www.combinatorics.ir

Transactions on Combinatorics
ISSN (print): 2251-8657, ISSN (on-line): 2251-8665
Vol. 01 No.1 (2012), pp. 35-43.

www.ui.ac.ir

COMPLEXITY INDICES FOR THE TRAVELLING SALESMAN PROBLEM AND DATA MINING

DRAGOŠ CVETKOVIĆ

Communicated by Alireza Abdollahi

Abstract. We extend our previous work on complexity indices for the travelling salesman problem (TSP) using graph spectral techniques of data mining. A complexity index is an invariant of an instance I by which we can predict the execution time of an exact algorithm for TSP for I. We consider the symmetric travelling salesman problem with instances I represented by complete weighted graphs G. Intuitively, the hardness of an instance G depends on the distribution of short edges within G. Therefore we consider some short edge subgraphs of G (minimal spanning tree and several others) as non-weighted graphs and several their invariants as potential complexity indices. Here spectral invariants (e.g. spectral radius of the adjacency matrix) play an important role. Spectral clustering algorithms are used including information obtained from the spectral gap in Laplacian spectra of short edge subgraphs.

1. Introduction

The well-known travelling salesman problem (TSP) consists in finding a Hamiltonian cycle of minimal weight in a weighted complete graph. A survey of research on TSP can be found, for example, in books [9], [8].

A complexity index is an invariant of an instance I by which we can predict the execution time of an exact algorithm for TSP for I. Complexity indices are useful in designing adaptive algorithms and in creating search strategies within branch and bound algorithms [4]. Although there are in the literature some empirical and theoretical considerations on complexity indices (see, for example, [4], [1], [10], [12], [13]), a good theory of such phenomena is still missing.

A number of complexity indices are based on spectral graph theory.

MSC(2010): Primary: 90C27; Secondary: 05C50.

Keywords: Combinatorial optimization, graph spectra, computational complexity.

Received: 22 November 2011, Accepted: 09 March 2012.

A spectral graph theory is a theory in which graphs are studied by means of eigenvalues of a matrix M which is in a prescribed way defined for any graph. This theory is called *M–theory*. Frequently used graph matrices are: A adjacency matrix, D diagonal matrix of vertex degrees, $L = D - A$ Laplacian and $Q = D + A$ signless Laplacian. *The* spectral graph theory is the union of all these particular theories together with interaction tools [5].

Example. The adjacency matrix of the graph shown in Fig. 1

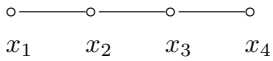

Fig.1

is given by $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

For the graph G on Fig.1 the characteristic polynomial reads

$$P_G(\lambda) = \begin{vmatrix} \lambda & -1 & 0 & 0 \\ -1 & \lambda & -1 & 0 \\ 0 & -1 & \lambda & -1 \\ 0 & 0 & -1 & \lambda \end{vmatrix} = \lambda^4 - 3\lambda^2 + 1\,.$$

Eigenvalues of G are zeros of $P_G(\lambda)$, i.e. $1.6180,\ 0.6180,\ -0.6180,\ -1.6180$ or

$$\frac{1+\sqrt{5}}{2},\quad \frac{-1+\sqrt{5}}{2},\quad \frac{1-\sqrt{5}}{2},\quad \frac{-1-\sqrt{5}}{2}$$

Note that eigenvalues and eigenvectors of a graph can be calculated in a polynomial time.

There are many applications of the theory of graph spectra to computer science [6]. This survey paper identifies the following areas of applications in computer science:

1. Expanders and combinatorial optimization,
2. Complex networks and the Internet topology,
3. Data mining,
4. Computer vision and pattern recognition,
5. Internet search,
6. Load balancing and multiprocessor interconnection networks,
7. Anti-virus protection versus spread of knowledge,
8. Statistical databases and social networks,
9. Quantum computing.
10. Bioinformatics,
11. Coding theory,
12. Control theory.

In the present paper we shall combine 1 and 3, i.e. to apply ideas of data mining to a problem of combinatorial optimization (TSP) using spectra of graphs.

Concerning applications in combinatorial optimization let us mention that one of early heuristics for graph bisection uses the *Fiedler vector*, i.e. the eigenvector belonging to the second smallest eigenvalue of the graph Laplacian. This eigenvalue is called *algebraic connectivity* of the graph and was introduced by M. Fiedler in [7].

