Arbeidsnotat
Working Paper

2013:7

Øyvind Halskau sr.
Kurt Jörnsten

Some new bounds for the travelling salesman problem

Høgskolen i Molde
Vitenskapelig høgskole i logistikk
MØREFORSKING
MOLDE

Øyvind Halskau sr.
Kurt Jörnsten

Some new bounds for the travelling salesman problem

Arbeidsnotat / Working Paper 2013:7

Høgskolen i Molde
Vitenskapelig høgskole i logistikk

Molde University College
Specialized University in Logistics

Molde, Norway 2013

ISSN 1501-4592 (trykt)
ISBN 978-82-7962-172-0 (trykt)
ISBN 978-82-7962-173-7 (elektronisk)

Some new bounds for the travelling salesman problem

by

Øyvind Halskau sr[1]. and Kurt Jörnsten[2]

[1] Corresponding author, Molde University College, Postbox 2110, NO-6402, Molde, Norway, oyvind.halskau@himolde.no , Phone: 00 47 71214000, Fax: 0047 71214100
[2] Norwegian school of business administration, Helleveien 30 NO-5045, Bergen, Norway. Kurt.jornsten@nhh.no

Abstract

The Clarke and Wright heuristic for the travelling salesman problem (TSP) has been used for several decades as a tool for finding good solutions for TSP and other vehicle routing problems (VRP). In this paper we offer a simple, but fundamental relationship between the cost of a Hamiltonian cycle measured in the original cost matrix and the cost of the same cycle measured in a saving matrix. This relationship leads to a new and simple lower bound for TSP that some times is better than more traditional bounds based on so-called 1-trees. We also offer some upper bounds for the optimal solution of TSP. Some examples are given in order to illustrate the new bounds and compare these with the classical ones.

Key words: Clark and Wright savings, Travelling Salesman Problem, lower bounds, upper bounds

1. Introduction

The Clark and Wright saving algorithm is one of the oldest heuristics for the vehicle routing problem, see Clark and Wright (1964) and probably also one of the most popular. It has been used as a basic heuristic in different software programs and as example of a heuristic in many text books. The literature about the Clark and Wright saving heuristic is huge. A recent article, (G. K Rand 2009) gives an comprehensive overview over its history and how this heuristic has been used in different versions, adapted to many different varieties of the VRP, used in practical applications in order to find good solutions to real life problems. Rand shows that the Clarke and Wright heuristic has been and still is used as an integrated part of many software packages dealing with VRP and its many extensions. This heuristic was originally made for dealing with VRP, but can of course also be used when one wants to find solutions to TSP. In this case one can define the depot one self. This gives the possibility to calculate as many saving matrices as there are nodes in the underlying graph. Each of them yielding potentially different solutions. In this paper we will restrict our selves to dealing with TSP. We offer a simple but revealing relationship between the cost of a Hamiltonian cycle measured in the original cost matrix and the cost of the same cycle measured in a saving matrix. This relationship gives rise to new and simple lower bounds for TSP as well as upper bounds for the same problem. By examples we show that these new lower bounds sometimes are better and sometimes are worse than more traditional and well known lower bounds for TSP. The rest of this paper is organised as follows: In section 2 the new relationship between the original cost matrix and the saving matrices are given. In section 3 the new lower bounds and its relationship to the traditional lower bounds are displayed. Further, a small illustrative example is given as well as three examples showing that it will be impossible to prove in general that one of these bounds always is better than the other. The details of these three examples are given in an appendix. In section 4 we offer some conclusions and thoughts for further research.

2. The travelling salesman problem and the saving algorithm

Let C be a cost matrix for a TSP with n nodes where the elements of the matrix is denoted by $c_{ij}, c_{ij} \geq 0$. In the original article by Clark and Wright the depot was given and the so-called saving values were calculated relatively to this depot. Applying the saving procedure to a TSP problem every node can be chosen as a depot. We denote the saving values relatively to the chosen depot d as s_{ij}^{d} and the resulting saving matrix based on the chosen depot as S^{d}. The savings can be calculated according to formula (2.1) below.

$$(2.1) \qquad s_{ij}^{d} = c_{id} + c_{dj} - c_{ij}$$

Note that what ever the values of the cost elements in C may be, the saving values for $s_{dj}^{d} = 0, \forall j$ and $s_{id}^{d} = 0, \forall i$. Hence, in S^{d} row number d and column number d all elements will be zero. We will denote the sum of all the elements in row number d in C as R_d and the sum of all the elements in column number d in C as K_d. Further, we will denote the cost of a Hamiltonian cycle H measured in C, as $H(C)$ and the corresponding cost in the saving matrix S^{d} with $H(S^{d})$.

It is well known that TSP is a NP-complete problem. However, many polynomial special cases have been identified over the years. Among these is the so-called constant-TSP, see Berenguer, X. (1979). In this special case of the TSP all the cost elements have the following structure: $c_{ij} = a_i + b_j$. The cost of every Hamiltonian cycle in such a cost matrix will have the same length and will be equal to $\sum_{i=1}^{n} a_i + \sum_{j=1}^{n} b_j$. The Clark and Wright saving algorithm is an example of a so called equivalent matrix transformation (EMT), see Gutin and Punnen (2002) page 23 – 24. Such transformations leave the optimal solution of TSP unchanged.

In Halskau and Jörnsten, (1995) and in Halskau (2000) the following simple lemma is proved. For convenience the proof is reproduced here.

Lemma 2.1

For any Hamiltonian cycle H the following relation holds for all choices of d:

$$(2.2) \qquad H(C) + H(S^{d}) = R_d + K_d$$

Proof:

Restructuring (2.1) gives $c_{ij} + s_{ij}^{d} = c_{id} + c_{dj}$. Then applying this structure on any Hamiltonian cycle H gives $H(C) + H(S^{d}) = \sum_{j=1}^{d} c_{jd} + \sum_{i=1}^{n} c_{di} = K_d + R_d$.

If the triangle inequality holds in C, the saving values will be non-negative and as a consequence will $H(S^{d})$ be non-negative for all Hamiltonian cycles and for all d. The result of the lemma can then be illustrated as in figure 1 below.

