.0	15.0	1
B9	20.0	15.8	13.8	1
B10	24.0	11.6	12.6	1
B11	13.0	17.0	11.5	1
B12	15.2	14.4	20.0	1
Depot Capacity (Demand Quantity)	3	5	4	

Step 2: Assign the Rank R=1 or 2 or …M for each depot depending on the value of DK(s,d) and. Repeat this for all schedules.
Step 3: Set R=1
Step 4: Set d=1
Step 5: Depot-wise, find all schedules whose rank=R and also find the total number of such schedules, say "SUM(d)".
Step 6: Compare "SUM(d)" with the capacity "CAPACITY(d)" of depot "d", that is,

If (SUM(d) < CAPACITY(d)), THEN

- Assign all the "SUM(d)" number of schedules to depot "d".
- CAPACITY(d) = CAPACITY(d)–SUM(d)

Else

- Find the first "CAPACITY(d)" schedules with minimum DK(s,d)
- Assign these "CAPACITY(d)" number of schedules to depot "d"
- CAPACITY(d)=0

End if

Step 7: Set d = d + 1; and Repeat Step 5 and Step 6 until d=M
Step 8: Set R = R + 1; and Repeat Step 4 to Step 7, until R=N

Since the other heuristic algorithm: VAM considered in this study is familiar to every one, it is not described here and only the procedure on obtaining the non-dimensional matrices: TODK and RODK are given below:

Procedure to obtain the Total Opportunity Dead Kilometers (TODK) matrix

Step 1: Route-wise (Schedule-wise), calculate the "dead kilometers": $\mathrm{DK}(\mathrm{s},\mathrm{d}) = [\mathrm{FDK}(\mathrm{s},\mathrm{d}) + \mathrm{BDK}(\mathrm{s},\mathrm{d})]$ for all s=1, 2, …N—Schedules and d=1, 2 … M-Depots
Step 2: Compute the Total Opportunity Dead Kilometers (TODK) matrix by summing both Depot Opportunity Dead Kilometers (DODK) matrix and Schedule Opportunity Dead Kilometers (SODK) matrix and these are computed as follows:

- Computing Depot Opportunity Dead Kilometers (DODK) matrix

  For all s=1, 2, …. N-Schedule do the following
  For all d=1, 2, …. M (Depot) do the following

  DODK(s,d) = DK(s,d) – Min {DK(s,d), d=1,2, ….M}
- Computing Schedule Opportunity Dead Kilometers (SODK) matrix

  For all d=1, 2, …. M-Depot do the following
  For all s=1, 2, …. N-Schedule do the following

  SODK(s,d) = DK(s,d) – Min {DK(s,d), s=1,2, ….N}
- Computing Total Opportunity Dead Kilometers (TODK) matrix

  TODK(s,d) = DODK(s,d) + SODK(s,d)

  For all s=1, 2, …. N- Schedule and for all d=1, 2, …. M-Depot

Procedure to obtain the Ratio Opportunity Dead Kilometers (RODK) matrix

Step 1: Route-wise (Schedule-wise), calculate the "dead kilometers": $\mathrm{DK}(\mathrm{s},\mathrm{d}) = [\mathrm{FDK}(\mathrm{s},\mathrm{d}) + \mathrm{BDK}(\mathrm{s},\mathrm{d})]$ for all s=1, 2, …N- Schedules and d=1, 2 … M-Depots

Step 2: Compute the Ratio Opportunity Dead Kilometers (RODK) matrix by dividing each cell of the dead kilometers matrix by the lowest dead kilometers in the entire dead kilometers matrix and these are computed as follows:

- Find the lowest dead kilometers value in the entire dead kilometers matrix and lit it be Min-DK.
- Computing Ratio Opportunity Dead Kilometers (RODK) matrix

RODK(s,d) = DK(s,d)/Min-DK
For all s=1, 2, …. N- Schedule and for all d=1, 2, …. M-Depot

All the five heuristic algorithms considered in this study were implemented in programming language Turbo C++ and run on Pentium IV 2.40 Ghz computer with 512 RAM.