The algebraic connectivity has been used in [2] to formulate the following discrete semidefinite programming model of the symmetric *travelling salesman problem* (STSP):

STSP semidefinite programming model:

minimize $F(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(-\frac{1}{2} d_{ij}\right) x_{ij} + \frac{\alpha}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}$

subject to
$x_{ii} = 2 + \alpha - \beta \ (i = 1, \ldots, n)$,

$\sum_{j=1}^{n} x_{ij} = n\alpha - \beta, \ (i = 1, \ldots, n)$,

$x_{ij} \in \{\alpha - 1, \alpha\} \ (j = 1, \ldots, n : \ i < j), \ X \geq 0$

Here $X \geq 0$ means that the matrix X is symmetric and positive semidefinite, while α and β are chosen so that $\alpha > h_n/n$ and $0 < \beta \leq h_n$ with $h_n = 2 - 2\cos(2\pi/n)$ being the algebraic connectivity of the cycle C_n. A solution X gives the Laplacian matrix L of an optimal Hamiltonian cycle via the formula $L = X + \beta I - \alpha J$, J being the all-1 matrix.

A natural semidefinite relaxation of the travelling salesman problem is obtained when discrete conditions are replaced by inequality conditions.

SDP relaxation:

minimize $F(X)$

subject to
$x_{ii} = 2 + \alpha - \beta \ (i = 1, \ldots, n)$,

$\sum_{j=1}^{n} x_{ij} = n\alpha - \beta, \ (i = 1, \ldots, n)$,

$\alpha - 1 \leq x_{ij} \leq \alpha, \ i, j = 1, \ldots, n, i < j, \ X \geq 0$

One can construct a branch and bound (B&B) algorithm based on the SDP relaxation.

The plan of the paper is as follows. In Section 2 we define the notion of a complexity index and explain basic ideas by summarizing our early work on the subject. Section 3 describes our work on complexity indices in 1990's. New ideas on using data mining techniques are explained in Section 4.

2. **Definition of a complexity index**

Definition 2.1. *Let A be an (exact) algorithm for solving an NP-hard combinatorial optimization problem C and let I be an instance of C of dimension n. A complexity index of I for C with respect to A is a real r, computable in polynomial time from I, by which we can predict (in a well defined statistical sense) the execution time of A for I.*

The efficiency of the complexity index can be statistically estimated measuring the linear correlation between the index value and the number of relaxation tasks solved within a B&B algorithm. Here we assume that the execution time of a B&B algorithm is proportional to the number of solved relaxation tasks.

The coefficient of linear correlation for two sequences $B = (b_i)$ and $C = (c_i)$ of length p is defined by

$$C_{BC} = \frac{1}{\vartheta_B \vartheta_C} \sum_{i=1}^{p} (b_i - \bar{m}_B)(c_i - \bar{m}_C),$$

where $\bar{m}_B$, $\bar{m}_C$ and ϑ_B, ϑ_C are mean values and variances of the corresponding sequences (b_i) and (c_i), respectively.

We consider the symmetric travelling salesman problem with instances I represented by complete graphs G with distances between vertices (cities) as edge weights (lengths).

Intuitively, the hardness of an instance G depends on the distribution of short edges within G. Therefore we consider some short edge subgraphs of G (minimal spanning tree, critical connected subgraph, critical 2-connected subgraph and several others) as non-weighted graphs and several their invariants as potential complexity indices. How short an edge should be to be considered as *short* depends on the context.

As a starting example consider an open tour TSP – B&B algorithm with minimal spanning tree problem (MST) as the relaxation.

If a minimal spanning tree is a path, it represents also a solution to the TSP. However, a path is also a tree with a minimal branching extent (in an intuitive sense).

The main idea is based on the expectation that a branch and bound algorithm will run *for longer the more the minimal spanning tree deviates from a path*, i.e. the greater "branching extent" it has.

Accordingly, any graph invariant characterizing well the "branching extent" in an intuitive sense, can be considered as a complexity index for the travelling salesman problem.

Originally, the following invariants have been considered: D the number of vertices of degree 2 in the minimal spanning tree and λ_1 the largest eigenvalue of the adjacency matrix of the minimal spanning tree.

The quantity D is maximal ($D = n-2$, where n is the number of vertices) if the tree reduces to a path, but it attains its minimal value $D = 0$ on a great number of trees.

