Asia-Pacific Journal of Operational Research
Vol. 21, No. 4 (2004) 447–462

EXPERIMENTAL ANALYSIS OF SOME VARIANTS OF VOGEL'S APPROXIMATION METHOD

M. MATHIRAJAN* and B. MEENAKSHI

Department of Management Studies, Indian Institute of Science
Bangalore 560 012, India
**msdmathi@mgmt.iisc.ernet.in*

Received April 2003
Revised October 2003

This paper presents a variant of Vogel's approximation method (VAM) for transportation problems. The importance of determining efficient solutions for large sized transportation problems is borne out by many practical problems in industries, the military, etc. With this motivation, a few variants of VAM incorporating the total opportunity cost (TOC) concept were investigated to obtain fast and efficient solutions. Computational experiments were carried out to evaluate these variants of VAM. The quality of solutions indicates that the basic version of the VAM coupled with total opportunity cost (called the VAM–TOC) yields a very efficient initial solution. In these experiments, on an average, about 20% of the time the VAM–TOC approach yielded the optimal solution and about 80% of the time it yielded a solution very close to optimal (0.5% loss of optimality). The CPU time required for the problem instances tested was very small (on an average, less than 10 s on a 200 MHz Pentium machine with 64 MB RAM).

Keywords: Transportation problem; heuristic; Vogel's approximation method; total opportunity cost; computational experiments.

1. Introduction

The transportation problem constitutes an important part of logistics management. In addition, logistics problems without shipment of commodities may be formulated as transportation problems. For instance, the decision problem of minimizing dead kilometers (Raghavendra and Mathirajan, 1987) can be formulated as a transportation problem (Vasudevan *et al.*, 1993; Sridharan, 1991). The problem is important in urban transport undertakings, as dead kilometers mean additional losses. It is also possible to approximate certain additional linear programming problems by using a transportation formulation (e.g., see Dhose and Morrison, 1996).

Various methods are available to solve the transportation problem to obtain an optimal solution. Typical/well-known transportation methods include the stepping stone method (Charnes and Cooper, 1954), the modified distribution method (Dantzig, 1963), the modified stepping-stone method (Shih, 1987), the simplex-type algorithm (Arsham and Kahn, 1989) and the dual-matrix approach (Ji and Chu, 2002). Glover *et al.* (1974) presented a detailed computational comparison of basic solution algorithms for solving the transportation problems. Shafaat and Goyal (1988) proposed a systematic approach for handling the situation of degeneracy encountered in the stepping stone method. Since our intention in this paper is not to attempt to obtain an optimal solution of the transportation problem, a detailed literature review on the basic solution methods is not presented.

All the optimal solution algorithms for solving transportation problems need an initial basic feasible solution to obtain the optimal solution. There are various heuristic methods available to get an initial basic feasible solution, such as "North West Corner" rule, "Best Cell Method," "VAM — Vogel's Approximation Method" (Reinfeld and Vogel, 1958), Shimshak *et al.*'s version of VAM (Shimshak *et al.*, 1981), Goyal's version of VAM (Goyal, 1984), Ramakrishnan's version of VAM (Ramakrishnan, 1988) etc. Further, Kirca and Satir (1990) developed a heuristic, called TOM (Total Opportunity-cost Method), for obtaining an initial basic feasible solution for the transportation problem. Gass (1990) detailed the practical issues for solving transportation problems and offered comments on various aspects of transportation problem methodologies along with discussions on the computational results, by the respective researchers. Recently, Sharma and Sharma (2000) proposed a new heuristic approach for getting good starting solutions for dual based approaches used for solving transportation problems. In this paper, the basic idea of Kirca and Satir (1990) is extended using the VAM procedure.

The paper is organized as follows: In Section 2, Kirca and Satir's heuristic (TOM) and the variants of VAM are briefly discussed. A computational experiment carried out to evaluate the variants of VAM and its results are described in Section 3. Important observations are summarized in the last section.

