CycleExpander to *construct* Directed Hamiltonian Circuit
HexCycleSpanner to *tighten* Directed Hamiltonian Circuit

PIPR:©: Dr.(Prof.) Keshava Prasad Halemane,
Professor - retired from
Department of Mathematical And Computational Sciences
National Institute of Technology Karnataka, Surathkal
Srinivasnagar, Mangaluru - 575025, India.
Residence: 8-129/12 SASHESHA, Sowjanya Road, Naigara Hills,
Bikarnakatte, Kulshekar Post, Mangaluru-575005. Karnataka State, India.
KpH8MACS4KREC2NITK@gmail.com [+91*9481022946*]
https://www.linkedin.com/in/keshavaprasadahalemane/
https://colab.ws/researchers/R-3D34E-09884-MI42Z
https://github.com/KpH8MACS4KREC2NITK
https://orcid.org/0000-0003-3483-3521
https://osf.io/xftv8/

ABSTRACT

HexCycleSpanner (HCS) is a novel tool designed to achieve an elementary cycle refinement operation (ECRO) that is used to tighten a Directed Hamiltonian Circuit (dHC) - to be used iteratively as an extremely powerful technique to transform any given dHC to an optimum Shortest Directed Hamiltonian Circuit. Application to solving Asymmetric Travelling Salesman Problem (aTSP) is quite evident. ECRO uses a HCS to achieve an exchange of an outgoing arc-triplet of the given dHC with a corresponding incoming arc-triplet from outside the given dHC, thus resulting in a reconstructed transformed permuted dHC, while retaining the original direction/orientation of the given dHC. Effectively, the removal of the outgoing arc-triplet separates the given dHC into three cut-segments that are again rejoined by the incoming arc-triplet resulting in the reconstructed transformed permuted dHC wherein the original orientation of the dHC is retained - as defined by the orientation of the three cut-segments. This ECRO is analogous to the simplex pivot operation of Linear Programming wherein an exchange of the outgoing basic variable row with the incoming nonbasic variable column is performed to transform a given simplex tableaux representation into another equivalent simplex tableau representation that may be closer to the optimum tableaux.

The concept of HexCycleSpanner can be generalized and extended to include incoming directed paths rather than incoming arcs to give rise to what may be called a CycleExpander (CyclExp) that may be used as an effective tool to construct a Directed Hamiltonian Circuit.

The absence of a '*global effectiveness measure*' ('*gem*') for aTSP - unlike in the case of LP - requires one to use a reference data base of optimum/shortest directed path between pairs of nodes to enhance the computational efficiency of the proposed iterative ECRO algorithm.

Keywords: Optimization, Algorithm, Computational Complexity,
Asymmetric Travelling Salesman Problem, Shortest Directed Hamiltonian Circuit,
HexCycleSpanner, CycleExpander, Cyclic Permutation

AMS MSC Mathematics Subject Classification: 90C35
ACM CCS Computing Classification System: G.2.2.4

1. INTRODUCTION

The earliest avatars of Hamiltonian Path and Hamiltonian Circuit can be traced all the way back to the earliest times of human intellectual endeavors, referring to the classic idea of the Knight's tours on a chess board. Shortest (minimum weight) Hamiltonian Circuit (sHC) solves the Travelling Salesman Problem (TSP) by their very definition. TSP is one of the most widely studied combinatorial optimization problems, for which any improved solution algorithm is considered to be a critical achievement in terms of the computational complexity associated with such problems.

Hamiltonian paths and Hamiltonian circuits have been extensively studied and reported because of its wide spread applications in various practical problem domains. Although various reports can be found in literature about the necessary conditions for Hamiltonicity and similarly on the sufficient conditions, there has not been any report of a strong/tight necessary & sufficient condition thereof. A necessary condition for a connected graph G(V,E) with $|V| = n \geq 3$ vertices to be Hamiltonian is established by Dirac's [D52] theorem which states that the minimum vertex degree be at least equal to n/2 that is half the number of vertices n. However, this turns out to be very weak necessary condition. Chvatal & Erdos [CE72] gives sufficient conditions for Hamilton Path and/or Hamiltonian Circuit, using the concepts of stability number (maximum stable set or maximum independent set), connectivity and bridges; later generalized by Kouider [K94]. A theorem of Bondy & Chvatal [BC76] gives the necessary & sufficient condition for Hamiltonicity of a simple graph, using the concept of the closure of a graph, although its utility seems to be somewhat limited except for theoretical analyses. Lin & Kernighan [LK73] developed a heuristic approach known as the 2-opt move based on twin-arc-exchange as was originally proposed by Flood [F56] to improve a given Hamiltonian Circuit. Gould [G91] [G03] [G14] gives an excellent survey of the developments in terms of the Hamiltonian problem on undirected graphs.

