A Search Algorithm based on Diatonic Scale for College Course Schedule

Dr B V Subba Rao[#1], J.Sirisha*[2]

[#]*Professor, dept of IT, PVP Siddhartha Institute of Technology, Vijayawada, India*
[*]*Asst.Professor, dept of IT, PVP Siddhartha Institute of Technology, Vijayawada, India*

Abstract— **This paper displays those ponder for a change of a chart coloring based nearby look calculation for the school course timetabling issue. A few models Furthermore methodologies will determine the issue is talked about. The fundamental thought of the approach may be through An heulandites algorithm to define the chromatic classes of the graph; then they ought make sort program as stated by particular sad for criteria (color index, downright weight, and the aggregate number from claiming occasions in every class), Furthermore At long last should run the calculation for nearby scan. The conditions, point and the comes about of the test need aid exhibited.**

Keywords— **School, college, color index, chromatic class**.

I. INTRODUCTION

Those school course plan [CCS] may be a combinatorial issue. To the to start with the long haul it might have been introduced clinched alongside [1]. CCS may be NP-hard issue [2], in any case it need an incredible useful relevance, for example [3][4]. It need been indicated tentatively that the heulandites methodologies provide for preferred comes about over the opposite methodologies for huge data information [5][6][7].

At present a quick algorithm for solving the problem has not been found yet. Though being accurate, the methods based on the approaches "backtracking" and "brute force" (and their variants with decreasing recursion tree among tested partial and complete solutions) can be used only for very small input. In the General case, if there are n events and k time slots, (n-1)!n2k checks for fixing the events on the timetable must be made [8].

There are many approaches to solve approximately the CCS, for instance:

- Constraint-based approaches. In addition, other methods are used, for example: object-oriented modeling of graphs and trees, "depth first search", combined methods with genetic algorithms and "backtracking" [9].

Techniques would utilized as well, Case in point: master arrangement of standards What's more graphs for vertices Furthermore edges Hosting qualities that store information ahead correlations between occasions [10][11][12].

• Meta-heuristic and hyper-heuristic methodologies. These are: "ant state optimization" [13]; "simulated annealing" [14]; "tabu search" [15]; "variable neighborhood search" [16] also how. The investigations need aid identified with finding those practically suitable strategy to a specific sort about issue and the development of ideal calculation As far as computational unpredictability Also memory utilization.

• Population-based methodologies. These are: hereditary Furthermore mimetic calculations [17] and their adjustments [18]. These calculations raise handy results to an adequate time.

• Graph-based methodologies. Clinched alongside these methodologies CCS may be quell Likewise An chart coloring issue [3][19]. A comparative methodology identified with nearby scan strategy will make examined clinched alongside All the more points The following.

2. On the CCS model. An portrayal of the CCS model may be depicted for [20]. The model incorporates matrices, vectors, parameters and delicate imperatives exhibited by weights. Those UCT issue may be mathematically figured Likewise a multi-criteria streamlining problem, Also then, Toward a weighing technique it is converted under an issue with a goal capacity. A mimetic algorithm (MA for cubic computational complexity) Also An hereditary calculation (GA with quadratic computational complexity), are exhibited for [21]. These calculations utilize the model recommended clinched alongside [20]. After those trials it might have been discovered out that mama dependably generates preferred results over.

Previously, [22] another algorithm that employments those assessment model recommended On [20] may be exhibited. In this methodology the occasions would gathered under gatherings. Those best result contingent upon those course of action of the occasions is searched. Main a little and only know could reasonably be expected routes from claiming grouping those occasions need been tried. The grouping for occasions clinched alongside gatherings for an rise to number about occasions over every particular case will be not pertinent on every one data information sets. To this reason An widespread approach how on would this may be introduced done [23].

II. A GRAPH COLORING BASED APPROACH

The graph-based approaches allow several unrelated groups of events to be processed independently from each other. These events correspond to the graph chromatic classes. In the scientific literature [24][19], various heuristic techniques for graph coloring adapted for CCS are described.