2. Variants of Vogel's Approximation Method

Since the basic idea of TOM is extended along with the VAM procedure, TOM is first briefly discussed here. The TOM is an effective application of the "best cell method" along with some tie-breaking features on the total opportunity cost (TOC) matrix. The TOC matrix is obtained by adding the "row opportunity cost matrix" (row opportunity cost matrix: for each row, the smallest cost of that row is subtracted from each element of the same row) and the "column opportunity cost matrix" (column opportunity cost matrix: for each column of the original transportation cost matrix the smallest cost of that column is subtracted from each element of the same column). When we apply the "best cell method" (least cost method) on the TOC matrix, if more than one "TOC cell" is competing for allocation, tie-breakers are used in the following sequence:

1. Make the allocation to *the cell* with the smallest cost.
2. In the case of a tie in (1), make allocation to *the cell* with the largest possible allocation.
3. In the case of a tie in (2), make allocation to *that cell* with first occurrence.

Kirca and Satir (1990) claimed that TOM performed as well as, or better than, VAM for 372 out of 480 problem instances tried out by them. Further, they claimed that TOM provided optimal solutions to 134 out of 480 problems and VAM did not result in an optimal solution even once.

As VAM usually yields a better initial solution than the other initial basic feasible solution methods, the reasoning behind the TOM and the note of Goyal (1991) motivated us to couple VAM principles and the basis of the TOM to derive two variants of VAM. These variants are (1) VAM, applied on the TOC matrix [VAM–TOC] and (2) VAM with tie-breakers applied on the TOC matrix [VAMT–TOC]. Further, the basic version of VAM and VAM with tie-breakers applied on the original transportation cost matrix (VAM–TC and VAMT–TC, respectively) is also included in the computational analysis mainly to study the effect of tie-breakers introduced by Kirca and Satir (1990). Thus, the VAM and three variants of VAM are considered in this study along with TOM.

The algorithms VAM–TOC and VAMT–TOC follow the steps of Kirca and Satir (1990) with the exception of allocations, which are done using VAM instead of the "best cell method" in the TOC matrix. Further, systematic details of these algorithms are given in Appendix A. The next section deals with the computational experiments, and the variants of VAM are proposed.

3. Computational Experiments

For evaluating the performance of the VAM and its variants and TOM, computational experiments were carried out. The experiments and the analysis of the experimental data are presented in this section. The main goal of the experiment was to evaluate the quality of the solutions obtained by VAM and its variants and TOM by comparing them with optimal solutions. An experimental approach of this sort relies on two elements: a measure of effectiveness and a set of test problems.

3.1. *Measure of effectiveness*

Since the performance of the algorithms may vary over a range of problem instances, the performances of the proposed heuristic algorithms are compared using the following measures.

Average relative percentage deviation (ARPD): The ARPD, which indicates the average performance of the variants of VAM and TOM with respect to the optimal solution over the number of problem instances, is computed using the following equations:

$$\mathrm{ARPD}(H) \equiv \sum_{i=1}^{N} \mathrm{RPD}(H, i),$$

$$\mathrm{RPD}(H, i) \equiv \{(\mathrm{HHS}_i - \mathrm{OS}_i)/\mathrm{OS}_i\} \times 100,$$

where ARPD(H) is the ARPD of heuristic "H," where H indicates VAM–TC or VAMT–TC, or VAM–TOC, or VAMT–TOC; RPD(H, i) is the relative percentage deviation between the solution obtained using heuristic "H" and the optimal solution of the instance "i"; HHS_i the heuristic "H" solution (total transportation cost) of the instance "i"; OS_i the optimal solution (total transportation cost) of the instance "i," and N the number of problem instances.

Number of best solutions (*NBS*): a frequency which indicates the number of instances the TOM, VAM and its variants yielded a solution within 0–3% loss of optimality over the number of problem instances.