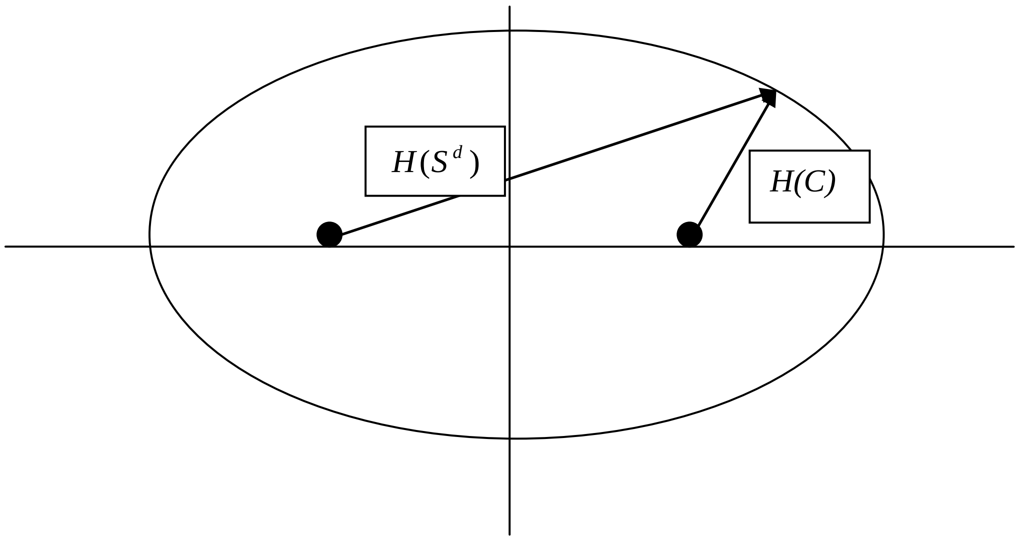

Fig1 Illustration of lemma 2.1

Lemma 2.1 and figure 1 shows that for any Hamiltonian cycle were the triangle inequality holds, the sum of the costs measured in the original cost matrix and the chosen saving matrix will end up on the periphery of an ellipse. Since there are only a finite number of Hamiltonian cycles, not every point on the periphery will represent a Hamiltonian cycle. For a given choice *d*, we can choose the foci points freely as long as the distance between the two focal points are strictly less than $K_d + R_d$.

Now let *H** denote the optimal solution for TSP. If the distance from the right hand side focal point to the right hand extreme point of the ellipse is larger than *H*(C)*, the optimal solution can not be represented in this ellipse. Hence, in order to include this possibility, the above mentioned distance must be less than the optimal solution. Since we do not necessarily know the optimal solution, we can let the distance from the focal point to the extreme point of the ellipse be less than any known lower bound for TSP, but still strictly positive.

Figure 1 indicates that for the optimal solution or Hamiltonian cycles with costs close to *H*(C)*, the cost of the cycle measured in *C* is smaller than the cost measured in the corresponding saving matrix. This turns out to be wrong as shown by a small example in section 3.

The classical way of applying the Clark and Wright heuristic is to calculate the savings for a given choice of the depot, then sorting the saving values in a decreasing order and then perform a greedy algorithm choosing the largest saving first and so on deleting any saving that leads to sub-cycles or that the degree of a node becomes larger than 2. One ends up with a path spanning all nodes but the depot. Then one goes back to the original cost matrix and calculate the costs of the found Hamiltonian cycle. This is not necessary any more. One can as well add the costs of the found path directly in the used saving matrix and then apply the equation in lemma 2.1 to find the cost in the original cost matrix. In other words the cost in a chosen saving matrix for a given Hamiltonian cycle and the cost of the same cycle in the original cost matrix are complementary. More over, any heuristic for the TSP can – with some minor changes – be applied on any of the saving matrices generating different solutions to the same basic problem. For more details about such possibilities, see Halskau and Jörnsten (1995) and Halskau (2000).

The following corollary follows directly from lemma 2.1.

Corollary 2.2

If the cost matrix C is symmetric then for any $d, H(C) + H(S^d) = 2R_d$

3. Bounds for the Travelling Salesman Problem

When solving the TSP to optimality, it is often convenient to have lower bounds. Such lower bounds can be of different qualities and can be found in many different ways. For an overview of such techniques, see for example Gutin and Punnen (2002) or Lawler et al. (1985).

In this paper we will restrict ourselves to consider some new simple bounds for TSP and compare these with other well known simple bounds.

Let C be any symmetric *nxn*-matrix. We let $R_i = \sum_{j=1}^{n} c_{ij}$ denote the row sum of row i and we let $R_{\min} = \min_i \{R_i\}$ and $R_{\max} = \max_i \{R_i\}$ be the smallest and the largest row sum, respectively. For symmetric matrices we let *MIST(C)* denote the minimal spanning tree in the graph with the cost matrix C. In a similar fashion *MAST(C)* will denote the cost of the maximal spanning tree. It is well known that the so-called *minimal spanning one tree* in a cost matrix for a symmetric TSP – that is the *MIST(C)* plus the smallest unused edge when making the *MIST* – is a lower bound for the TSP. We will denote this as *MIN(1-tree)(C).* Further, we can delete one node i from the graph, make the *MIST* among the remaining nodes and then add the two smallest edges from the deleted node. We then have a *1-tree*. This tree is evidently also a lower bound for TSP and will be equal to or larger than the *MIN(1-tree).* We will denote these lower bounds as DB_i and let $DB_{\max} = \max_i \{DB_i\}$. Hence, we have the following string of inequalities for some simple and well known lower bounds for TSP.

$$(3.1) \qquad MIST(C) \le MIN(1-tree)(C) \le DB_i \le DB_{\max} \le H^*(C)$$

Lemma 3.1

In any cost matrix where the triangle inequality holds, the following inequalities hold:

$$H(C) \le \min_d (R_d + K_d)$$
$$R_d + K_d - \min_d \{R_d + K_d\} \le H(S^d) \le R_d + K_d$$

Proof:

From lemma 2.1 we have – since $H(S^d) \ge 0$, $H(C) = R_d + K_d - H(S^d) \le R_d + K_d$. Since we can choose any d the first inequality comes forth.

Again, from lemma 2.1 we have that $H(S^d) = R_d + K_d - H(C) \le R_d + K_d$. On the other hand we have that $H(S^d) = R_d + K_d - H(C) \ge R_d + K_d - \min_d \{R_d + K_d\}$.

If we restrict the cost matrices to be symmetrical we get corollary 3.1.