In the case of directed graphs, Bondy & Murty [BM08] mentions that Redei [R34] showed, by induction arguments on the number of vertices, that every tournament has a directed Hamiltonian Path. But every tournament need not be Hamiltonian, as for example a transitive tournament. Camio (1959) proved that every nontrivial strong tournament has a directed Hamiltonian Circuit. Chen & Manalastas [CM83] showed that every finite strongly connected digraph of stability 2 has a Hamiltonian path; for which Bondy [B95] gives a short and elegant proof. Using the concept of directed ear decomposition Knuth [K74] shows that every strong digraph admits a coherent feedback arc set. Bessy & Thomasse [BT03] [BT04] showed that every stong digraph admits a coherent cyclic order. Sebo [S07], Iwata & Matsuda [IM07] report results related to the existence of coherent cyclic order in strong digraphs.

However there has neither been any report regarding any tight necessary & sufficient conditions for the existence of Hamiltonicity nor has there been any computationally efficient algorithms reported for determining Hamiltonicity in case of general networks (directed graphs/multigraphs). Zhang [Z00] [CJMZ00] mentions that Kannellakis-Papadimitriou [KP80] arc-exchange update for asymmetric TSP adapted from Lin-Kernighan [LK73] edge exchange strategy for symmetric TSP seems to be a promising heuristic approach for dHC. Ban-Jensen & Gutin [BG09] [BG18] gives an excellent exposition of the developments in the algorithms for directed graphs. Gutin & Punnen [GP04] presents the details on the various developments in solving aTSP.

Here we will see how an elementary cycle refinement operation (ECRO) is developed as a novel method to tighten a given Directed Hamiltonian Circuit (dHC) using a HexCycleSpanner (HCS) designed as a tool for the purpose - to be used iteratively as a computationally efficient algorithm

to determine the shortest directed Hamiltonian circuit and thus solve the asymmetric travelling salesman problem (aTSP). Also explained herein is the novel concept of a CycleExpander (CyclExp) that can be applied in the construction of a dHC.

2. HEX-CYCLE-SPANNER

A *HexCycleSpanner* (HCS) designed as a tool for the purpose of defining an *elementary cycle refinement operation* (ECRO) that transforms a given *Directed Hamiltonian Circuit* (dHC) into another one with improved properties - for example with reduced weight / length - is a 6-cycle (HCS is not a directed circuit) with six arcs and having some special characteristic features : it forms a cycle of six directed arcs, three alternating arcs oriented in one way and belonging to the given dHC whereas the other three alternating arcs directed in the opposite sense and belonging outside of the given dHC. An ECRO uses a HCS to achieve an exchange of judiciously chosen outgoing arc-triplet with a corresponding incoming arc-triplet, resulting in a reconstructed transformed permuted dHC. The HCS effectively cuts the given dHC into three cut-segments by removing outgoing arc-triplet and rejoins the three cut-segments using the incoming arc-triplet - in such a way that the resulting reconstructed transformed permuted dHC has exactly the same orientation as the original orientation of the given dHC - the original cycle orientation being defined by the orientation of the three cut-segments of the given dHC. Note that one canNOT achieve such an *orientation-retaining transformation* by cutting the given dHC into two pieces - three cut-segments is the minimum requirement to retain the original orientation in the transformed dHC. Thus an ECRO achieves an exchange between the outgoing arc-triplet and the incoming arc-triplet - this arc-triplet-pair forms the required HexCycleSpanner - that is a 6-cycle wherein the outgoing & incoming arcs are in alternating positions and in opposing orientation with respect to one another.