3.2. *Experimental design*

The performance of the VAM and its variants, and TOM are compared over 640 problem instances. The problems are randomly generated as per the experimental design framework presented in Kirca and Satir (1990) but are restricted to "full dense" transportation problems (the transportation problem is fully dense, if there exists a route from each origin to each destination). The details of the experimental design used are as follows:

- Problem size (m supply points $\times$ n demand points): The sizes of the transportation problems experimented with are $(10 \times 20), (10 \times 40), (10 \times 60)$, and (10×100).
- Cost structure (C_{ij}: i = supply point $1, 2, \ldots, m$ and j = demand point $1, 2, \ldots, n$): Problems with four cost ranges (R) are tested. The mean cost is taken to be equal to 500. The ranges used are

  $$R = (20, 100, 500, 1000).$$

  For each range, the costs are randomly generated from the following uniform distribution:

  $$\mathrm{U}\,(C_{ij} : [\text{mean cost} - R/2, \text{mean cost} + R/2]).$$

- Supply and demand structure (S_i and D_j): The mean demand is equal to 100. Given the mean demand, the mean supply is expressed as

  $$\text{Mean supply} = K\{(n \times \text{mean demand})/m\},$$

  where K indicates the degree of imbalance between total supply and total demand. The mean supply values are generated for four imbalance coefficients, namely $K = (1, 2, 5, 10)$

Table 1. Summary of experimental design.

No.	Problem factor	Levels	# Levels
1	Problem size ($m \times n$)	$\{10 \times 20; 10 \times 40; 10 \times 60, 10 \times 100\}$	4
2	Degree of imbalance (K)	$\{1, 2, 5, 10\}$	4
3	Cost structure — range (R)	$\{20, 100, 500, 1000\}$	4
Number of problem configurations			$4 \times 4 \times 4 = 64$
Problem instances per configuration			10
Total problem instances			$64 \times 10 = 640$
Cost structure (C_{ij}): U(C_{ij}: [mean cost $- R/2$, mean cost $+ R/2$]), where mean cost = 500			
Supply (S_i): U(S_i: [0.75 × mean supply, 1.25 × mean supply]), where mean supply = [($K \times n \times$ mean demand)/m] and mean demand = 100			
Demand (D_j): U(D_j: [75, 125])			

The S_i and D_j are then generated from the uniform distributions of

$$\mathrm{U}\,(S_i\colon [0.75 \times \text{mean supply}, 1.25 \times \text{mean supply}])$$

$$\mathrm{U}\,(D_j\colon [75, 125])$$

The experimental design for generating test problems using the above three parameters is summarized in Table 1. The experimental design adopted in this paper is implemented in C++. For each combination of values for $[(m \times n), K, R]$, 10 problem instances are randomly generated, yielding a total of $640[= 10 \times (4 \times 4 \times 4)]$ problem instances. All the 640 problem instances are unbalanced. For 76 of these problem instances, the total demand is greater than the total supply. For the remaining 564 problem instances, the total supply is greater than the total demand.

3.3. *Evaluation of VAM and its variants and TOM against the optimal solution*

There are many procedures available in the literature for getting an optimal solution to the transportation problem. Briefly, the procedure used in this research is as follows. For each problem instance, a linear programming model is developed and solved using the optimization package, LINDO (Schrage, 1991). In order to get a linear programming model for each problem instance, a matrix generator program was developed and implemented using Turbo C++. The matrix generator program will convert the problem data, viz. transportation costs, supply as well as demand, into the required linear programming model of the transportation problem. The matrix generator program and a sample data file for generating a linear programming model are given in Appendix B.

VAM and each of its variants and TOM were implemented using Turbo C++. For each problem instance, the heuristic solutions were obtained using VAM and each of its variants and TOM. The performance of the VAM and its variants and TOM in comparison with the optimal solution is presented below.

Performance measures — ARPD: First, for each problem instance, the value of the "RPD" of each variants of VAM and TOM with respect to the optimal solution were computed using the equation given in Section 3.1. Secondly, for each level of $[(m \times n), K, R]$, the values of average RPD (ARPD) was computed over 10 problem instances. Further, for each level of $[(m \times n), K]$, the average {ARPD} was computed over 40 problem instances (that is, over the number of cost ranges "R" and 10 problem instances within each cost range "R"). They are presented in Table 2.

From Table 2, it is clear that for each variant of VAM and TOM, the values of average {ARPD} significantly vary over the parameters considered. Therefore, changes in problem parameters have an influence on the performance of all the variants of VAM and TOM.

Table 2. Average performance of the variants of VAM and TOM.