4 Computational experiments

A computational experiment is appropriate in order to provide a perspective on the relative effectiveness of any proposed heuristic algorithm. An experimental approach of this type relies on two elements; an experimental design and the measure of effectiveness. These are discussed first as a prelude to the discussion on the evaluation of heuristics.

4.1 Experimental design

In this study, computational experiment is carried out with an objective to evaluate the absolute quality of the solutions obtained by the heuristic algorithms by comparing them with optimal solution. In order to generate meaningful test instances an experimental design is developed based on the observation made in Bangalore Metropolitan Transport Corporation (BMTC), Bangalore, India. Accordingly, we identified three important problem parameters based on our observation made in BMTC. They are number of depots (**ND**), Depot-Capacity (**DC**), and Dead Kilometers (**DK**). The other parameters, number of buses will be computed once we generate depot capacity for each depot of a particular problem configuration.

The parameter, number of depots 'ND' (which indicates the problem size of the bus-depot matching problem), was assumed based on the past data on number of depots observed in BMTC, Bangalore/India. As the depot capacity varies across the depot, it is assumed that the capacities of the depots are uniformly distributed. Once it is decided the capacity of each depots, the other problem parameter on number of buses (NB) will be computed by summing the capacities of each depot. As the data on dead kilometers varies for each buses based on the allocation made, it is assumed that the data on dead kilometers for each buses are uniformly distributed. Furthermore, in this study there are three different levels of uniform distributions used for both depot capacity and dead kilometers. The uniform distribution was chosen because it is a relatively high-variance distribution, which would allow the heuristics to be tested under conditions relatively unfavorable to them.

The experimental design for generating test problems as reflected here [See Table 2 for experimental design summary] was implemented in programming language Turbo C++ and run on Pentium IV 2.40 Ghz computer with 512 RAM. Accordingly ten problem instances for each combination of values for (ND, DC, DK) were randomly generated, yielding a total of $270[= 3 \times 3 \times 3 \times 10]$ problem instances.

4.2 Measure of effectiveness

The performance of the algorithms may vary over a range of problem instances. Therefore, the performances of the proposed heuristic algorithms were compared using the standard performance measures, viz., average relative percentage deviation (ARPD), indicating the average performance of heuristics and maximum relative percentage deviation (MRPD), indicating the worst case performance of heuristics.

Let TDK_t be the total dead kilometers (TDK) given by tth Heuristic, where t=1, 2, 3, 4 and 5 refer to RA-DK, RA-TODK, RA-RODK, VAM-TODK, and VAM-RODK respectively. Let TDK_I be the minimum total dead kilometers given by (0-1) ILP model, presented in the Appendix 1. The RPD (Relative Percentage Deviation) for the tth heuristic is given by $\mathbf{RPD_t} = ((\mathbf{TDK_t} - \mathbf{TDK_I})/\mathbf{TDK_I}) \times \mathbf{100}$. For each level (ND, DC, DK), the average RPD over ten problem instances, was computed and termed ARPD of the tth heuristic solution. In addition, we also observed, for each level (ND, DC, DK), the maximum RPD (MRPD) for a given heuristic among ten problem instances.

4.3 Performance evaluation of heuristic algorithms against optimal solution

Each problem instance was run through the heuristic algorithms and (0-1) ILP model and obtained as well as recorded the respective minimum total dead kilometers. For each level of (ND, DC, DK) the value of "ARPD" (for average performance of heuristics) and "MRPD" (for worst case performance of heuristics) were computed with respect to optimal total dead kilometers, obtained from (0-1) ILP model, presented in the Appendix 1, using LINGO and these are presented in Tables 3 and 4 respectively. Furthermore, irrespective of the problem configurations, the overall ARPD and MRPD were also computed over 270 instances and the same is shown in Figs. 1 and 2.

