Journal of Theoretical and Applied Information Technology
30th September 2011. Vol. 31 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MELAKARTHA RAGA GENERATION THROUGH BREADTH FIRST SEARCH ALGORITHM

[1]B.SANTHI, [2]N.SAIRAM

[1,2] Professor., School of Computing, SASTRA UNIVERSITY

E-mail: [1]shanthi@cse.sastra.edu, [2]sairam@cse.sastra.edu

ABSTRACT

Raga collections are mostly used in the music information systems. Melakartha ragas are the most important concept in the music theory. All the existing ragas in music theory are derived from mela kartha ragas. The main objective of this paper is to generate all 72 ragas automatically. This paper identifies an application of graph theory in music information technology. An undirected graph is constructed using the melakartha algorithm with seven swaras and its variations as its vertices. By applying the Breadth First Search algorithm, to the constructed graph, we generated all possible paths that represent the 72 Melakartha ragas.

Keywords: *Melakartha Raga, Breadth First Search, Swaras*

1. INTRODUCTION

The Melekartha System is the most scientific system of creation and classification of ragas and forms the basis of the classical carnatic music system. Melakartha system uses a powerful scientific algorithm to create the fundamental 72 ragas of the carnatic music system. All other ragas in the carnatic music system are derived from one/more of these 72 ragas. The derived ragas are called janya ragas. Section 2 describes the twelve keys of an octave (key board) in the melakartha system

In section 3 melakartha algorithm is described. Section 4 describes briefly the Breadth First Search algorithm for finding all possible paths in a graph. In Section 6 results derived from our algorithm are shown. Concluding remarks and future directions are discussed in section 7.

2. MELAKARTHA SYSTEM

The Melakartha system divides the 12 keys of an octave (in the keyboard) into seven swaras as follows:

2.1. Shadjama

S = Shadjama = First key of the octave

2.2. Rishaba

R1 = Shudhdha Rishabha = Second key of the octave

R2 = Chathushruthi Rishabha = Third key of the octave

R3 = Sathshruthi Rishabha = Fourth key of the octave

2.3. Gandhara

G1 = Shudhdha Gandhara = Third key of the octave

G2 = Sadharana Gandhara = Fourth key of the octave

G3 = Anthara Gandhara= Fifth key of the octave

2.4. Madhyama

M1 = Shudhdha Madhyama = Sixth key of the octave

M2 = Prathi Madhyama = Seventh key of the octave

2.5. Panchama

P = Panchama = Eight key of the octave

2.6. Daivatha

D1 = Shudhdha Daivatha = Nineth key of the octave

D2 = Chathushruthi Daivatha = Tenth key of the octave

D3 = Sathshruthi Daivatha = Eleventh key of the octave

2.7. Nishadha

N1 = Shudhdha Nishadha = Tenth key of the octave

N2 = Kaishika Nishadha = Eleventh key of the octave

N3 = Kaakali Nishadha = Twelveth key of the octave

Journal of Theoretical and Applied Information Technology
30th September 2011. Vol. 31 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. THE MELAKARTHA ALGORITHM:

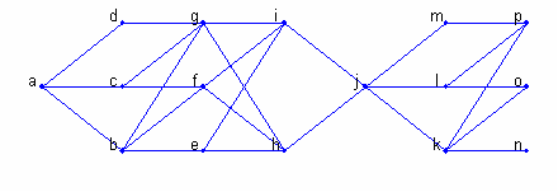
Step (i): A Melakartha Raga has all the 7 swaras in it.
Step (ii): A Melakartha Raga cannot have multiple entries for the same swara.
Step (iii): The same frequency cannot occupy more than one swara,
Step (iv): The swara order (increase in frequency called ArOhaNa) is Sa(Shadjama),Ri (Rishabha),Ga (Gandhara),Ma (Madhyama),Pa (Panchama),Dha
(Daivatha),Ni (Nishadfha).
Based on the above algorithm we can form the Melakartha ragas as follows.
• Sa and Pa are the same in all ragas as they have only one entry.
• Ma has two entries so we can form two ragas for every given combination of all other swaras i.e. if we are given a sa,ri,ga,pa,da and ni combination
• R1 has 3 entries and Ga has 3 entries where two entries are common to both (i.e.R2=G1 and R3=G2). Hence, considering the fact that frequency of Ga has to be always greater than Ri (i.e. the key for Ri cannot come after the key for Ga) ,we can have totally 6 combinations for Ri,Ga as follows:
R1G1, R1G2, R1G3, R2G2, R2G3 and R3G3
• Similarly we can have 6 entries for Dha,Ni combination as follows:
D1N1, D1N2, D1N3, D2N2, D2N3 and D3N3
Thus we have unique 2 entries for Ma, 6 entries for Ri,Ga and 6 entries for Ni,Dha. This makes a total of 2x6x6 = 72 unique combinations of all the 7 swaras defined based on 12 keys in an octave.
Thus using the Melakartha algorithm we can create 72 Melakartha ragas. A melakartha raga is also called as a Sampoorna raga as it has all the 7 swaras in it.

4. ALGORITHM FOR FINDING ALL POSSIBLE COMBINATIONS FROM SHADJAMA TO NISHADHA

Step(i) Consider Shadjama as a root node at level zero
Step(ii) Level one nodes are shudhdha Rishabha, Chathushruthi Rishaba and
Sathshruthi Rishaba
Step(iii) Consider Shudhdha Gandhara, Sadharana Gandhara, Anthara Gandhara as level two nodes
Step(iv) Shudhdha Madhyama, Prathi Madhyama are considered as level 3 nodes
Step(v) Panchama is considered as a level five node.
Step(vi) shudhdha Daivatha, Chathushruthi Daivatha and Sathshruthi Daivatha are considered as nodes in level six.
Step(vii) The leaf nodes shudhadha Nishadha, Kaishika Nishadha and Kaakali
Nishadha.
Step(viii) Traversing the above constructed tree using Breadth First Search algorithm
we get all the seventy two distinct melakartha ragas.

5. RESULTS

The seven swaras of the melakartha system is depicted in the following graph.



a = SA; b = RE1 ; c = RE2 ; d = RE3 ; e = GA1 ; f = GA2 ; g = GA3 ; h = MA1 ;
i = MA2; j = PA ; k = DA1 ; l = DA2 ; m = DA3; n = NI1 ; o = NI2 ; p = NI3

• Starting with Shadjama and end with
N1 =shudhdha Nishadha creates 12 unique ragas
• Starting with Shadjama and end with
N2 = Kaishika Nishadha creates 24 unique ragas
• Starting with Shadjama and end with
N3 = Kaakali Nishadha creates 36 unique ragas.
Thus our algorithm generates all the 72 unique sampoorna ragas from 7 swaras

The output of the program is given in the following table.