Corollary 3.1

In any symmetrical cost matrix where the triangle inequality holds, the following bounds hold:

$$DB_{\max} \le H(C) \le 2R_{\min}$$
$$2(R_d - R_{\min}) \le H(S^d) \le 2R_d$$

New lower bounds for symmetric TSP

These new bounds will depend on which node that is chosen as the depot and the associated saving matrix. We will denote these new lower bounds as SB_d. Since the depot can be chosen in *n* different ways, we will get *n* lower bounds. Hence, the best one among these will be denoted as $SB_{\max} = \max_d \{SB_d\}$.
Corollary 3.1 can now be sharpened as shown in lemma 3.2

Lemma 3.2

In any symmetrical cost matrix where the triangle inequality holds the following bounds hold:

$$H(C) \in [\max\{SB_{\max}, DB_{\max}\}, 2R_{\min}]$$
$$H(S^d) \in [2R_d - 2R_{\min}, \min\{MAST(S^d), 2R_d - \max\{SB_{\max}, DB_{\max}\}\}]$$
where $SB_{\max} = \max_d \{2R_s - MAST(S^d)\}$

Proof:
From Corollary 2.2 we have that $H^*(C) + H^*(S^d) = 2R_d$. Hence, since $H^*(C)$ is at its minimum, $H^*(S^d)$ will be at its maximum. Since all the saving values from the chosen depot *d* to any of the other nodes are zero, the two edges from the depot in the optimal solution can be deleted without changing the value of $H^*(S^d)$. The remaining graph will be a spanning path among all the nodes except the depot. A spanning path is a tree and hence $H^*(S^d) \le MAST(S^d)$. Note that if $MAST(S^d)$ becomes a path, the optimal solution is found.

From the above consideration we can obtain a new lower bound for the optimal solution measured in *C*.

$$H^*(C) + H^*(S^d) = 2R_d \Rightarrow H^*(C) = 2R_d - H^*(S^d) \ge 2R_d - MAST(S^d) = SB_d.$$

Including this new bound in the sequence of inequalities given by 3.1, we have

$$(3.2) \qquad MIST(C) \le MIN(1-tree)(C) \le DB_i \le \max\{SB_{\max}, DB_{\max}\} \le H^*(C)$$

In a similar fashion we obtain new upper bounds for $H^*(S^d)$ as shown below

$$H^*(C) + H^*(S^d) = 2R_d \Rightarrow H^*(S^d) = 2R_d - H^*(C) \le 2R_d - \max\{SB_{\max}, DB_{\max}\}.$$

Hence,

$$(3.3) \qquad H^*(S^d) \le \min\{MAST(S^d), 2R_d - \max\{SB_{\max}, DB_{\max}\}\}$$

In order to illustrate these bounds we offer a small example. Consider the small graph in figure 2 and the corresponding cost matrix in table 1 below.

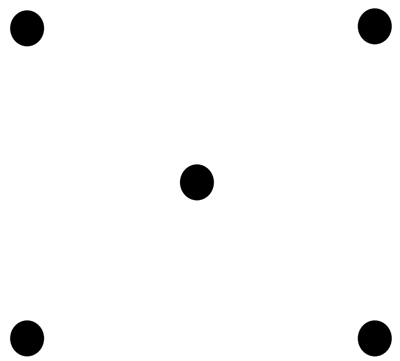

Fig. 2 Graph of the small example

Table 1 Cost matrix for the small example

	1	2	3	4	5	Row sum
1	-	2	$\sqrt{2}$	2	$2\sqrt{2}$	$4+3\sqrt{2}$
2		-	$\sqrt{2}$	$2\sqrt{2}$	2	$4+3\sqrt{2}$
3			-	$\sqrt{2}$	$\sqrt{2}$	$4\sqrt{2}$
4				-	2	$4+3\sqrt{2}$
5					-	$4+3\sqrt{2}$

Now the *MIST(C)* consists of the edges: (1,3); (2,3); (3,4); and (3,5) and the cost becomes $4\sqrt{2}$. Adding the smallest edge not used in the *MIST* – for example (1,2) – gives a cost for *MIN(1-tree)(C)* of $2+4\sqrt{2}$.

It is easily seen that an optimal cycle will be *H** = 1 – 2 – 3 – 5 – 4 – 1 with cost $6+2\sqrt{2}$ which is strictly larger than the two lower bounds found so far.

Due to the symmetry of the graph it is only necessary to calculate the two lower bounds DB_1 and DB_3. Deleting node 1, gives the same 1-tree as *MIN(1-tree)*. Hence, $DB_1 = 2 + 4\sqrt{2}$.

Deleting node 3, will give the 1-tree consisting of the edges (1,2); (1,4); (4,5) and the added edges (3,2) and (3,5) gives a cost of 6+2$\sqrt{2}$, confirming that the H* above is an optimal solution. Hence, DB_3 finds the optimal solution.

A Hamiltonian cycle with a maximal cost will be for example the cycle 1 – 5 – 3 – 2 – 4 – 1 with cost 2+6$\sqrt{2}$. This is strictly less than two times the smallest row sum, which becomes 8$\sqrt{2}$.

Due to the same symmetry we only need to find the saving tables corresponding to the choices of node 1 and 3 as the depot. These two saving matrices are given in table 2 and 3 respectively

Table 2 Saving values when node 1 is chosen as the depot node

	1	2	3	4	5
1	-	0	0	0	0
2		-	2	4-2$\sqrt{2}$	2$\sqrt{2}$
3			-	2	2$\sqrt{2}$
4				-	2$\sqrt{2}$
5					-

For this saving matrix the maximal spanning tree will consists of the three edges (5,2); (5,3), and (5,4) with cost 6$\sqrt{2}$. The lower corresponding lower bound will be $SB_1 = 2R_1 - MAST(S^1) = 8 + 6\sqrt{2} - 6\sqrt{2} = 8$ which is better than DB_1 but less than DB_3 .

Table 3 Saving values when node 3 is chosen as the depot node

	1	2	3	4	5
1	-	$2\sqrt{2}-2$	0	$2\sqrt{2}-2$	$2-\sqrt{2}$
2		-	0	0	$2\sqrt{2}-2$
3			-	0	0
4					$2\sqrt{2}-2$
5					-

For this saving matrix the maximal spanning tree will consists of the three edges (1,4); (1,5); and (4,5) with cost 6$\sqrt{2}$ - 6. The lower corresponding lower bound will be $SB_3 = 2R_3 - MAST(S^3) = 8\sqrt{2} - 6\sqrt{2} + 6 = 6 + 2\sqrt{2}$ which is equal to DB_3 .

Note that in this case the cost of the optimal cycle measured in the saving matrix S^3 becomes $H^*(S^3) = 2 + 4\sqrt{2}$ which is less than $H^*(C)$.