Table 2 Summary of experimental design

Problem Factor	Number of Levels	Values
Number of Depots (ND)	3	20, 25, and 30
Depot Capacity (DC)	3	[50–150], [100–150], [100–250]
Dead Kilometers (DK)	3	[5–50], [5–500], [250–500]
Number of Problem Configurations		$= (3 \times 3 \times 3) = 27$
Number of Instances per Configuration		=10
Total Number of Problem Instances		$= 27 \times 10 = 270$

Table 3 Average performance of the heuristic algorithms—(ARPD score over ten instances)

ND	DC	NB	DK	RA-DK	RA-TODK	RA-RODK	VAM-TODK	VAM-RODK
20	[50–150]	2310	[5–50]	12.6	12.2	12.2	3.1	3.0
			[5–500]	28.8	27.1	27.0	2.7	2.6
			[250–500]	1.6	1.5	1.6	0.2	0.3
	[100–150]	2507	[5–50]	14.6	13.7	13.7	3.1	3
			[5–500]	27.5	26.1	26.1	2.5	2.9
			[250–500]	1.8	1.7	1.8	0.1	0.2
	[100–250]	3307	[5–50]	14.1	13.7	13.8	3.8	3.7
			[5–500]	35.0	33.0	33.0	5.9	6.0
			[250–500]	1.8	1.7	1.8	0.2	0.3
25	[50–150]	2499	[5–50]	15.4	15.3	15.3	4.2	4.2
			[5–500]	35.4	34.2	34.2	4.2	3.9
			[250–500]	1.7	1.6	1.7	0.2	0.3
	[100–150]	3099	[50–150]	12.8	12.5	12.5	3.4	3.3
			[5–500]	28.0	27.1	27.1	2.2	2.6
			[250–500]	1.3	1.3	1.5	0.1	0.2
	[100–250]	4149	[5–50]	16.0	15.5	15.5	4.7	4.7
			[5–500]	37.6	36.2	36.2	3.9	3.7
			[250–500]	1.8	1.8	1.9	0.2	0.3
30	[50–150]	2981	[5–50]	15.6	15.2	15.2	5.1	4.7
			[5–500]	37.5	36.3	36.3	3.8	3.6
			[250–500]	1.6	1.6	1.7	0.2	0.3
	[100–150]	3731	[5–50]	11.5	11.0	11.0	3.4	3.5
			[5–500]	28	27.1	27.1	3.4	3.0
			[250–500]	1.2	1.2	1.3	0.1	0.2
	[100–250]	5031	[5–50]	14.3	13.7	13.7	3.8	3.8
			[5–500]	35.0	33.7	33.7	3.6	3.6
			[250–500]	1.5	1.5	1.5	0.2	0.3

The analysis based on the average performance of the heuristic algorithms (Table 3) and worst case performance of the heuristic algorithms (Table 4) indicate that in almost every problem configuration the solution method: VAM-TODK and VAM-RODK out perform the others. Furthermore, it is to be highlighted that these two methods resulted very near optimal solutions in a majority of the cases (Tables 3 & 4 and Figs. 1 & 2).

From the average case analysis and worst case analysis it is observed that the heuristic algorithm: Vogel's Approximation Method (VAM) yield very close to optimal solution when it is applied on both non-dimensional matrices in comparison with raking algorithm (somewhat close to the real life practice method) applied on both original dead kilometer and non-dimensional matrices.