Let G = (V, E) be an undirected graph with set V = {v1, v2, ..., vn} of vertices and set E = {(vi, vj)|vi ∊ V, vj ∊ V} of edges, where unordered pair (vi, vj) is an edge in graph G meaning that vertex vi is adjacent (connected) with vertex vj. (Figure 1).

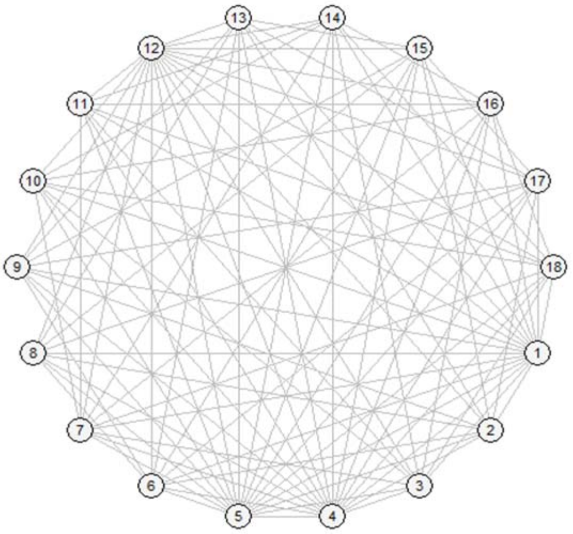

Fig. 1 Undirected graph G = (V, E)

The events in the timetable correspond to the graph vertices. The set of the edges represent the possible conflicts between events, i.e. every edge shows that its adjacent vertices (events) may not be carried out at the same time, because they use a common resource (student, lecturer or auditorium).

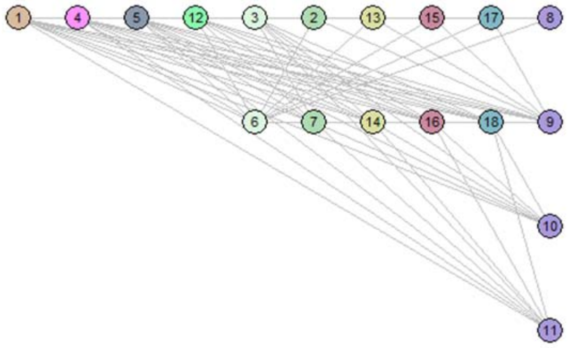

Fig. 2 Colored graph G = (V, E)

Figure 2 shows how vertices are colored and arranged in chromatic classes. The events in each column may be carried out at the same time, because they are not in conflict with each other. If all events have the same duration – t, then the minimum number of time slots that will be needed to conduct all events will be t.γ(G), where γ(G) is the chromatic number of graph G (i.e. the minimum number of coolers needed to color graph vertices). It should be pointed out that from each permutation of 1, 2, ..., γ(G), i.e. (γ(G))! an acceptable timetable can be generated [8]. However, in the real CCS the durations of the events are different.

Those chart (vertex) coloring issue may be restricted to appointing a color (from a provided for large number for colors) from claiming every of the vertices of the chart. This issue belongs of the class for NP-hard issues What's more to its accurate result it may be important to utilize a brute-force method, for example, such that backtracking [19]. To graphs with an expansive amount about vertices What's more edges (for instance more than 50) those correct systems need aid not viable (regarding performance). For such cases, those heulandite (or approximate) techniques are utilized. In these methods, On the result discovered will be not optimal, it may be near the optimal, Be that it might make palatable sufficient for the provided for issue [25].

An heulandites methodology to coloring the vertices of the chart is suggested in [26]. This method, to graphs, dependent upon those connections the middle of events, may be utilized within finding the chromatic classes. Initially, the vertices of the chart contingent upon their degree need aid sort program to a plunging route. Then, from exited to good know nonadjacent vertices need aid shaded for the principal shade. Those shaded vertices (and those contiguous will them edges) need aid uprooted starting with those graph, and the remaining vertices relying upon their degree need aid sort program On An plunging manner once more. Once more from cleared out on straight constantly on nonadjacent vertices need aid hued with a second color, and so forth. , until the greater part vertices of the chart are shaded.