Problem factor			Average {ARPD} over "R"				
Size ($m \times n$)	Degree of imbalance (k)	Cost structure, Range (R)	Basic variants of VAM		Proposed variants of VAM		Kirca and Satir
			VAM–TC	VAMT–TC	VAM–TOC	VAMT–TOC	TOM
10 × 20	1	{20, 100, 500, 1000}	4.41[a]	4.42	3.74	3.74	6.44
	2	{20, 100, 500, 1000}	10.70	10.70	1.53	1.53	41.06
	5	{20, 100, 500, 1000}	12.04	12.04	2.10	2.10	55.07
	10	{20, 100, 500, 1000}	18.09	18.09	1.35	1.35	118.10
Overall average			11.31	11.31	2.18	2.18	55.17
10 × 40	1	{20, 100, 500, 1000}	2.73	2.72	2.40	2.40	8.62
	2	{20, 100, 500, 1000}	3.78	3.81	0.13	0.13	48.32
	5	{20, 100, 500, 1000}	10.13	10.15	0.16	0.16	86.10
	10	{20, 100, 500, 1000}	14.81	15.45	0.40	0.40	142.90
Overall average			7.86	8.03	0.77	0.77	71.49
10 × 60	1	{20, 100, 500, 1000}	4.71	7.51	4.01	4.01	8.46
	2	{20, 100, 500, 1000}	2.77	18.22	0.35	0.35	31.66
	5	{20, 100, 500, 1000}	1.71	31.19	0.12	0.12	81.43
	10	{20, 100, 500, 1000}	1.73	8.88	0.18	0.18	95.79
Overall average			2.73	16.45	1.17	1.17	54.33
10 × 100	1	{20, 100, 500, 1000}	2.67	7.26	2.18	2.19	8.22
	2	{20, 100, 500, 1000}	2.09	20.65	0.16	0.16	29.10
	5	{20, 100, 500, 1000}	2.44	29.39	0.23	0.23	71.87
	10	{20, 100, 500, 1000}	3.62	15.16	0.29	0.30	109.59
Overall average			2.70	18.12	0.72	0.72	54.69

[a]Each "cell" value indicates the average of "ARPD" over the 40 problem instances (that is, 10 problem instances for each "R").

Table 2 indicates that irrespective of the problem parameters considered in this study, on an average, the variants VAM–TOC and VAMT–TOC yielded more efficient results as compared with VAM–TC, VAMT–TC, and TOM. This indicates that coupling TOC with VAM yields consistently better starting solutions than obtainable with either the basic version of VAM or TOM. Also, the "tie-breakers" used in this study do not have any influence on the solution yielded by VAM coupled with TOC (see Figure 1). However, the "tie-breakers" have some influence (see, Figure 2) in the solutions of the basic version of VAM. That is, on an average, it appears that the "tie-breakers" along with VAM progressively increase the "average of {ARPD with respect to optimal solution}" as the size ($m \times n$) increases! Thus, it appears from both Figures 1 and 2 that "tie-breakers" do not have any positive influence on the performance of VAM.

Incidentally, it is observed that our results have shown that all the variants of VAM (including the basic version of VAM) outperform TOM, much against the claims of its author. This, however, was not our objective in this study.

The behavior of the performance measure shown in Table 2 for each problem size ($m \times n$), considered would lead us to the conclusion that inferences are applicable to any problem size.

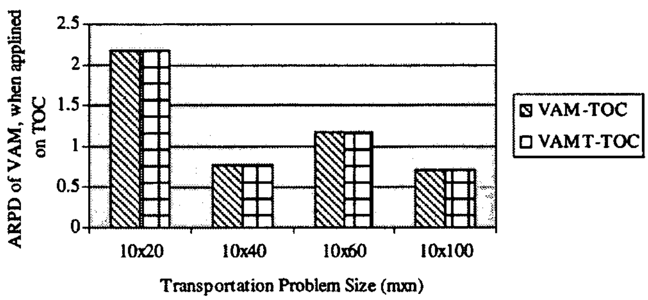

Fig. 1. Effect of tie-breakers on VAM, when applied on the TOC matrix.
Note: VAM with or without "tie-breakers" applied on TOC yielded the same ARPD.