One of the immediate limitations of the present study is that we could have used the best known heuristics available for (0-1) ILP problem and compared with the set of heuristic algorithms used in this study, as bus-depot matching problem can be formulated as (0-1) ILP problem. However, we have focused the problem studied

Table 4 Worst case performance of the heuristic algorithms—(MRPD score over ten instances)

ND	DC	NB	DK	RA-DK	RA-TODK	RA-RODK	VAM-TODK	VAM-RODK
20	[50–150]	2310	[5–50]	17.3	17.1	17.1	4.2	5.1
			[5–500]	34.7	33.1	33.1	6.0	7.3
			[250–500]	2.0	1.8	1.9	0.5	0.7
	[100–150]	2507	[5–50]	16.8	15.0	15.0	4.3	3.6
			[5–500]	32.3	32.2	32.2	6.0	5.1
			[250–500]	2.1	2.0	2.2	0.3	0.3
	[100–250]	3307	[5–50]	15.0	14.9	14.9	5.3	5.7
			[5–500]	39.2	37.9	37.9	9.8	11.2
			[250–500]	2.1	2.0	2.0	0.4	0.5
25	[50–150]	2499	[5–50]	16.7	17.3	17.3	5.1	5.2
			[5–500]	40.6	39.6	39.6	6.1	5.5
			[250–500]	1.8	1.8	2.0	0.2	0.5
	[100–150]	3099	[50–150]	14.4	13.7	13.7	4.9	5.0
			[5–500]	38.5	35.6	35.6	4.2	4.8
			[250–500]	1.6	1.6	1.8	0.1	0.2
	[100–250]	4149	[5–50]	17.7	17.2	17.2	5.4	5.8
			[5–500]	42.0	40.9	40.9	6.2	5.1
			[250–500]	2.0	2.0	2.0	0.2	0.4
30	[50–150]	2981	[5–50]	17.1	17	17.0	6.7	6.2
			[5–500]	42.2	41.3	41.3	5.4	5.5
			[250–500]	1.7	1.8	2.0	0.3	0.5
	[100–150]	3731	[5–50]	13.6	13	13	4.0	4.3
			[5–500]	32.8	29.8	29.8	5.1	4.0
			[250–500]	1.7	1.4	1.8	0.2	0.8
	[100–250]	5031	[5–50]	16.3	15.2	15.2	4.2	4.1
			[5–500]	40.3	38.4	38.4	5.8	4.9
			[250–500]	1.6	1.6	1.6	0.2	0.4

here as transportation problem, the best know heuristics for (0-1) ILP are not considered as potential candidate for bus-depot matching problem.

5 Conclusions

This paper addresses how various simple heuristic algorithms perform while solving large scale (such as 30 depots and 5,031 buses as in the experimental design) bus-depot matching problem of any URTS in comparison with optimal procedure. From the computational analysis, it is observed that all the heuristic algorithms considered in this study takes very meager computational time in Pentium IV for the large size bus-depot matching problem, having 30 depots and 5,031 buses.

The computational analysis carried out in this paper highlighted that the heuristic algorithm: Vogel's Approximation Method (VAM) has very high probability to result

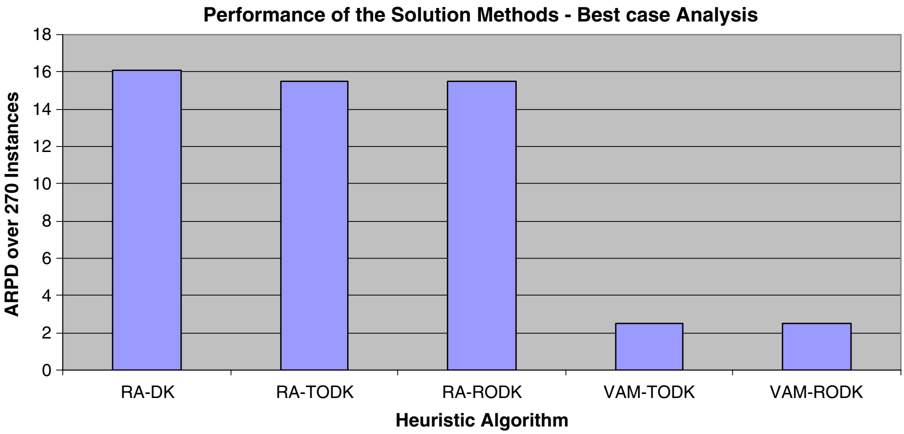

Fig. 1 Performance of the solution methods—average case analysis

near optimal solution for bus-depot matching problem when it is applied on both cases of non-dimensional matrices, which are generated from the given dead kilometers matrix.