In this small example it turns out that the two simple lower bounds SB_{max} and DB_{max} become equal. A natural question to ask if this always is the case, and if not, will one of them always become better than the other? In the appendix three examples are given showing that the answers to these questions are negative.

Example 1 in the appendix is a graph with 16 nodes distributed in a grid where the distance horizontally and vertically between two neighbouring nodes is one.

It is easy to see that a minimal 1-tree in this matrix has the cost 16 and there exists several Hamiltonian cycles with the same cost, for example the cycle 1 – 5 – 9 – 13 – 14 – 15 – 16 – 12 – 11 – 10 – 6 – 7 – 8 – 4 – 3 – 2 – 1. Since the minimal 1-tree finds the optimum, all the lower bounds obtained by deleting a node will give the same result.

Due to the symmetry of the graph it is only necessary to find the saving bounds generated by choosing node 1, 5 or 6 as depot nodes. The results are summarised in table 4 below:

Table 4 The results using example 1 in the appendix

Node number	DB_i	$MAST(S^d)$	R_d	SB_i	$2R_d$	$2R_d - DB_{max}$	$2R_d - 2R_{min}$
1	**16**	64.12	38.49	12.87	76.98	70.98	26.12
5	**16**	52.54	32.53	12.52	65.06	49.03	14.20
6	**16**	37.44	25.43	**13.42**	50.86	34.86	0

As can be seen from table 4.The saving bounds can not compete with the minimal 1-tree in this case. The gap is fairly large. We can also see that any Hamiltonian cycle will be in the interval [16, 50.86]. We also see that the optimal cycle has a cost that is strictly less than the smallest row sum. The cost of any Hamiltonian cycle measured in the saving matrices 1, 5, and 6 will be in the intervals [26.12, 64.12], [14.20, 49.03], and [0, 34.86], respectively.

Example 2 in the appendix is a graph with 9 nodes distributed like a cross. Due to the symmetry of the graph it is only necessary to consider bounds associated with nodes 1, 2, and 5.

It is easily seen that the *MIST(C2)* = 8 and that *(1-tree)(C2)* $= 8 + \sqrt{2} \approx 9.41$. The optimal cycle becomes 1 – 2 – 5 – 4 – 3 – 8 – 9 – 7 – 6 – 1 with cost $H^*(C2) = 6 + 2\sqrt{5} + 2\sqrt{2} \approx 13.30$. The other bounds are given in table 5 below.

Table 5 The results using example 2 in the appendix

Node number	DB_i	$MAST(S^d)$	R_d	SB_i	$2R_d$	$2R_d - DB_{max}$	$2R_d - 2R_{min}$
1	**10.24**	40.26	20.13	12.05	40.26	30.04	16.26
2	9.41	28.60	14.30	12.07	28.60	18.36	4.60
5	**10.24**	11.51	12.00	***12.29***	24.00	13.76	0

As can be seen from the table, the best bounds found by deleting a node is strictly larger than the *MIN(1-tree).* On the other hand this bound is outperformed by all the three bounds based on the saving procedure, the best being found by using node 5 giving 12.29 as a lower bound for the TSP which is less than 0.8% from the optimal solution. In this example the optimal solution is larger than the smallest row sum. Disregarding the fact that we know the optimal solution, any Hamiltonian cycle will be in the interval [12.29, 48]. The cost of any Hamiltonian cycle measured in the saving matrices 1, 2, and 5 will be in the intervals [16.26, 30.04], [4.60, 18.36], and [0, 13.76], respectively.

Example 3 is identical with example 2 apart from that node 5 has been deleted from the graph. It is easily seen that the *MIST(C3)* = $4+3\sqrt{2}\approx 8.24$ and that *(1-tree)(C3)* $=4+4\sqrt{2}\approx 9.66$. The optimal cycle becomes 1 – 2 – 4 – 3 – 8 – 9 – 6 – 7 – 1 with cost $H^*(C3)=4+2\sqrt{5}+3\sqrt{2}\approx 12.71$. The other bounds are given in table 6 below.
Due to the symmetry it is only necessary to consider node 1 and 2.

These three examples show that it is in general, not possible to prove that one of the two candidates for a lower bound always will better than the other.

Table 6 The results using example 3 in the appendix

Node number	DB_i	$MAST(S^d)$	R_d	SB_i	$2R_d$	$2R_d - DB_{\max}$	$2R_d - 2R_{\min}$
1	**10.48**	24.12	18.13	***12.14***	36.26	25.26	9.66
2	**10.48**	14.43	13.30	12.07	26.60	16.12	0

As can be seen from the table, the best bounds found by deleting a node is strictly larger than the *MIN(1-tree)*, but these bounds are out performed by both bounds found by the saving procedure. On the other hand this bound is outperformed by both bounds based on the saving procedure, the best being found by using node 1 giving 12.14 as a lower bound for the TSP which is less than 5% from the optimal solution. In this example the optimal solution is smaller than the smallest row sum. Disregarding the fact that we know the optimal solution, any Hamiltonian cycle will be in the interval [12.14, 26.60]. The cost of any Hamiltonian cycle measured in the saving matrices 1 and 2 will be in the intervals
[29.66, 30.04] and [4.60, 18.36], respectively.

4. Conclusions and further research

We have presented new and simple lower bounds for TSP based on the Clarke and Wright heuristic. These lower bounds sometimes outperform more classical simple bounds obtained by making minimal 1-trees, and sometimes not. A natural question to ask can be if it is possible to find special cost structures such that one á priori can decide which type of a lower bound is the largest. Further, both the treated lower bounds can be calculated in *n* different ways. Hence, it could be convenient to find assumptions that make it unnecessary to calculate all the bounds of a specific type, that is, that the search for the best bound can be restricted to a subset of the nodes performing as starting nodes for the lower bound. Finally, will it be possible to establish saving based bounds for cost matrices where the triangle inequality does not hold or for asymmetric cost matrices?