3. STANDARD TEMPLATE FOR A GENERIC HEX-CYCLE-SPANNER

Effective application of ECRO requires an efficient method of finding an appropriate HCS from among the various possible choices. To facilitate the search for an appropriate HCS it is essential to develop a standard *template* for a generic HCS. For this purpose, consider a 9-cycle dHC (1,2,3,4,5,6,7,8,9) with outgoing arc-triplet (9,1) (3,4) & (6,7) and incoming arc-triplet (9,4) (6,1) & (3,7) so that the given dHC gets cut into three cut-segments (1,2,3) (4,5,6) & (7,8,9) by the removal of the three outgoing arcs and these three cut-segments get rejoined by the three incoming arcs, resulting in a reconstructed transformed permuted 9-cycle dHC (1,2,3,7,8,9,4,5,6) wherein the original orientation of the given 9-cycle dHC is retained - as defined by the orientation of the three cut-segments (1,2,3) (4,5,6) & (7,8,9).

Here, the HCS is the 6-cycle (9,1,6,7,3,4) - wherein the three outgoing arcs and the three incoming arcs are positioned mutually adjacent to one another - each outgoing arc is flanked by two incoming arcs on either end and each incoming arc is flanked by two outgoing arc on either end - the incoming arc-triplet oriented in the opposite sense relative to the orientation of the outgoing arc-triplet. Each *outgoing arc* of a given dHC has a *one-to-one correspondence* with a *directed path* from its tail-end to its head-end, containing the corresponding cut-segment of the given dHC flanked on either end by the same two incoming arcs that are adjacent to that outgoing arc, thus defining the cut-segment associated with this outgoing arc - this is an essential characteristic of a HCS.

Note that each of the interior nodes of these three cut-segments - represented by the nodes 2, 5, 8 of the given dHC - can in fact represent a directed path joining the end nodes of the corresponding cut-segment, so that this template becomes a generic template applicable to any given dHC in general.

As said earlier, an outgoing (*og91Arc*) arc *o91* is flanked by two incoming arcs, one is an incoming (*tail94Link*) arc *i94* incident on the tail-end of *og91Arc o91* and points to the starting node of a cut-segment *cs456* whereas the other is an incoming (*head61Link*) arc *i61* incident on the head-end of *og91Arc o91* and links to the terminal node of this cut-segment *cs456* - forming a directed path *i94cs456i61* from the tail-end of *og91Arc o91* to the head-end of the *o91* passing through (1) *tail94Link i94* (2) cut-segment *cs456* (3) *head61Link i61* - establishing a one-to-one correspondence between the outgoing arc *o91* and the cut-segment *cs456* - this cut-segment itself being flanked by these two incoming arcs, one being the *tail94Link i94* and the other being the *head61Link i61* forming the directed path *i94cs456i61*. Observe similar association in case of the second outgoing arc *og34Arc o34* with the *tail37Link* i37 the cut-segment cs789 and the *head94Link i94* to form the directed path *i37cs789i94* and also in case of the third outgoing arc *og67Arc o67* with the *tail61Link i61* the cut-segment cs123 and the *head37Link i37* to form the directed path *i61cs123i37*. It is essential to note that the role of an incoming arc being either a *tail-Link* or a *head-Link* is relative to the specific outgoing arc *og-Arc* and not a permanent / static role. Also note that *each of the three outgoing arcs get flanked by the very same two incoming arcs that flank the corresponding cut-segment to form the directed path from its tail-end to its head-end* - this is an essential characteristic of a HCS.