Furthermore, it is important to observe here that in the application of each of the heuristic algorithms considered in this study, no physical change of buses from any depot to any other is envisaged. In terms of implementation what is involved is just a change in the servicing of some of the routes by some of the depots. It is also important to observe here that these heuristic algorithms do not call for any new investment in physical assets; hence considerable savings are possible through a rational approach to decision in an area which is considered relatively insignificant in many URTS.

As bus-depot matching problem can be formulated as (0-1) ILP problem, an immediate research direction is to identify the best known heuristics available in the literature for (0-1) ILP problem and compare the quality of these heuristics with the set of heuristic algorithms considered in this study.

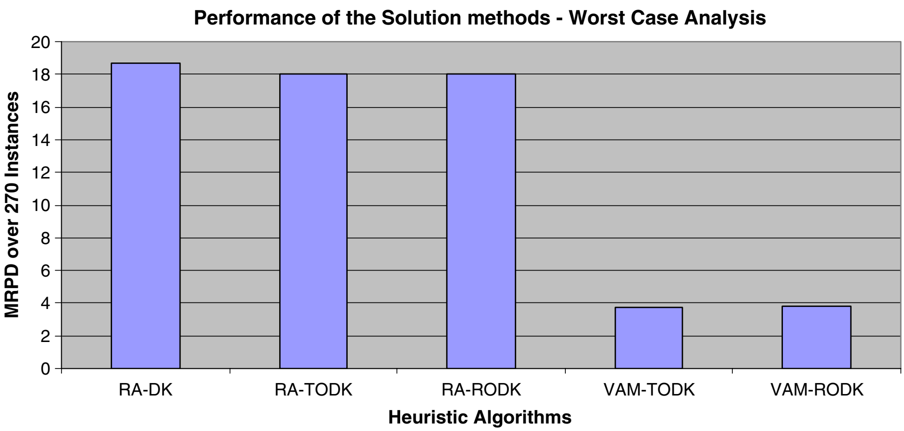

Fig. 2 Performance of the solution methods—worst case analysis

Acknowledgements The authors gratefully acknowledge the valuable comments and remarks of two anonymous referees that improved the quality of the paper.

APPENDIX 1: Mathematical programming models for bus-depot matching problem

A (0-1) Integer Linear Programming Model [Source: [6]]

The decision variables The decision variables are defined as follows:

$$X_{ds} = \begin{cases} 1 & \text{if the bus schedule } s \text{ is allocated to depot } d \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where d=1,2,..., M refers to the depots and s=1,2,..., N to the buses or schedules.

The objective function The objective considered is the minimization of the overall dead kilometers per day for the organization. This is given by

$$\text{Minimize} \quad Z = \sum_{d=1}^{M}\sum_{s=1}^{N}\sum_{t=1}^{2} D_{dt}.X_{ds}, \quad (2)$$

where D_{dt} is the distance between depot d and terminus t and the summation over t extends to the two termini, namely, the starting and ending termini of the respective schedules. Alternatively, it can be expressed by

$$\text{Minimize} \quad Z = \sum_{d=1}^{M}\sum_{s=1}^{N} (f_{ds} + b_{ds})X_{ds}, \quad (3)$$

where f_{ds} is the dead kilometers in the forward direction if schedule s is allocated to depot d and b_{ds} is that in the backward direction.

The constraints There are two basic constraints in this model, one pertaining to the capacity of the depots and the other related to the allocation of schedules.

The *depot capacity constraint* is to ensure that the number of buses allocated to a depot is within the total service capacity of the depot in terms of the maximum number of buses it can service and maintain. This is expressed by

$$\sum_{s=1}^{N} X_{ds}C_d \quad \text{for all } d = 1, 2, ...M \quad (4)$$

where C_d is the capacity of depot d.