The largest eigenvalue λ_1 reflects more precisely the branching extent of a tree. Given n, the number of vertices of a tree, the quantity λ_1 varies between $2cos\frac{\pi}{n+1}$ and $\sqrt{n-1}$, both bounds being attained on exactly one tree (a path and a star, respectively) [5]. Since the path P_n has the least branching extent in the intuitive sense and the star $K_{1,n-1}$ has the maximal one, the quantity λ_1 has at least a good property that it characterizes extremal trees in the above sense. Any invariant which is considered as a branching extent parameter should fulfil this criterion.

Further, let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be eigenvalues (of the adjacency matrix) and $d_1, d_2, \ldots, d_n$ vertices degrees of vertices in a graph. The quantity

$$S_k = \sum_{i=1}^{n} \lambda_i^k \quad (k = 0, 1, 2, \ldots)$$

is called the *k-th spectral moment* of the graph. For trees we have

$$S_4 = 6(n-1) - 2\sum_{i=1}^{n} d_i^2.$$

It has been shown that S_4 is maximal for the star $K_{1,n-1}$ and minimal for the path P_n.

The following two indices are also considered:

$F_s = d_1! + d_2! + \cdots + d_n!$ and $F_p = d_1!d_2!\cdots d_n!$.

Within early experiments in 1980's a number of instances of the TSP have been generated by means of a random generator using a uniform distribution in the interval (0,1) for the weights. For each instance we have computed the considered indices and the number N of the solved relaxation tasks when running the branch and bound algorithm.

Since the input matrices are randomly generated the indices and the number N are random variables. The linear correlation coefficient and the Spearman correlation coefficient between indices and N have been calculated. The results are given in Tables 1-4.

The first and the second column give the correlation coefficient between all mentioned indices and quantities N and $\log N$. The Spearman rank correlation coefficient is given in the third column.

Correlation coefficients have been found between the number of relaxation tasks and largest eigenvalue of minimal spanning tree for another branch and bound algorithm and for series of 100 randomly generated (uniform edge weight distribution) TSP instances with different number of vertices.

The results are presented in Table 5.

These experiments have shown that the considered complexity indices give some useful results only for a small number of vertices (say, up to 12). This is to be expected since the edges of a minimal spanning tree represent a small portion of the set of all edges of an instance with large n. Another disadvantage is that a minimal spanning tree is not uniquely determined.

3. **Experiments in 1990's**

Ten years later several invariants have been considered as complexity indices for the TSP with respect to B&B algorithms based on the SDP relaxation.

	N	$\log N$	N (Spearman)
D	0.361	0.605	0.500
λ_1	0.433	0.598	0.514
S_4	0.501	0.612	0.536
F_s	0.463	0.421	0.568
F_p	0.531	0.534	0.570

TABLE 1. 200 graphs on 8 vertices

	N	$\log N$	N (Spearman)
D	0.293	0.529	0.439
λ_1	0.378	0.543	0.502
S_4	0.469	0.587	0.545
F_s	0.643	0.375	0.530
F_p	0.690	0.517	0.551

TABLE 2. 100 graphs on 12 vertices

	N	$\log N$	N (Spearman)
D	0.349	0.499	0.477
λ_1	0.225	0.408	0.407
S_4	0.340	0.505	0.571
F_s	0.036	0.179	0.493
F_p	0.269	0.359	0.575

TABLE 3. 100 graphs on 14 vertices

	N	$\log N$	N (Spearman)
D	0.372	0.513	0.542
λ_1	0.383	0.333	0.329
S_4	0.498	0.444	0.537
F_s	0.206	0.144	0.473
F_p	0.235	0.168	0.581

TABLE 4. 100 graphs on 16 vertices

number of vertices	10	12	14
correlation coefficient	0.299	0.301	0.175

TABLE 5. 100 graphs of each order

Let X be the solution of the SDP relaxation problem with $\beta = h_n$ and $\alpha = 1$ and let $L = X + h_n I - J$ with entries ℓ_{ij}. Then L determines the weighted graph $W_L = (V, E_L, C_L)$, where $V = \{1, 2, \ldots, n\}$ is the vertex set, the edge set $E_L = \{\{i, j\} \in E | \ell_{ij} < 0\}$ and the weight matrix $C_L = 2I - L$, the corresponding unweighted graph $G_L = (V, E_L)$ and a stochastic matrix $S_L = I - \frac{1}{2}L$. The graph G_L plays the role of a short edge subgraph.