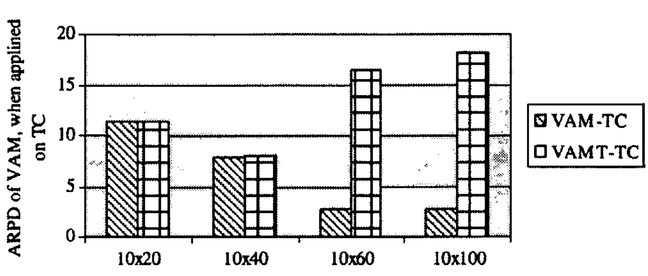

Fig. 2. Effect of tie-breakers on VAM, when applied on the TC matrix.
Note: VAM with or without "tie-breakers" applied on TC yielded different ARPD.

Similar inferences could be made with respect to the worst-case analysis, carried out using the performance measure: maximum relative percentage deviation. However, in this paper, these results are not presented.

Performance measure — Number of best solutions (NBS): From the detailed results obtained, the number of times the VAM and its variants and TOM yielded 0%, 0.5%, 1%, 2%, and 3% loss of optimality were observed over the 640 problem instances. These are presented in Table 3. The performance measure NBS also provides the same indications as those of ARPD that the variant VAM–TOC and VAMT–TOC are better options for obtaining an initial basic feasible solution. (VAM–TOC and VAMT–TOC, yielded the optimal solution 20% of the times and about 80% of the times yielded very efficient solutions with 0.5% loss of optimality.) Further, the basic version of the VAM appears to be better than TOM in all cases (0–3% loss of optimality).

Table 3. Number of best solutions w.r.t. loss of optimality in %.

Loss of optimality (%)	Number of problem instances w.r.t. loss of optimality in %				
	Basic variants of VAM		Proposed variants of VAM		Kirca and Satir
	VAM–TC	VAMT–TC	VAM–TOC	VAMT–TOC	TOM
0	1	0	128[a]	113	0
0.5	153	130	465	466	79
1	200	183	510	509	132
2	325	277	549	547	212
3	391	314	569	569	234

[a]Indicates 128 times the solution yielded by VAM–TOC matched with the optimal solution.

Table 4. Performance of VAM–TOC w.r.t. "0% loss of optimality vs. cost range.

Problem size	Number of problem instances with 0% loss of optimality in the cost range:				Total problem instances
	$R = 20$: [490–510]	$R = 100$: [450–550]	$R = 500$: [250–750]	$R = 1000$: [0–1000]	
10 × 20	19	—	10	—	29
10 × 40	30	—	—	—	30
10 × 60	30[a]	—	—	—	30
10 × 100	31	8	—	—	39
Total problem instances	110	8	10	—	128

[a]Indicates, 30 times the solution yielded by VAM–TOC matched with the optimal solution for the cost range $R = 20$.

Using the detailed results, Table 4 was developed to study the impact of cost range "R," with 0% loss of optimality solution. It appears from Table 4 that, the chances of getting 0% loss of optimality is high when the cost range is small.

The last, by no means insignificant, observation is that the CPU time required for the large size transportation problems tested in this study [that is (10×100)] is very small (less than 10 s on a Pentium 200 MHz machine with 64 MB RAM).

4. Conclusions

Two variants of Vogel's approximation method are proposed in this paper by coupling the basic idea of Kirca and Satir (1990) with VAM. In order to empirically evaluate the VAM and its variants and TOM, 640 problem instances were generated. The performance analyses of these methods were carried out with the optimal solution.

Based on the test problems generated and used in this study, the method: VAM–TOC *optimally* solved 128 out of 640 instances. Further, if the user is interested in getting a very fast and good movement strategy on their transportation problem without undue concern for the "best solution," they may prefer to implement the heuristic VAM–TOC and this is expected to provide, on an average, a very nearly optimal solution.

The CPU time required for the transportation problems tested [that is (10×100)] is on the average less than 10 s on a Pentium machine (200 MHz with 64 MB RAM). Thus, with today's computational power, any large-scale problem can be solved using more than one efficient variant of VAM without any computational difficulty in the decision support systems environment to obtain a near optimal solution.

Further, this study has unwittingly shown a result contrary to earlier research findings. That is, all the variants of VAM (including the basic version of VAM) outperform TOM. Lastly, from the computational analysis, it appears that the "tie breaker" does not have any influence on the performance of VAM–TOC (the best variant).