References:

Berenguer, X: A Characterization of Linear Admissible Transformations for the m-Travelling Salesman Problem.European J. Oper. Res. 3, 232-249. [4:2], (1979)

Gutin, G. and Punnen, A. P. The travelling salesman problem and its variations.' Combinatorial Optimization, volume 12. Kluwer Academic Publishers, (2002)

Halskau, Ø. and Jörnsten, K. The Clark and Wright Heuristic Revisited**.** Proceedings from Nordic Operations Research Conference, University of Iceland, Reykjavik, (1995)

Halskau, Ø. Decompositions of Travelling Salesman problems. Doctoral thesis. Norwegian School of Economics and Business Administration,(2000)

Lawler, Lenstra et al. (ed):The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization. John Wiley & Sons, (1985)
Rand, G. K.The life and times of the Saving Method for Vehicle Routing Problems. Orion, volume 25 (2), pp 125- 145, (2009)

Appendix

Example 1

Table 7 The complete graph has 16 nodes in a grid with the following coordinates.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x-co	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
y-co	3	3	3	3	2	2	2	2	1	1	1	1	0	0	0	0

This gives the following symmetrical cost matrix

Table 8 The cost matrix for example 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	1	2	3	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{13}$	3	$\sqrt{10}$	$\sqrt{13}$	$3\sqrt{2}$
2		-	1	2	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$
3			-	1	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$
4				-	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{13}$	$2\sqrt{2}$	$\sqrt{5}$	2	$3\sqrt{2}$	$\sqrt{13}$	$\sqrt{10}$	3
5					-	1	2	3	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	2	$\sqrt{5}$	$2\sqrt{2}$	$\sqrt{13}$
6						-	1	2	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$
7							-	1	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$
8								-	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{13}$	$2\sqrt{2}$	$\sqrt{5}$	2
9									-	1	2	3	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
10										-	1	2	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
11											-	1	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$
12												-	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1
13													-	1	2	3
14														-	1	2
15															-	1
16																-

Due to the symmetry, there will be only 3 different row sums:

$$R_1 = R_4 = R_{13} = R_{16} = 12 + 6\sqrt{2} + 2\sqrt{5} + 2\sqrt{10} + 2\sqrt{13} \approx 38.49$$

$$R_2 = R_3 = R_5 = R_8 = R_9 = R_{12} = R_{14} = R_{15} = 8 + 4\sqrt{2} + 4\sqrt{5} + 2\sqrt{10} + \sqrt{13} \approx 32.53$$

$$R_6 = R_7 = R_{10} = R_{11} = 8 + 6\sqrt{2} + 4\sqrt{5} \approx 25.43$$

It is easy to see that a minimal 1-tree in this matrix has the cost 16 and there exists several Hamiltonian cycles with the same cost, for example the cycle 1 – 5 – 9 – 13 – 14 – 15 – 16 –

12 – 11 – 10 – 6 – 7 – 8 – 4 – 3 – 2 – 1. Hence the minimal 1-tree finds the optimal value. We also see that the smallest row sum is larger than the minimal 1-tree. Based on node no.1 we get the saving matrix in table 9.

Table 9 The saving matrix based on node 1 as the depot node

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2		-	**2**	2	0.59	1.41	1.82	1.93	0.76	1.24	1.54	1.78	0.84	1.16	1.44	1.63
3			-	**4**	0.76	2	3.24	3.75	1.17	2	2.82	3,37	1.39	2	2.61	3.08
4				-	0.84	2.18	3.82	**5.16**	1.39	2.40	3.59	4.61	1.76	2.56	3.44	4.24
5					-	1.41	1.24	1.16	**2**	1.82	1.59	1.44	2	1.93	1.78	1.64
6						-	2.65	2.58	2	2.65	**2.83**	2.78	2.41	2.34	2.19	2.05
7							-	4.40	2	3.06	4.06	**4.43**	2.41	3.16	3.84	4.24
8								-	2	3.16	4.58	**5.77**	2.57	3.50	4.53	5.40
9									-	3.24	2.83	2.61	**4**	3.75	3.37	3.05
10										-	4.06	3.84	3.82	4.40	**4.43**	4.24
11											-	5.43	3.59	4.58	5.43	**5.65**
12												-	3.44	4.53	5.80	**6.85**
13													-	**5.16**	4.61	4.24
14														-	**5.77**	5.40
15															-	**6.85**
16																-

From the table above we get the maximal spanning tree shown in figure A1 below. The value of this tree becomes 64.12 and hence, $SB_1 = 2R_1 - MAST(S^1) = 76.99 - 64.12 = 12.$, which is strictly less than the *(1-tree)(C).*

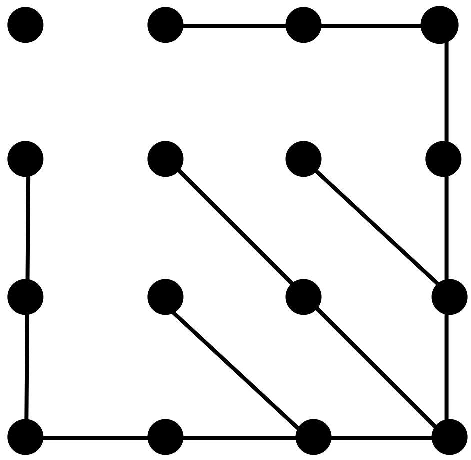

Fig.3 Maximal spanning tree based on saving matrix for node 1

Based on node no.5 we get the saving matrix in table 10.

Table 10 The saving matrix based on node 5 as the depot node

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	**1.41**	1.24	1.16	0	0.59	0.76	0.84	0	0.18	0.41	0.56	0	0.07	0.22	0.36
2		-	**2.65**	2.58	0	1.41	2	2.18	0.8	0.83	1.41	0.97	0.25	0.65	1.08	1.41
3			-	**4.40**	0	1.83	3.24	3.82	0.41	1.41	1.64	3.16	0.63	1.31	2.06	2.68
4				-	0	1.93	3.75	**5.16**	0.56	1.75	3.16	4.32	0.92	1.79	2.83	3.77
5					-	0	0	0	0	0	0	0	0	0	0	0
6						-	**2**	2	0.59	1.41	1.82	1.93	1	1	1	1
7							-	**4**	0.76	2	3.24	3.78	1.17	2	2.83	3.37
8								-	0.84	2.18	3.82	**5.16**	1.39	2.41	3.59	4.61
9									-	1.41	1.24	1.16	**2**	1.82	1.59	1.44
10										-	2.65	2.28	2	2.65	**2.83**	2.78
11											-	4.40	2	3.06	4.06	**4.43**
12												-	2	3.16	4.58	**5.77**
13													-	**3.24**	2.83	2.61
14														-	**4.06**	3.84
15															-	**5.43**
16																-

From the table above we get the maximal spanning tree shown in figure 4 below. The value of this tree becomes 52.44 and hence, $SB_5 = 2R_5 - MAST(S^5) = 65.06 - 52.54 = 12.52$, which is strictly less than the *(1-tree)(C)* and $DB_{\max}$.