The most efficient indices turned out to be the following ones:

I_1**::** the number of edges of G_L

I_2**::** the second smallest eigenvalue of the Laplacian of G_L

I_3**::** the entropy of S_L, i.e. the value equal to

$$\sum_{(i,j)\in E_L} (\ell_{ij}/2) \log_2(-\ell_{ij}/2) - n/2$$

I_4**::** $\sum_{i=1}^{n} |\mu_i - \mu_i^*|$, where $\mu_1, \mu_2, \ldots, \mu_n$ and $\mu_1^*, \mu_2^*, \ldots, \mu_n^*$ are sequences of nondecreasing eigenvalues of the Laplacians of G_L and a Hamiltonian circuit, respectively.

I_5**::** the same sum as in I_4 but with eigenvalues of the Laplacian of W_L instead of G_L.

I_6**::** the number of vertices of the G_L with degrees greater than 2.

The efficiency of indices I_k, $k = 1, \ldots, 6$, has been investigated in [3].

For each dimension 20, 25, 30, 35 we have considered 50 randomly generated TSP instances with distances uniformly distributed in the interval [1,999]. To each instance a B&B algorithm based on SDP relaxation is applied.

The coefficients of the linear correlation between values of indices I_k $(k = 1, \ldots, 6)$ and the number of relaxation tasks for dimensions n= 20, 25, 30, 35 are summarized in Table 6. Results indicate that the most reliable indices are I_1, I_4 and I_6 with almost significant correlation.

index / n	I_1	I_2	I_3	I_4	I_5	I_6
20	0.53	0.35	0.51	0.53	0.53	0.53
25	0.48	0.49	0.21	0.48	0.48	0.49
30	0.29	0.21	0.32	0.29	0.42	0.33
35	0.56	0.52	0.37	0.56	0.38	0.55
average value	0.47	0.39	0.35	0.47	0.45	0.48

TABLE 6. Values of the linear correlation coefficients

The obtained experimental results, although much more encouraging than earlier, indicate the luck of theoretical explanations of phenomena with complexity indices, the need for experiments with instances of higher dimensions and, perhaps, the need for better classification of graph invariants than the intuitive approach.

Our previous work on complexity indices for the travelling salesman problem is summarized in [4].

4. **Data mining in TSP – an idea for investigations**

Now we want to extend our previous work on complexity indices for the travelling salesman problem using graph spectral techniques of data mining.

Data mining discovers interesting and unknown relationships and patterns in huge data sets. It is used in many domains such as image processing, web searching, computer security and many others including those outside computer science. Among many tools used in data mining, spectral techniques play an important role [14]. Here belong, in particular, clustering and ranking the vertices of a graph. A description of *spectral clustering* methods is given in the tutorial [11].

Since hidden details of short edge subgraphs really determine the hardness of an instance, we use techniques of data mining to find them. In particular, we use the spectral clustering algorithms to partition short edge subgraphs into clusters.

We shall present an algorithm for graph clustering which is based on the Laplacian matrix of a graph.

Let G be a connected graph on n vertices. Eigenvalues in non-decreasing order and corresponding orthonormal eigenvectors of the Laplacian $L = D - A$ of G are denoted by $\nu_1 = 0, \nu_2, \ldots, \nu_n$ and $u_1, u_2, \ldots, u_n$, respectively.

In order to construct k clusters in a graph we form an $n \times k$ matrix U containing the vectors $u_1, u_2, \ldots, u_k$ as columns. We have a geometric representation of G in the k-dimensional space R^k: we just take rows of U as point coordinates representing the vertices of G. Edges are straight line segments between the corresponding points. Now classical clustering methods (say k-means algorithm) should be applied to this new graph presentation. The complexity of this algorithm is polynomial [11].

The sum of squares of lengths of all edges in the representation of G is equal to $\nu_1 + \nu_2 + \cdots + \nu_k$. This is the minimal value over all representations obtained via matrix U with orthonormal columns. Such an extremal graph representation must have remarkable properties. It enhances the clustering properties of the original data and clusters can now be easily detected.

The number k of clusters can be determined by the *eigengap heuristic*:

Choose k such that first smallest k Laplacian eigenvalues are very small and the next one is relatively large.