4. FINDING A HEX-CYCLE-SPANNER FOR A dHC

Figure-1 gives the node arc incidence matrix for a generic 9-cycle dHC. Find an outgoing arc j1=(i9,i1) which is flanked on each end by two incoming arcs, j10=(i9,i4) at the tail-end and j11=(i6,i1) at the head-end, both being oriented in the opposite sense to that of this outgoing arc, and the farther end i4 & i6 of these two incoming arcs define a cut-segment (i4,i5,i6) of the given dHC that has a one-to-one correspondence with the outgoing arc (i9,i1); the ends of this cut-segment being adjacent to another two outgoing arcs, j4=(i3,i4) and j7=(i6,i7) that are joined by an incoming arc j12=(i3,i7). This set of three outgoing arcs (i9,i1) (i3,i4) & (i6,i7) along with the corresponding three incoming arcs (i9,i4) (i6,i1) & (i3,i7) form a 6-cycle HCS (i9,i1,i6,i7,i3,i4). Of course, for each possible choice of an outgoing arc, there may be several choices of incoming arcs and therefore several possible choices of HCS that may be found, and the one that best satisfies the requirements, say for example reducing the overall cost/weight/length of the dHC needs to be chosen for performing the desired elementary cycle refinement operation. This process of selecting the matching pairs of outgoing & incoming arc-triplets is analogous to the selection of a matching pair of pivot row (outgoing basic variable) & pivot column (incoming nonbasic variable) in simplex pivot selection for linear programming problems.

Note that the evaluation of a potential candidate (i9,i1) as an outgoing arc includes the confirmation of the existence of corresponding adjacent arcs (i9,i4) & (i6,i1) which are themselves adjacent to two outgoing arcs (i3,i4) & (i6,i7) and also an incoming arc (i3,i7) such that the given dHC gets cut into three segments (i1,i2,i3) (i4,i5,i6) & (i7,i8,i9) by the removal of the outgoing arc-triplet (i9,i1) (i3,i4) & (i6,i7) and these three segments again get rejoined by the incoming arc-triplet (i9,i4) (i6,i1) & (i3,i7) thus resulting in the reconstructed transformed permuted dHC (i1,i2,i3,i7,i8,i9,i4,i5,i6). In this case, the effective reduction in the overall cost/weight/length indicated by (6w1 + 3w7 + 9w4) < (9w1 + 3w4 + 6w7) justifies the ECRO

defined by this exchange of the outgoing arc-triplet (i9,i1) (i3,i4) & (i6,i7) with the incoming arc-triplet (i6,i1) (i3,i7) & (i9,i4) as shown in Figure-2.

↓	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10	j11	j12
i1	−	+									−	
i2		−	+									
i3			−	+								+
i4				−	+					−		
i5					−	+						
i6						−	+				+	
i7							−	+				−
i8								−	+			
i9	+								−	+		

Figure-1: Node Arc incidence matrix with cycle orientation indicated by ordering of nodes

↓	i1	i2	i3	i4	i5	i6	i7	i8	i9
i1	+	1w2							
i2		+	2w3						
i3			+	3w4			3w7		
i4				+	4w5				
i5					+	5w6			
i6	6w1					+	6w7		
i7							+	7w8	
i8								+	8w9
i9	9w1			9w4					+

Figure-2: Node Node Adjacency Matrix with Weights

It may be worth to note a word of caution here. In the process of finding a matching pair of an outgoing arc-triplet with an incoming arc-triplet, if one is not careful to keep a watch on the relative positioning of the three outgoing arcs and the corresponding three incoming arcs in relation to the orientation of the given dHC, it is easy to get into a situation where the three cut-segments when rejoined to yield a reconstructed transformed permuted dHC may possibly end up being three mutually arc-disjoint cycles rather than a single dHC. For example, in the above case of a given dHC with a 9-cycle, if the three outgoing arcs (i9,i1) (i3,i4) & (i6,i7) are matched with the correspond incoming arcs (i9,i7) (i3,i1) & (i6,i4) the result is three mutually arc-disjoint 3-cycles (i1,i2,i3) (i4,i5,i6) & (i7,i8,i9) which is an unintended result.

To overcome this problem, or rather to prevent such an unacceptable situation, it is recommended to emphasize the fact that each of the three *outgoing arc*s, flanked in-between two incoming arcs, has a *one-to-one correspondence* with a *directed path* from its tail-end to its head-end, wherein the first and the last arcs are the two incoming arcs flanking that outgoing arc and the thus flanked segment of the directed path is indeed the *cut-segment* of the dHC corresponding to this outgoing arc. The outgoing arc (i9,i1) associated with the directed path (i9,(i4,i5,i6),i1) which holds the

segment (i4,i5,i6) flanked by the very same two end arcs (i9,i4) & (i6,i1) that are flanking the outgoing arc (i9,i1). This is an essential characteristic of a HCS.