Generally, it appears from this study that the VAM is expected to yield a very efficient starting solution when applied in conjunction with TOC instead of with the original transportation cost. This inference indicates that studies using some form of non-dimensionalized transportation costs along with VAM and its variants may be a fruitful research direction.

Acknowledgments

The authors would like to acknowledge the referees for their helpful comments and suggestions. Also, the authors gratefully acknowledge the Singapore–MIT Alliance, School of MPE, Nanyang Technological University, Singapore, particularly Associate Professor Appa Iyer Sivakumar for the support and resources made available towards completing the writing up of this research.

References

Arsham, H and AB Kahn (1989). A simplex-type algorithm for general transportation problems: An alternative to Stepping-Stone. *Journal of Operational Research Society*, 40(6), 581–590.

Charnes, A and WW Cooper (1954). The Stepping-Stone method for explaining linear programming calculations in transportation problems. *Management Science*, 1(1), 49–69.

Dantzig, GB (1963). *Linear Programming and Extensions*, Princeton, NJ: Princeton University Press.

Dhose, ED and KR Morrison (1996). Using transportation solutions for a facility location problem. *Computers and Industrial Engineering*, 31(1/2), 63–66.

Gass, SI (1990). On solving the transportation problem. *Journal of Operational Research Society*, 41(4), 291–297.

Glover, F, D Karney, D Klingman and A Napier (1974). A computation study on start procedures, basis change criteria, and solution algorithms for transportation problems. *Management Science*, 20(5), 793–813.

Goyal, SK (1984). Improving VAM for unbalanced transportation problems. *Journal of Operational Research Society*, 35(12), 1113–1114.

Goyal, SK (1991). A note on a heuristic for obtaining an initial solution for the transportation problem. *Journal of Operational Research Society*, 42(9), 819–821.

Ji, P and KF Chu (2002). A dual-matrix approach to the transportation problem. *Asia-Pacific Journal of Operations Research*, 19(1), 35–45.

Kirca, O and A Satir (1990). A heuristic for obtaining an initial solution for the transportation problem. *Journal of Operational Research Society*, 41(9), 865–871.

Raghavendra, BG and M Mathirajan (1987). Optimal allocation of buses to depots — a case study. *OPSEARCH*, 24(4), 228–239.

Ramakrishnan, CS (1988). An improvement to Goyal's modified VAM for the unbalanced transportation problem. *Journal of Operational Research Society*, 39(6), 609–610.

Reinfeld, NV and WR Vogel (1958). *Mathematical Programming.* Englewood Cliffs, New Jersey: Prentice-Hall.

Schrage, L (1991a). *LINDO — An Optimization Modeling System*, 4th edition. CA, USA: The Scientific Press.

Schrage, L (1991b). *LINDO — User's Manual: Release 5.0.* CA, USA: The Scientific Press.

Shafaat, A and SK Goyal (1988). Resolution of degeneracy in transportation problems. *Journal of Operational Research Society*, 39(4), 411–413.

Sharma, RRK and KD Sharma (2000). A new dual based procedure for the transportation problem. *European Journal of Operational Research*, 122, 611–624.

Shih, W (1987). Modified Stepping-Stone method as a teaching aid for capacitated transportation problems. *Decision Sciences*, 18, 662–676.

Shimshak, DG, JA Kaslik and TD Barclay (1981). A modification of Vogel's approximation method through the use of heuristics. *INFOR*, 19, 259–263.

Sridharan, R (1991). Allocation of buses to depots: a case study. *Vikalpa, The Journal for Decision Makers*, 16(2), 27–32.

Vasudevan, J, E Malini and DJ Victor (1993). Fuel savings in bus transit using depot-terminal bus allocation model. *Journal of Transport Management*, 17(7), 409–416.

Appendix A

A Systematic Procedure for **VAMT–TOC**

Step 1: *Balance* the given transportation problem if either (total_supply > total_demand) OR (total_supply < total_demand).

Step 2: *Obtain* the "Total Opportunity Cost (TOC)" matrix.