NI1 - Shudhdha Nishadha
NI2 - Kaishika Nishadha
NI3 - Kaakali Nishadha

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	NI1 - Shudhdha Nishadha	NI2 - Kaishika Nishadha	NI3 - Kaakali Nishadha
SA	1. SARE3GA3MA2PADA1NI1	13. SARE3GA3MA2PADA1NI2	37. SARE3GA3MA2PADA1NI3
	2. SARE3GA3MA1PADA1NI1	14. SARE3GA3MA2PADA2NI2	38. SARE3GA3MA2PADA2NI3
	3. SARE2GA3MA2PADA1NI1	15. SARE3GA3MA1PADA1NI2	39. SARE3GA3MA2PADA3NI3
	4. SARE2GA3MA1PADA1NI1	16. SARE3GA3MA1PADA2NI2	40. SARE3GA3MA1PADA1NI3
	5. SARE2GA1MA2PADA1NI1	17. SARE2GA3MA2PADA1NI2	41. SARE3GA3MA1PADA2NI3
	6. SARE2GA1MA1PADA1NI1	18. SARE2GA3MA2PADA2NI2	42. SARE3GA3MA1PADA3NI3
	7. SARE1GA3MA2PADA1NI1	19. SARE2GA3MA1PADA1NI2	43. SARE2GA3MA2PADA1NI3
	8. SARE1GA3MA1PADA1NI1	20. SARE2GA3MA1PADA2NI2	44. SARE2GA3MA2PADA2NI3
	9. SARE1GA2MA2PADA1NI1	21. SARE2GA1MA2PADA1NI2	45. SARE2GA3MA2PADA3NI3
	10. SARE1GA2MA1PADA1NI1	22. SARE2GA1MA2PADA2NI2	46. SARE2GA3MA1PADA1NI3
	11. SARE1GA1MA2PADA1NI1	23. SARE2GA1MA1PADA1NI2	47. SARE2GA3MA1PADA2NI3
	12. SARE1GA1MA1PADA1NI1	24. SARE2GA1MA1PADA2NI2	48. SARE2GA3MA1PADA3NI3
		25. SARE1GA3MA2PADA1NI2	49. SARE2GA1MA2PADA1NI3
		26. SARE1GA3MA2PADA2NI2	50. SARE2GA1MA2PADA2NI3
		27. SARE1GA3MA1PADA1NI2	51. SARE2GA1MA2PADA3NI3
		28. SARE1GA3MA1PADA2NI2	52. SARE2GA1MA1PADA1NI3
		29. SARE1GA2MA2PADA1NI2	53. SARE2GA1MA1PADA2NI3
		30. SARE1GA2MA2PADA2NI2	54. SARE2GA1MA1PADA3NI3
		31. SARE1GA2MA1PADA1NI2	55. SARE1GA3MA2PADA1NI3
		32. SARE1GA2MA1PADA2NI2	56. SARE1GA3MA2PADA2NI3
		33. SARE1GA1MA2PADA1NI2	57. SARE1GA3MA2PADA3NI3
		34. SARE1GA1MA2PADA2NI2	58. SARE1GA3MA1PADA1NI3
		35. SARE1GA1MA1PADA1NI2	59. SARE1GA3MA1PADA2NI3
		36. SARE1GA1MA1PADA2NI2	60. SARE1GA3MA1PADA3NI3
			61. SARE1GA2MA2PADA1NI3
			62. SARE1GA2MA2PADA2NI3
			63. SARE1GA2MA2PADA3NI3
			64. SARE1GA2MA1PADA1NI3
			65. SARE1GA2MA1PADA2NI3
			66. SARE1GA2MA1PADA3NI3
			67. SARE1GA1MA2PADA1NI3
			68. SARE1GA1MA2PADA2NI3
			69. SARE1GA1MA2PADA3NI3
			70. SARE1GA1MA1PADA1NI3
			71. SARE1GA1MA1PADA2NI3
			72. SARE1GA1MA1PADA3NI3

6. CONCLUSION AND FUTURE DIRECTIONS

We have shown that starting from shadjama and ending with nishadham forms a Raga. Swaras are given as input to the Breadth first search algorithm, to find the all possible paths. These distinct paths are representation of ragas. Thus we have proposed an efficient algorithm for generating 72 main ragas based on the melakartha rules. This work can be extended to identify unknown ragas through pattern matching. The efficiency of this method can be improved by using the parallel breadth first search algorithm.

REFERENCES:

[1]. Jean-Paul Tremblay , Paul G. Sorenson, An Introduction to Data Structures With Applications, McGraw Hill Computer Science Series, second edition, 1984.

[2]. Narasing Deo, Graph Theory with Application to Engineering and **Computer Science,** Prentice-Hall, Englewood Cliffs, N.J., 1974.

[3]. Nicos Christofides, Graph Theory. An algorithmic Approach, Academic press, 1975.

[4]. www.carnaticindia.com.

[5]. S. Bhagyalekshmy, Ragas in Carnatic music, CBH Publications, 1990.

[6]. P Sambamoorthy, *South Indian Music* Book III, The Indian Music Publishing House, 1973.