Now we can see that the earlier situation with the outgoing arc (i9,i1) flanked by the incoming arcs (i9,i7) and (i3,i1) defines a cut-segment (i3,i4,i5,i6,i7) which along with these two incoming arcs flanking this outgoing arc doesn't result in a directed path from the tail-end i9 to the head-end i1; whereas this cut-segment again gets cut by the removal of its two end arcs (i3,i4) & (i6,i7) as outgoing arcs thus shrinking the cut-segment to (i4,i5,i6) while having no linking arcs between this cut-segment and the other two cut-segments (i1,i2,i3) & (i7,i8,i9) thus resulting in three mutually arc-disjoint cycles.

5. CYCLIC PERMUTATION

In addition to bringing about an exchange between the outgoing arc-triplet and the incoming arc-triplet, the above defined ECRO using HCS can in fact be considered as the basic / elementary swapping operation in the process of generating cyclic permutations of dHC, wherein each ECRO brings about a single swap/exchange of the two of the three selected cut-segments w.r.t. the other in the cyclic order - as in the earlier example, transforming the given dHC (1,2,3,4,5,6,7,8,9) to the reconstructed transformed permuted dHC (1,2,3,7,8,9,4,5,6) wherein the two cut-segments, say (4,5,6) and (7,8,9) get swapped w.r.t. the third cut-segment that is (1,2,3).

It is advised to keep track of the information associated with each of the potential candidate outgoing arc, as to the arc-triplet pair defining the HCS involved in the exchange as well as the change in the cost/weight/length associated with the corresponding ECRO. This will facilitate the determination of the best choice of the HCS and hence the best ECRO for the given dHC under consideration.

6. CYCLE-EXPANDER

The concept of HexCycleSpanner can be generalized and extended to define a CycleExpander (CyclExp) by allowing incoming directed paths instead of incoming arcs in the definition of a HexCycleSpanner. This will enable a CyclExp to be used as an effective tool in expanding a given directed circuit to include more and more nodes so as to result in the construction of a Directed Hamiltonian Circuit.

7. CYCLE-EXPANDER AND HEX-CYCLE-SPANNER TO SOLVE ATSP

The above concepts of CycleExpander (CyclExp) and HexCycleSpanner (HCS) can directly be applied in the construction of a dHC and on which ECRO can be performed using HCS in achieving further improvements in the dHC to determine the optimum/shortest directed Hamiltonian circuit thus solving the Asymmetric Travelling Salesman Problem - even for general networks / directed multigraphs.

Although ECRO using HCS is analogous to simplex pivoting operation of linear programming there is no '*global effective measure*' ('*gem*') defined for aTSP - unlike in the case for linear programming for which the 'spdspds' [H22] algorithm exploits the '*gem*' defined by the '*infeasibility index*' to design a computationally efficient algorithm. In the case of aTSP the best

that may be done if at all is to maintain a reference data base of 'optimum/*shortest paths*' between pairs of nodes, which can be used effectively to retain such optimum path segments in a dHC when choosing the best possible HCS for the application of ECRO.

We recommend that the Painted Network Algorithm of Rockafellar [R98] be adapted for path finding in the application of CyclExp and HCS to develop a computationally efficient solution strategy for aTSP. Appropriate fine-tuning of the algorithm with the use of CyclExp and HCS for the purpose of solving aTSP can result in extraordinary gains in computational efficiency.

We recommend that one initiates a directed ear decomposition [BG09] of the given network (directed graph / multigraph) using a depth first search on minimum cost outgoing arc at every node and to choose - either the *primary ear circuit* defined here as the one wherein every arc corresponds to the optimum (minimum cost/weight/length) outgoing arc from its initial node - or the *leap ear circuit* defined here as the one obtained by *conjoining* the primary ear circuit with other suitable ones through a *segment exchange operation* to form the maximal cardinality (number of nodes) ear while possibly compromising on the cost/weight/length associated with the arcs at the conjoining nodes - as a good starting point to apply the CyclExp for constructing a dHC which can then be further refined / improved through ECRO by the application of HCS.