A partial justification of this heuristic is based on the following statement:

If the graph has k components, then the multiplicity of eigenvalue 0 is k and there is a gap to the $(k+1)$-th eigenvalue (which is positive).

The obtained clustering partition of a short edge subgraph can be used to define several new spectral and structural complexity indices thus refining previous attempts.

In each of k clusters one can find a shortest open (partial) tour. The problem is to integrate these partial tours into a complete tour using edges between the clusters. The task reminds to that appearing in the well-known k-OPT heuristics. The complexity of finding an integral closed optimal tour using a brute force algorithm is equal to $2^{k-1}(k-1)!n_1!n_2!\cdots n_k!$ where $n_1, n_2, \ldots, n_k$ are cardinalities of clusters.

One should find complexity indices based on the number of edges in the selected short edge subgraph, the mentioned parameters of the cluster partition as well as on several parameters indicating the quality of the partition (see [11]). Using randomly generated instances one should check the efficiency of these indices similarly as in experiments before.

Experiments along these lines and further theoretical work are needed to verify these considerations.

Acknowledgments

Supported by the Serbian Ministry for Education and Science, Grants ON174033 and III44006. Presented at the International Conference on Operations Research, August 30 to September 2, 2011, Zurich, Switzerland.

References

[1] B. Ashok and T.K. Patra, Locating phase transitions in computationally hard problems, *J. Physics*, 2010, to appear.

[2] D. Cvetković, M. Čangalović and V. Kovačević-Vujčić, Semidefinite programming methods for the symmetric travelling salesman problem, *Integer Programming and Combinatorial Optimization, Proc. 7th Internat. IPCO Conf.*, Graz, Austria, June 1999, Lecture Notes Comp. Sci. 1610, Springer, Berlin, 1999, 126–136.

[3] D. Cvetković, M. Čangalović and V. Kovačević-Vujčić, Complexity indices for the travelling salesman problem based on a semidefinite relaxation, *SYM-OP-IS '99, Proc. XXVI Yugoslav Symp. Operations Research*, Beograd, 1999, 177–180.

[4] D. Cvetković, M. Čangalović and V. Kovačević-Vujčić, Optimization and highly informative graph invariants, *Two Topics in Mathematics, ed. B.Stanković, Zbornik radova 10(18)*, Matematički institut SANU, Beograd 2004, 5–39.

[5] D. Cvetković, P. Rowlinson and S. K. Simić, *An Introduction to the Theory of Graph Spectra*, Cambridge University Press, Cambridge, 2009.

[6] D. Cvetković and S.K. Simić, Graph spectra in computer science, *Linear Algebra Appl.*, **434**(2011), 1545–1562.

[7] M. Fiedler, Algebraic connectivity of graphs, *Czech. J. Math.*, **23(98)**(1973), 298–305.

[8] G. Gutin and A. Punnen, (Eds.), *The Travelling Salesman Problem and Its Variations*, Kluwer Academic Publishers, Dordrecht, 2002.

[9] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan and D.B. Shmoys, (Eds.), *The Traveling Salesman Problem*, John Wiley and Sons, Chichester - New York - Brisbane - Toronto - Singapore, 1985.

[10] K. Ko, P. Orponen, U. Schöning and O. Watanabe, What is a hard instance of a computational problem ?, *Structure in Complexity Theory. Conf. 1986*, Lecture Notes in Computer Science, Springer 1986, 197–217.

[11] U. von Luxburg, A tutorial on spectral clustering, *Stat. Comput.* **17**(2007), 395–416.

[12] C.R. Reeves and A.V. Eremeev, Statistical analysis of local search landscapes, *J. Oper. Res. Soc.*, **55**(2004), 687–693.

[13] B.D. Reyck and W. Herroelen, On the use of the complexity index as a measure of complexity in activity networks, *Europ. J. Oper. Res.*, **91**(1996), 347–366.

[14] R. Sawilla, *A survey of data mining of graphs using spectral graph theory*, Defence R&D Canada, Ottawa, Technical Memorandum TM 2008-317, Ottawa, 2008.

Dragoš Cvetković

Faculty of Electrical Engineering, University of Belgrade, and Mathematical Institute SANU, P.O. Box 367, 11000 Belgrade, Serbia

Email: ecvetkod@etf.rs