The fact that *each bus should get allocated to exactly one of the depots* is expressed by the constraints of the form

$$\sum_{d=1}^{M} X_{ds} = 1, \quad \text{for all } s = 1, 2, ...N. \quad (5)$$

For some buses, there may exist specific bus-depot assignment requirements. For example, assume that the double-decker buses cannot be allocated to some depots due to the low-level railway under-bridges prohibiting their movement and the policy is to allocate these buses only to Depot-6. Such *specific bus-depot matching* can be enforced through the following additional constraints:

$$X_{Ir} = 1 \text{ and } X_{dr} = 0, \tag{6}$$

For all r {A} and d≠I where I is the specific depot and {A} is the index set of all double decker buses.

The model described by (1–6) is a 0-1 Integer Linear Programming (ILP) problem. It may be observed that the matrix of coefficients formed by the constraints follow the unimodularity property [4]. The model can therefore be solved using LP approach and get integer solutions.

APPENDIX 2: A numerical example for bus-depot matching problem

The numerical problem presented here has three depots and 12 buses. The details on the current allocation of these 12 buses to the existing three depots are given in Table 5 of this Appendix. Using Table 5, the current details on depot wise the existing dead kilometers (dead kilometers for each of the schedule is computed as per the definition of dead kilometers as the sum of the distance from staring terminus to depot and ending terminus to depot) is computed and presented in Table 6 of this Appendix. By scanning both starting and ending terminus of the given 12 buses, the distinct starting and/or ending terminus are observed and the same is given in Table 7 of this Appendix along with existing distances between these terminus from each of the depot. Using Table 5 and Table 7 along with the definition of the dead kilometers, the schedule wise (bus wise) dead kilometers is computed for each of the depot, in case that particular schedule (bus) is allocated to that depot, is shown in Table 8 of this Appendix. The Table 8 is presented in the form of Transportation Tableau in Table 1 for the numerical example with three depots and 12 buses.

Table 5 Schedule details (existing)

Sl. No.	Depot	Bus/Route No.	Starting terminus (ST)	Distanc From ST to Depot	Ending terminus (ET)	Distance From EP to Depot
1	D1	B1	SBS	8.0	BBS	10.2
2		B2	CMT	5.0	CMT	5.0
3		B3	MBS	12.0	BBS	10.2
1	D2	B4	GNR	4.0	SNR	5.4
2		B5	SNR	5.4	GNR	4.0
3		B6	GNR	4.0	GNR	4.0
4		B7	BBS	9.0	SBS	10.0
5		B8	CMT	7.0	BBS	9.0
1	D3	B9	MBS	6.3	SBS	7.5
2		B10	MBS	6.3	MBS	6.3
3		B11	CMT	4.0	SBS	7.5
4		B12	SNR	9.0	BBS	11.0

Table 6 Existing dead kilometers, depot wise and overall

Input Parameters	Details on the "Input Parameters" for the Depot		
	D1	D2	D3
Capacity	3	5	4
Existing Routes	B1, B2, B3	B4, B5, B6, B7, B8	B9, B10, B11, B12
Existing Dead KM	50.4	61.8	57.9
Existing Total Dead Kilometers			170.1

Table 7 Terminus-depot: distance matrix

Terminus Code (Terminus Name)	Distance from 'Terminus' to the Depot		
	D1	D2	D3
Bangalore Bus Station (BBS)	10.2	9.0	11.0
City Market (CMT)	5.0	7.0	4.0
Gandhi Nagar (GNR)	9.4	4.0	7.6
Malleswaram Bus Station (MBS)	12.0	5.8	6.3
Shivajinagar Bus Station (SBS)	8.0	10.0	7.5
Srinagar (SNR)	5.0	5.4	9.0