An example of the implementation of the algorithm with 35 events, 48 students, 11 lecturers and 8 auditoriums is presented below. Initially, a matrix of the conflicts among events (with shared students, lecturers, or auditoriums) is created. This matrix has dimension n x n (n is the number of the events). At the first row and the first column of the matrix, the events that correspond to the lectures are located (Table 1). The matrix is symmetric to its main diagonal, and its elements are non-negative whole numbers. The value of each item shows the "cost" of the conflict between events ni and nj depending on the quantity of shared resources that are involved in this conflict [20].

Table I
A Matrix Of The Conflicts Among Events

e\e	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	...	e35
e1	0	12	12	12	12	12	12	12	12	420	...	12
e2	12	0	144	192	192	144	144	120	120	12	...	0
e3	12	144	0	192	192	144	144	120	120	12	...	0
e4	12	192	192	0	240	192	192	168	168	12	...	0
e5	12	192	192	240	0	192	192	168	168	12	...	0
e6	12	144	144	192	192	0	144	120	120	12	...	0
e7	12	144	144	192	192	144	0	120	120	12	...	0
e8	12	120	120	168	168	120	120	0	96	12	...	0
e9	12	120	120	168	168	120	120	96	0	12	...	12
e10	420	12	12	12	12	12	12	12	12	0	...	12
...	...	...	...	...	...	...	...	...	...	...	...	...
e35	12	0	0	0	0	0	0	0	12	12	...	0

An undirected graph is constructed by the matrix of the conflicts. In this graph, the vertices are the events, and the edges are the conflicts between them. In fact, the matrix of the conflicts is another way for a graph to be represented by an adjacency matrix. In this matrix each element is assigned a value which corresponds to the weight of the conflict (Figure 3).

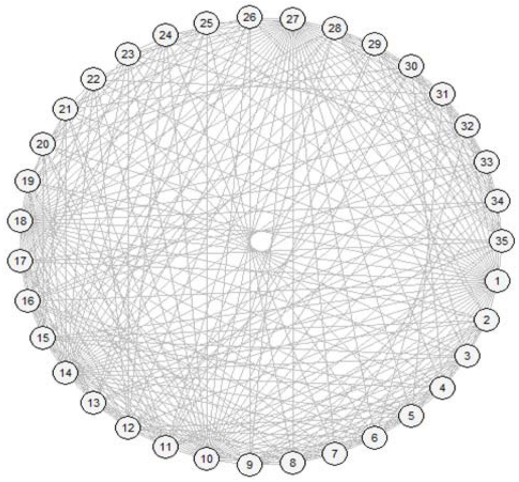

Fig. 3 An undirected graph with 35 vertices

After using the heuristic algorithm (described above), each of the vertices of the graph is assigned a particular color (Figure 4).

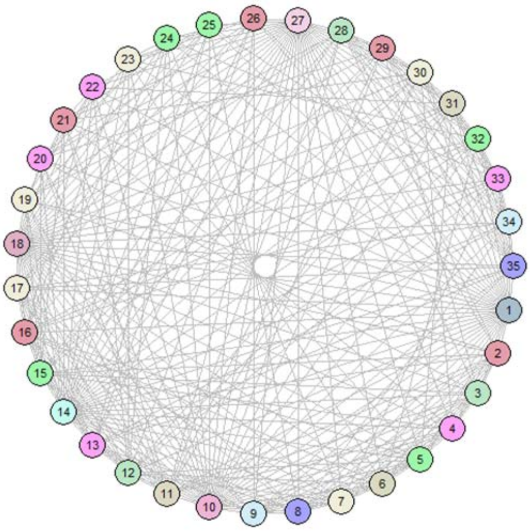

Fig. 4 An undirected colored graph with 35 vertices

The vertices colored with the same color form a chromatic class. These vertices in separate columns can be sorted, i.e., when drawing the graph every chromatic class in a separate column is displayed (Figure 5).