Step 3: *Apply* VAM with Tie-breakers on TOC and *obtain* feasible allocations. That is, when we apply the VAM on the TOC matrix, if more than one "TOC cell" is competing for allocation, the following tie-breakers are used in sequence:

1. Make the allocation to the cell with the smallest cost.
2. In the case of a tie in (1), make allocation to the cell with the largest possible allocation.
3. In the case of a tie in (2), make allocation to that cell with first occurrence.

Step 4: *Compute* total transportation cost for the feasible allocations obtained in Step 3 using the original balanced-transportation cost matrix.

A Systematic Procedure for **VAM–TOC**

Heuristic "VAM–TOC" that follows is identical to Heuristic "VAMT–TOC" except in Step 3. So, only the *modified Step 3* is given below:

Step 3: *Apply* VAM on TOC and *obtain* feasible allocations.

Appendix B

Matrix Generator Program for generating ILP model in MPS (Mathematical Programming Structure) format of Transportation Problem PLUS sample data files

---

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<stdlib.h>
#include<iomanip.h>
#include  "TRANCON.h"

void main ()
{
     clrscr();
```

```
     int i,j;
       int cost[S][D];
       int ns,nd,count,flag=0;
       long demand[D],supply[S];
long totdem=0,totsup=0;
     ifstream ndbfile,deadfile;
     ofstream mpsfile;
ndbfile.open("NSD.dat",ios::in);
ndbfile>>ns>>nd;
cout<<ns<<endl;

for(int d=1; d<=nd; d++)
  {
    ndbfile>>demand[d];
    totdem = totdem + demand[d];
  }

for (int b=1; b<=ns; b++)
  {
     ndbfile>>supply[b];
     totsup = totsup + supply[b];
  }
ndbfile.close();

deadfile.open("TCOST.dat",ios::in);
for (i=1; i<=ns; i++)
  {
     for (j=1; j<=nd; j++)
      {
       deadfile>>cost[i][j];
      }
  }
deadfile.close();
//Balancing the transportation problem**********Begin********
if (totdem > totsup)
{
flag = 1;
ns++;
 supply[ns] = totdem - totsup;
 totsup = totdem;
// cout<<"Dummy Supply :"<<supply[ns]<<endl;
// getch();
```

```
 for(d=1; d<=nd; d++)
  {
   cost[ns][d] = 0;
   }
 }
else
{
if (totdem < totsup)
  {
  flag = 2;
   nd++;
     demand[nd] = totsup - totdem;
//   cout<<"Dummy Demand : "<<supply[nd]<<endl;
   totdem = totsup;
   for(b=1; b<=ns; b++)
    {
     cost[b][nd] = 0;
    }
  }
}//Balancing the transportation problem**********End********
mpsfile.open("DATA.mps",ios::out);
mpsfile<<"NAME"<<setw(12)<<"TP"<<endl;
mpsfile<<"ROWS"<<endl;
mpsfile<<"  N"<<setw(4)<<"OBJ"<<endl;
  for (i=1; i<=ns; i++)
    {
      if (i==1)
       {
          for(j=1; j<=nd; j++)
          {
          if (j==1)
           {
//           mpsfile<<"  L"<<"  S"<<setfill('0')<<setw(3)<<i<<endl;
             mpsfile<<"  E"<<"  S"<<setfill('0')<<setw(3)<<i<<endl;
             mpsfile<<"  E"<<"  D"<<setfill('0')<<setw(2)<<j<<endl;
           }
         else
           {
             mpsfile<<"  E"<<"  D"<<setfill('0')<<setw(2)<<j<<endl;
           }
          }
       }
         else
```