One may choose the best possible HCS that will potentially result in the best possible reduction in the cost / weight / length as a result of the application of the corresponding ECRO. For this purpose, it may be useful to keep track of the data associated with every potential choice of HCS defined by the corresponding pair of arc-triplets, and choose the best among them that result in the best possible improvement in the reconstructed / transformed / permuted dHC.

Alternatively, one may consider some pre-processing using the concept of bead-bridge decomposition into strong components may provide useful information to possibly enhance the computational efficiency by working on the components and then integrating the results later.

Also, some appropriate shortest path algorithm (e.g. Floyd-Warshall) may be applied on the given network, initially, in order to develop the required reference data base of min-cost directed path segments between node pairs, which will later facilitate in the decision regarding the best possible choice for a HCS. If any sub-sequence of any of these paths is a matching cycle-segment of the dHC under investigation, then such a cycle-segment cannot be subjected to any further cuts by application of ECRO using any HCS, because it corresponds to the best possible choice in terms of minimum-cost / least-weight / shortest-length sub-sequence / cycle-segment of the dHC and hence no further improvement can possibly be achieved through the application of any ECRO using whatever available HCS.

Alternatively, one may even explore the possibility of starting with the optimum arborescence along with the corresponding root-location problem mentioned by Fujishige & Kamiyama [FK12] [K14] instead of the Floyd-Warshall type approach, before the actual application of the CyclExp to construct a dHC and then proceed with the application of ECRO with HCS.

8. CONCLUSION

The ECRO operation and the associated tools CyclExp & HCS facilitate the design of algorithms for constructing dHC for a given network (directed graph / multigraph) and also to refine a given dHC to determine the optimum (minimum-cost / least-weight / shortest-length) dHC and hence solve aTSP. ECRO using CyclExp & HCS is flexible to adopt various levels of fine-tuning that may be required, irrespective of the details of whichever algorithm is used, in order to address various possible requirements including one of computational complexity.

9. REFERENCES

[BG09] Jorgen Ban-Jensen & Gregory Gutin (2009); "Digraphs : Theory, Algorithms and Applications"; Second Edition; Springer Monographs in Mathematics; 2009.

[BG18] Jorgen Ban-Jensen & Gregory Gutin (2018); "Classes of Directed Graphs"; Springer Monographs in Mathematics; 2018.

[BT03] Bessy S. & Thomasse S. (2003) : "Every strong digraph has a spanning strong subgraph with at most n+2α−2 arcs"; J. Combin. Theory Ser. B 87; pp 289–299; 2003.

[BT04] Bessy S. & Thomasse S. (2004) : "Three min-max theorems concerning cyclic orders of strong digraphs"; in Integer Programming and Combinatorial Optimization; pp 132–138; Lecture Notes in Comput. Sci.; Vol. 3064; Springer, Berlin; 2004.

[BM08] Bondy J.A. & Murty U.S.R. (2008) : "Graph Theory" (GTM244); Springer; 2008.

[BC76] Bondy J.A. & Chvatal V. (1976) : "A method in Graph Theory"; Discrete Math. 15; pp 111-135; 1976.

[B95] Bondy J.A. (1995); "A short proof of the Chen-Manalastas theorem"; Discrete Mathematics 146; pp 289-292.

[C59] Camio P. (1959): "Chemins et circuits hamiltoniens des graphes complets"; C. R. Acad. Sci. Paris; 249; pp 2151-2152; 1959.

[CM83] Chen C.C. & Manalastas P.Jr. (1983); "Every finite strongly connected digraph of stability 2 has a Hamiltonian path"; Discrete Math. 44; pp 243-250.

[CJMZ00] Cirasella, J., Johnson, D.S., McGeoch, L.A., & Zhang, W. (2000); "The asymmetric Traveling Salesman Problem: Algorithms, instance generators, and tests"; Proc. of ALENEX 2001.

[CE72] Chvatal V. & Erdos P. (1972) : "A Note on Hamiltonian Circuits"; Discrete Math. 2; pp 111-113.

[C12] Cook,W.J. (2012); "In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation"; Princeton University Press, Princeton, 2012.

[D52] Dirac G.A. (1952) : "Some theorems on abstract graphs"; Proc. London Math. Soc. 3(2); pp 69-81; 1952.