Table 8 Dead kilometers due to bus-depot matching

Bus/Schedule/Route	Dead Kilometers due to bus to depot matching for its allocation		
	D1	D2	D3
B1	$8.0 + 10.2 = 18.2$	$9.0 + 10.0 = 19.0$	$11.0 + 7.5 = 18.5$
B2	$5.0 + 5.0 = 10.0$	$7.0 + 7.0 = 14.0$	$4.0 + 4.0 = 8.0$
B3	$12.0 + 10.2 = 22.2$	$5.8 + 9.0 = 14.8$	$6.3 + 11.0 = 17.3$
B4	$9.4 + 5.0 = 14.4$	$4.0 + 5.4 = 9.4$	$7.6 + 9.0 = 16.6$
B5	$9.4 + 5.0 = 14.4$	$4.0 + 5.4 = 9.4$	$7.6 + 9.0 = 16.6$
B6	$9.4 + 9.4 = 18.8$	$4.0 + 4.0 = 8.0$	$7.6 + 7.6 = 15.2$
B7	$8.0 + 10.2 = 18.2$	$9.0 + 10.0 = 19.0$	$11.0 + 7.5 = 18.5$
B8	$5.0 + 10.2 = 15.2$	$7.0 + 9.0 = 16.0$	$4.0 + 11.0 = 15.0$
B9	$12.0 + 8.0 = 20.0$	$5.8 + 10.0 = 15.8$	$6.3 + 7.5 = 13.8$
B10	$12.0 + 12.0 = 24.0$	$5.8 + 5.8 = 11.6$	$6.3 + 6.3 = 12.6$
B11	$5.0 + 8.0 = 13.0$	$7.0 + 10.0 = 17.0$	$4.0 + 7.5 = 11.5$
B12	$5.0 + 10.2 = 15.2$	$5.4 + 9.0 = 14.4$	$9.0 + 11.0 = 20.0$

References

1. Carter, W.C.: Appendix: Non-dimensionalizing (Scaling or Normalizing). http://pruffle.mit.edu/3.016/Appendices/NonDimensional.pdf (accessed on 11th December 2009). (2009)
2. Mathirajan, M.: An efficient heuristic for minimizing dead kilometers in Urban Transport Systems. J Transp Manag **17**, 125–133 (1993)
3. Mathirajan, M., Meenakshi, B.: Experimental analysis of some variants of Vogel's Approximation Method. Asia Pac J Oper Res **21**(4), 447–462 (2004)
4. Murthy, K.G.: Linear Programming. Wiley, New York (1983)
5. Prakash, S., Balaji, B.V., Tuteja, D.: Optimizing dead kilometers in urban bus routes through a non dominated solution approach. Eur J Oper Res **114**, 465–473 (1999)
6. Raghavendra, B.G., Mathirajan, M.: Optimal allocation of buses to depots: A case study. OPSEARCH **24**(4), 228–239 (1987)
7. Sharma, V., Prakash, S.: Optimizing dead kilometers in urban bus routes. J Transp Eng ASCE **112**(I), 121–129 (1986)
8. Sridharan, R.: Allocation of buses to depots: A case study. Vikalpa, The Journal for Decision Makers **16**(2), 27–32 (1991)
9. Temponi, C., Bryant, M.D., Fernandez, B.: Integrating of business function models into an aggregate enterprise systems model. Eur J Oper Res **199**(3), 793–800 (2009)
10. Van der Perre, P.P.G., Van Oudheusden, D.D.L.: Reducing depot-related costs of large bus operators a case study in Bangkok. Eur J Oper Res **96**, 45–53 (1996)
11. Vasudevan, J., Malini, E., Victor, D.J.: Fuel savings in bus transit using depot-terminal bus allocation model. J Transp Manag **17**, 409–416 (1993)
12. Vidya Shankar, T.R., Raghavendra, B.G., Mathirajan, M. (1989). A heuristic approach to minimize dead kilometers in urban transport systems. Chapter 6. In: Tripathy, A., Ahmedabad, I.I.M. (eds.) Modelling and Analysis of Large Systems