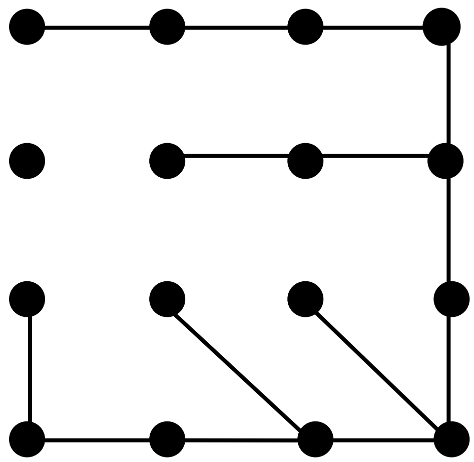

Fig. 4 The maximal spanning tree based on node 5 as the depot

Based on node no.6 e get the saving matrix in table 11.

Table 11 The saving matrix based on node 6 as the depot node

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-	**1.41**	0.83	0.65	1.41	0	0.18	0.25	0.83	0.18	0	0.04	0.41	0.49	0.64	0.78
2		-	**1.41**	1.24	0.59	0	0.59	0.76	0.18	0	0.18	0.41	0.07	0	0.07	0.22
3			-	**2.65**	0.18	0	1.41	2	0	0.18	0.83	1.41	0.04	0.25	0.65	1.08
4				-	0.07	0	1.82	**3.24**	0.04	0.41	1	1.65	0.82	0.87	1.07	1.47
5					-	0	0	0	1.41	0.59	0.18	0.07	1.24	0.76	0.41	0.22
6						-	0	0	0	0	0	0	0	0	0	0
7							-	**2**	0.18	0.59	1.41	1.82	0.41	0.76	1.24	1.59
8								-	0.25	0.76	2	**3.24**	0.63	1.17	2	2.83
9									-	1.41	0.83	0.65	**2.65**	2	1.41	1.08
10										-	1.41	1.24	1.82	**2**	1.82	1.59
11											-	2.65	1.41	2	2.65	**2.83**
12												-	1.31	2	3.06	**4.06**
13													-	**3.24**	2.47	2.06
14														-	**3.24**	2.83
15															-	**4.06**
16																-

From the table above we get the maximal spanning tree shown in figure 5 below. The value of this tree becomes 37.44 and hence, $SB_6 = 2R_6 - MAST(S^6) = 50.86 - 37.44 = 13.42$, which is strictly less than the *(1-tree)(C) and DB.*

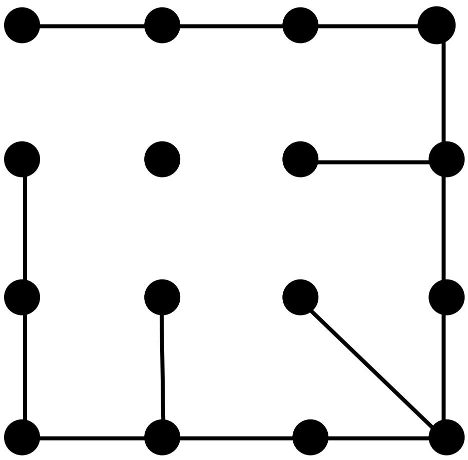

Fig.5 Maximal spanning tree based on node 6 as the depot node.

Example 2

We consider a graph with 9 nodes. The coordinates for the nodes are given in table 12.

Table 12. Coordinates for the nodes in example 2

Node	1	2	3	4	5	6	7	8	9

x-co	2	2	0	1	2	3	4	2	2
y-co	4	3	2	2	2	2	2	1	0

The cost matrix *C)* is given in table 13.

Table 13 Cost matrix *C2* for example 2

	1	2	3	4	5	6	7	8	9
1	-	1	$2\sqrt{2}$	$\sqrt{5}$	2	$\sqrt{5}$	$2\sqrt{2}$	3	4
2		-	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	3
3			-	1	2	3	4	$\sqrt{5}$	$2\sqrt{2}$
4				-	1	2	3	$\sqrt{2}$	$\sqrt{5}$
5					-	1	2	1	2
6						-	1	$\sqrt{2}$	$\sqrt{5}$
7							-	$\sqrt{5}$	$2\sqrt{2}$
8								-	1
9									-

It is easily seen that the *MIST(C2)* = 8 and that *(1-tree)(C2)* $= 8 + \sqrt{2} \approx 9.41$. The optimal cycle becomes 1 – 2 – 5 – 4 – 3 – 8 – 9 – 7 – 6 – 1 with cost $H^*(C2) = 6 + 2\sqrt{5} + 2\sqrt{2} \approx 13.30$. Due to the symmetry of the graph it is sufficient to calculate the *DB* only for node 1, 2, and 5. We get $DB_1 = 8 + \sqrt{5} \approx 10.24; DB_2 = 8 + \sqrt{2} \approx 9.41; DB_5 = 6 + 3\sqrt{2} = 10.24$.

Hence, none of these lower bounds finds the optimal value.

The saving values based on node 1 as the depot node are given in table B14.

Table 14 Saving matrix for example 2 based on node 1 as a depot node

	1	2	3	4	5	6	7	8	9
1	-	0	0	0	0	0	0	0	0
2		-	1.59	1.82	**2**	1.82	1.59	2	2
3			-	**4.06**	2.83	2.06	1.66	3.59	**4**
4				-	3.24	2.37	2.06	3.83	4
5					-	3.24	2.83	**4**	4
6						-	**4.06**	3.83	4
7							-	3.59	**4**
8								-	**6**
9									-

The corresponding maximal spanning tree is illustrated in figure 6 and has the value 28.12. The row sum $R_1 = 10 + 4\sqrt{2} + 2\sqrt{5} \approx 20.13$. Hence, $SB_1 = 2R_1 - MAST(S^1) = 40.26 - 28.12 = 12.08$, which is larger than *DB*