```
          {
          mpsfile<<" E"<<" S"<<setfill('0')<<setw(3)<<i<<endl;
          }
    }
mpsfile<<"COLUMNS"<<endl;
 for(i=1; i<=ns; i++)
  {
    for(j=1; j<=nd; j++)
     {
       mpsfile<<"
S"<<setfill('0')<<setw(3)<<i<<"D"<<setfill('0')<<setw (2)<<j
     <<"  S"<<setfill('0')<<setw(3)<<i<<"       1"<<endl;
       mpsfile<<"
S"<<setfill('0')<<setw(3)<<i<<"D"<<setfill('0')<<setw (2)<<j
       <<" D"<<setfill('0')<<setw (2)<<j<<"         1"<<endl;
       mpsfile<<"
S"<<setfill('0')<<setw(3)<<i<<"D"<<setfill('0')<<setw(2)<<j
     <<" OBJ"<<" "<<cost[i][j]<<endl;
      }
  }
mpsfile<<"RHS"<<endl;
  for (i=1; i<=ns; i++)
   {
    mpsfile<<"   RHS"<<"        S"<<setfill('0')<<setw(3)<<i<<"
"<<supply[i]<<endl;
   }
  for(j=1; j<=nd; j++)
   {
     mpsfile<<"   RHS"<<"        D"<<setfill('0')<<setw(2)<<j<<"
"<<demand[j]<<endl;
   }
 mpsfile<<"ENDATA";
 mpsfile.close();
}
```

Content of ''TRANCON.h"

```
#define S 50
#define D 120
#define NUM_FILES 50
#define FILE_NAME_SIZE 10
```

Sample data on "NSD.dat"

```
10  20
105 93  114 83  99  76  107 115 108 85  77 84 116
83  92  100 110 114 115 77
241 187 250 205 179 249 169 215 227 205
```

Sample data on "TCOST.dat"

```
501 501 501 491 510 506 494 508 509 490 496 509
498 490 502 503 499 502 493 491
494 491 500 492 492 507 495 495 496 499 506 491
493 503 505 493 504 498 505 494
506 499 500 508 504 503 509 503 506 496 508 496
501 502 494 501 509 506 508 502
509 498 505 506 490 495 501 491 509 490 492 509
505 495 492 503 505 497 494 500
493 493 500 497 490 505 491 509 497 505 502 508
503 507 493 494 493 491 507 500
500 501 491 493 495 494 499 500 509 503 509 490
507 506 491 502 498 502 500 501
501 500 494 507 494 506 506 495 503 491 508 490
508 510 491 505 503 499 502 508
499 496 503 508 505 509 495 505 499 509 496 493
504 501 497 496 503 507 507 490
505 496 493 505 499 501 500 505 494 503 490 490
504 503 509 501 504 491 496 501
503 490 498 499 499 490 490 499 502 494 498 507
502 508 500 494 510 510 510 498
```

M. Mathirajan is on the faculty in the Department of Management Studies at Indian Institute of Science (IISc), Bangalore, India. He received an M.Sc. (Mathematics) degree from Madurai Kamaraj University, Tamilnadu, India, a Post-Graduate Diploma in Operations Research from Anna University, Chennai, India, and an M.Sc. (Engineering) degree by research in Applied Operations Research area as well as a Ph.D. in Operations Management from IISc, Bangalore, India. He was a post-doctoral fellow for the period: February 2002 to July 2003 at Singapore–MIT Alliance of Nanyang Technological University, Singapore. He was selected and nominated as young representative of Operational Research Society of India to present his paper in the International Session of APORS of the 1999 Fall Annual Conference of Operational Research Society of Japan (ORSJ), Tokyo, Japan. His Ph.D. thesis entitled "Scheduling of Parallel Heterogeneous Batch Processors" was adjudged the best thesis for the "M. N. Gopalan Award of 2002 — Annual Convention of ORSI". His research interest is in the areas of Mathematical Modeling for Production Planning and Scheduling of Batch Processors, Routing and Scheduling in Transportation Planning, Distribution in Logistics Management, Heuristic Methods for Distribution Management Problems, Container Terminal Management, and Scheduling Problems. He has published about 20 research articles in both national and international journals and also presented more than 19 research papers in both national and international conferences. He is a co-author of one book and co-author of two monographs and has guided a number of graduate and post-graduate student projects.

B. Meenakshi was a Project Assistant (September 1999 to January 2002) in the Department of Management Studies at Indian Institute of Science (IISc), Bangalore. She obtained her MCA degree with distinction from Madurai Kamaraj University, Tamilnadu, India. In addition, she did certificate courses on (a) Web Programming with Java, offered by Computer Society of India, (b) Visual Basic with MS Access, offered by Bitech, Madurai and (c) Graduate Course in Computer Application, offered by Cybernetic Infotech System, Madurai. Her interest is in the area of software development for managerial problems related to distribution in logistics management and portal.