[FK12] Fujishige, S. & KamiyamA,N. (2012); "The root location problem for arc-disjoint arborescences"; Discrete Applied Mathematics 160 (2012) pp. 1964-1970.

[G91] Gould R.J. (1991); "Updating the Hamiltonian Problem: A Survey"; Journal of Graph Theory; Vol. 15; No. 2; pp 121-157.

[G03] Gould R.J. (2003); "Advances on the Hamiltonian Problem: A Survey"; Graphs Combin. 19; pp 7-52.

[G06] Gould R.J. & Zhao K. (2006); "A new sufficient condition for Hamiltonian graphs"; Ark. Mat.; 44; pp 299-308.

[G14] Gould R.J. (2014); "Recent Advances on the Hamiltonian Problem: Survey-III"; Graphs and Combinatorics; 30: pp 1-46.

[GP04] Gutin,G., Punnen, A.P. (2004); "The Traveling Salesman Problem and its Variants"; Kluwer Academic Publishers.

[H22] Halemane, K.P. (2022); "Unbelievable $O(L^{1.5})$ worst case computational complexity achieved by *spdspds* algorithm for linear programming problem"; https://arxiv.org/abs/1405.6902

[HK70] Held M. & Karp R.M. (1970); "The traveling salesman problem and minimum spanning trees"; Operations Research; 18; pp. 1138-1162.

[HK71] Held M. & Karp R.M. (1971); "The traveling-salesman problem and minimum spanning trees: Part II"; Mathematical Programming; **1**, 6–25.

[IM07/08] Iwata S. & Matsuda T. (2007/2008) : "Finding coherent cyclic orders in strong digraphs"; Technical report, Graduate School of Information Sciences and Technology, University of Tokyo; 2007; Combinatorica; 28; 83; 2008.

[K14] Kamiyama, N. (2014); "Arborescence Problems in Directed Graphs: Theorems and Algorithms"; Interdisciplinary Information Sciences Vol. 20, No. 1 (2014) pp.51-70.

[KP80] Kanellakis, Paris-C. & Papadimitriou, Christos H. (1980);
"Local Search for the Asymmetric Traveling Salesman Problem";
Op. Res. J. Vol. 28, No. 5 (Sep. - Oct., 1980), pp. 1086-1099 (14 pages)
Published by INFORMS.

[K74] Knuth D.E. (1974); "Wheels within wheels";
J. Combin. Theory Ser. B 16; pp 42–46.

[K94] Kouider M. (1994):
"Cycles in Graphs with prescribed stability and connectivity";
J. Combin. Theory Ser. B 60; pp 315-318.

[LK73] Lin, S. & Kernighan, K,W. (1973);
"An Effective Heuristic Algorithm for Traveling Salesman Problem";
Operations Research, Vol. 21, No. 2 (Mar. - Apr., 1973), pp. 498-516.

[FM56] Flood, Merrill M. (1956); "The Traveling-Salesman Problem";
doi.org/10.1287/opre.4.1.61 Op.Res.J. Vol.4 No.1 pp.61-75.

[R34] Redei L. (1934); "Ein kombinatorischer Satz";
Acta. Litt. Sci. Szeged 7; pp 39–43.

[R98] R. T. Rockafellar (1998); "Network Flows and Monotropic Optimization";
Athena Scietific.

[S07] Sebo A. (2007) : "Minmax relations for cyclically ordered digraphs";
J. Combin. Theory Ser. B 97; pp 518–552; 2007.

[Z00] Zhang, W. (2000);
"Depth-First Branch and Bound vs Local Search : A Case Study";
Proc. 17-th National Conf. on Artificial Intelligence (AAAI-2000),
Austin, Texas, July 30-August 3, 2000, pp.930-935.

10. DEDICATION

To my ಅಜ್ಜ (ajja) & ಅಜ್ಜಿ (ajji) & Karinja Halemane Keshava Bhat & Thirumaleshwari and to my ಅಪ್ಪ (appa) & ಅಮ್ಮ (amma) Shama Bhat & Thirumaleshwari for their teachings through love, that quality matters more than quantity; to my wife Vijayalakshmi for her ever consistent love & support; to my daughter Sriwidya.Bharati and my twin sons Sriwidya.Ramana & Sriwidya.Prawina for their love & affection.