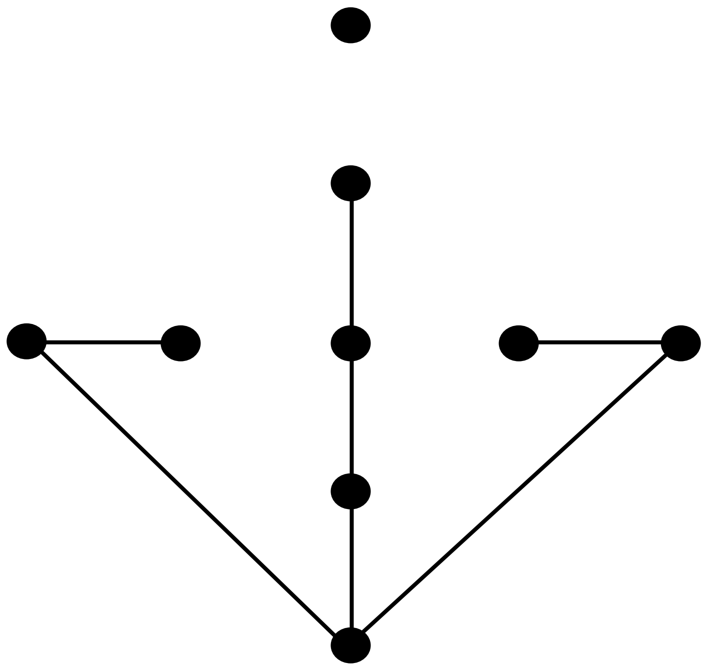

Fig.6 Maximal spanning tree for saving matrix based on node 1 as depot node in example 2

Choosing node 2 as the depot node gives the saving matrix in table 14.

Table 14 Saving matrix for example 2 based on node 2 as a depot node

	1	2	3	4	5	6	7	8	9
1	-	0	**0.41**	0.18	0	0.18	0.41	0	0
2		-	0	0	0	0	0	0	0
3			-	**2.65**	1.24	0.97	0.47	2	2.41
4				-	1.41	0.82	0.97	2	2.18
5					-	1.41	1.24	**2**	2
6						-	**2.65**	2	2.18
7							-	2	2.41
8								-	**4**
9									-

The corresponding maximal spanning tree is illustrated in figure 7 and has the value 16.53. The row sum $R_2 = 7 + 2\sqrt{2} + 2\sqrt{5} \approx 14.30$. Hence,
$SB_2 = 2R_2 - MAST(S^2) = 28.60 - 16.53 = 12.07$, which is larger than $DB_{\max}$.

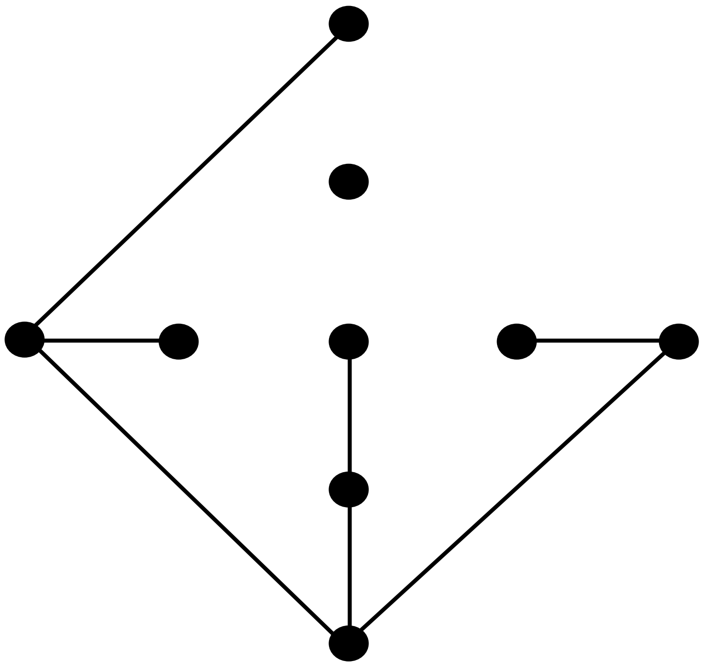

Fig.7 Maximal spanning tree for saving matrix based on node 2 as depot node in example 2

Choosing node 5 as the depot node gives the saving matrix in table 15.

Table 15 Saving matrix for example 2 based on node 5 as a depot node

	1	2	3	4	5	6	7	8	9
1	-	**2**	**1.17**	0.76	0	0.76	1.17	0	0
2		-	0.76	0.59	0	0.59	0.76	0	0
3			-	**2**	0	0	0	0.76	**1.17**
4				-	0	0	0	0.59	0.76
5					-	0	0	0	0
6						-	**2**	0.59	0.76
7							-	0.76	**1.17**
8								-	**2**
9									-

The corresponding maximal spanning tree is illustrated in figure 8 and has the value 11.51. The row sum $R_5 = 12$. Hence, $SB_5 = 2R_5 - MAST(S^5) = 24 - 11.51 = 12.49$, which is larger than $DB_{\max}$.

Hence, in this example all the saving bounds outperform the bounds obtained by deleting a node and then making a 1-tree in the original matrix as described in section 2.

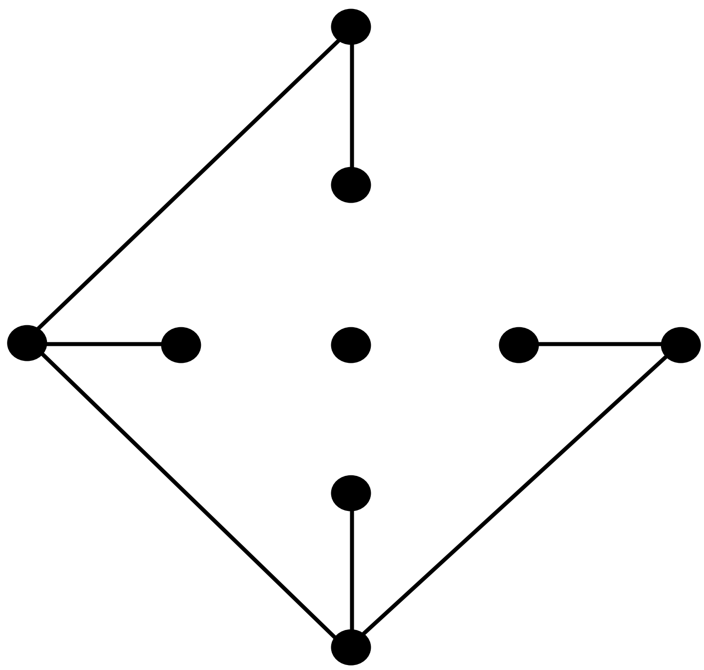

Fig.8 Maximal spanning tree for saving matrix based on node 1 as depot node in example 2

Example 3

This example is the same as the previous one except that node 5 has been removed. The cost matrix is given in table 16.

Table 16. Cost matrix *C3* for example 3

	1	2	3	4	6	7	8	9
1	-	1	$2\sqrt{2}$	$\sqrt{5}$	$\sqrt{5}$	$2\sqrt{2}$	3	4
2		-	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{5}$	2	3
3			-	1	3	4	$\sqrt{5}$	$2\sqrt{2}$
4				-	2	3	$\sqrt{2}$	$\sqrt{5}$
6					-	1	$\sqrt{2}$	$\sqrt{5}$
7						-	$\sqrt{5}$	$2\sqrt{2}$
8							-	1
9								-

It is easily seen that $MIST(C3) = 4 + 3\sqrt{2} \approx 8.24$ and that *(1-tree)(C3)* $= 4 + 4\sqrt{2} \approx 9.66$. The optimal cycle becomes 1 – 2 – 4 – 3 – 8 – 9 – 6 – 7 – 1 with cost $H^*(C3) = 4 + 2\sqrt{5} + 3\sqrt{2} \approx 12.71$. Due to the symmetry of the graph it is sufficient to calculate the *DB* only for node 1 and 2. We get $DB_1 = DB_2 = 4 + 3\sqrt{2} + \sqrt{5} \approx 10.48$.

Hence, none of these lower bounds finds the optimal value.

Due to the symmetry it is sufficient to calculate the saving values based on node 1 and 2 only. The saving values based on node 1 as a depot are shown in table 17.

Table 17. Saving matrix for example 3 with node 1 as a depot

	1	2	3	4	6	7	8	9
1	-	0	0	0	0	0	0	0
2		-	1.59	**1.82**	1.82	1.59	2	2
3			-	**4.06**	2.06	1.66	3.59	**4**
4				-	2.47	2.06	3.83	4
6					-	**4.06**	3.83	4
7						-	3.59	**4**
8							-	**6**
9								-

From table 17 we find the maximal spanning tree as illustrated in figure 9. The cost is 24.12. The edges used are marked with bold numbers.
Hence, $SB_1 = 2R_1 - MAST(S^1) = 36.26 - 24.12 = 12.14$, which is larger than DB_{max} .

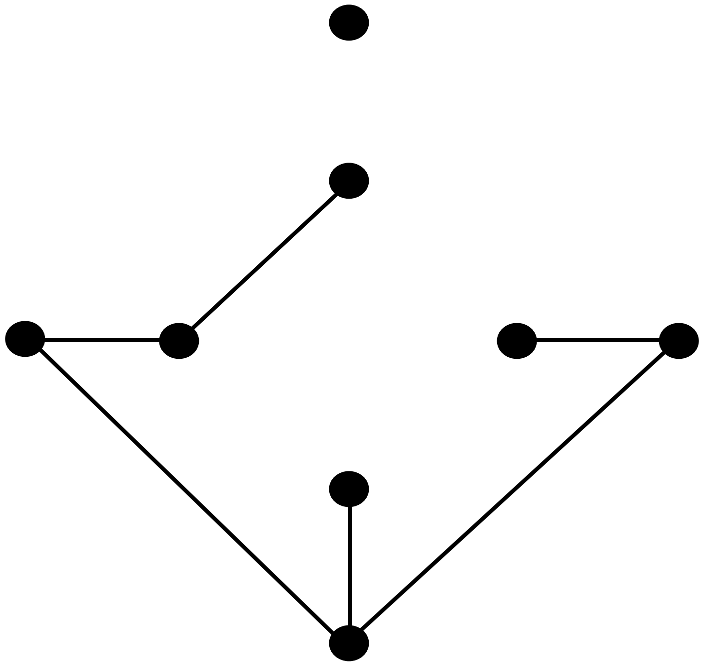

Fig.9 The maximal spanning tree for the saving matrix based at node 1 as the depot.

The saving values based on node 2 as a depot are shown in table 18.

Table 18 Saving matrix for example 3 with node 2 as a depot

	1	2	3	4	6	7	8	9
1	-	0	**0.41**	0.18	0.18	0.41	0	0
2		-	0	0	0	0	0	0
3			-	**2.65**	0.65	0.47	2	**2.41**

4				-	0.83	0.65	2	2.18
6					-	**2.65**	2	2.18
7						-	2	**2.41**
8							-	**4**
9								-

From table 18 we find the maximal spanning tree as illustrated in figure 10. The cost is 14.53. The edges used are marked with bold numbers.
Hence, $SB_2 = 2R_2 - MAST(S^2) = 26.60 - 14.53 = 12.07$, which is larger than DB_{max} .

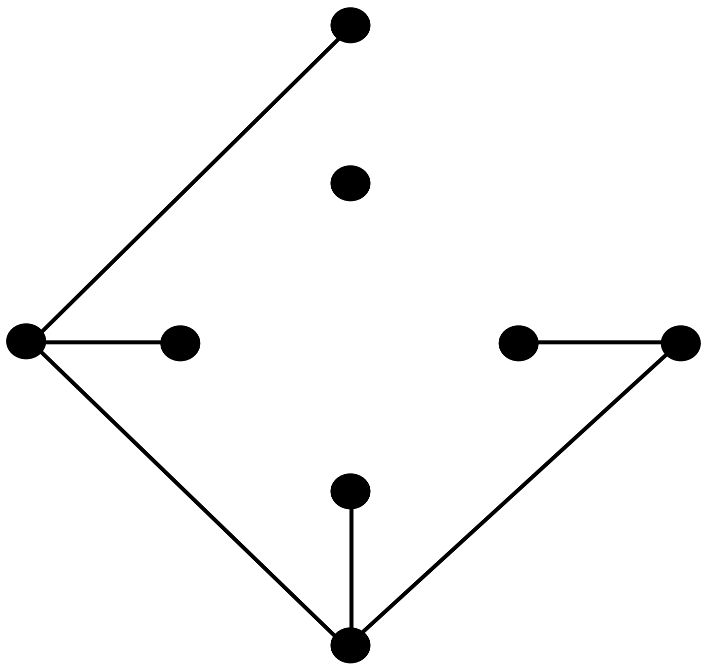

Fig.10 The maximal spanning tree for the saving matrix based at node 2 as the depot.

Høgskolen i Molde
PO.Box 2110
N-6402 Molde
Norway
Tel.: +47 71 21 40 00
Fax: +47 71 21 41 00
post@himolde.no
www.himolde.no

Møreforsking Molde AS
Britvegen 4
N-6411 MOLDE
Norway
Tel.: +47 71 21 42 90
Fax: +47 71 21 42 99
mfm@himolde.no
www.mfm.no