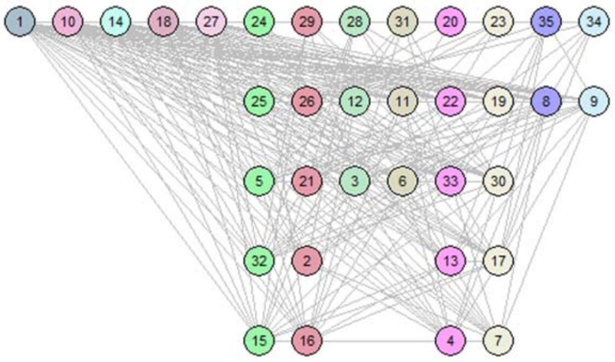

Fig. 5 The vertices arranged in chromatic classes

Since during those execution of the strategy In view of nearby scan those statute of the occasions may be of significance [21], it may be vital that the chromatic classes be orchestrated On a fitting way (such Similarly as one giving work to the best workable result), i. E. Those chromatic classes must be sort program taking after exactly criteria.

In the current study, three separate variants about an plan of the chromatic classes will make tested:. 1) contingent upon the shade indexes (set following those execution of the heulandite calculation for coloring the vertices of the graph).

2) relying upon those downright weight of the occasions clinched alongside An chromatic class.

3) contingent upon the downright check of occasions Previously, An chromatic population.

The effects got then afterward the execution of the neighborhood quest algorithm In view of chromatic classes (LSCC) (for those chart introduced done figure 3) need aid as takes after: Toward shade index: 25. 22; by downright count: 14. 55 What's more by aggregate weight: 13. 82. The best result is gotten when those chromatic classes were sort program by aggregate weight about occasions On them.

III. Experimental results

The aim of the experiments is to determine the best criterion to sort chromatic classes. The experimental conditions are the following: PC with Windows Server 2012 R2 Standard edition, 64-bit operating system with processor Intel(R) Xeon(R) E5645 at 2.40 GHz 2.39 GHz; RAM memory: 4,88 GB.

In Table 2, the results of the algorithm execution on twelve input data sets (IDS1, .., IDS12) are shown. The chromatic classes are sorted by color index, total weight and total count. The input data sets and the results obtained by the integrated information system for university course timetabling are presented in [27][28].

TABLE 2
RESULTS FOR ALL INPUT DATA SETS

IDS	E	S	L	A	Con-flicts	Color index	Total count	Total weight
1	152	23	43	25	1470	132.05	111.64	103.83
2	199	35	46	25	2071	120.53	102.65	101.18
3	95	23	24	16	866	28.28	27.40	27.97
4	87	20	25	13	775	48.55	47.49	43.04
5	153	28	37	17	1904	89.39	90.27	80.53
6	121	28	39	16	1140	83.06	76.65	74.40
7	131	30	32	17	1436	61.17	59.85	56.98
8	125	31	33	16	1284	87.84	84.04	76.82
9	584	108	108	59	7754	299.42	241.14	228.28
10	396	89	78	42	4775	178.87	156.95	157.63
11	623	116	129	62	8550	362.35	323.37	309.41
12	496	109	110	52	5812	228.66	210.10	210.83

The abbreviations used in Table 2 are as follow: E – events; S – students; L – lecturers; A – auditoriums;

IV. Conclusion And Future Work

The outcomes gotten show that the CCLS algorithm created better results The point when the chromatic classes were positioned Toward downright weight paradigm (in 9 for 12 data information sets). Best clinched alongside three of the cases, the CCLS algorithm discovered exceptional results At the foray paradigm might have been those downright number for occasions clinched alongside a chromatic population (Figure 6). For these three situations (IDS3, IDS10, Furthermore IDS12), the outcomes acquired need aid similar for the individuals attained At foray the chromatic classes Eventually Tom's perusing aggregate weight (the contrasts the middle of values would slight). In the different nine cases, those contrasts between those qualities of the results found would critical. Those Decision that might make drawn may be that the outcomes gotten then afterward the execution of the CCLS calculation will make better On the chromatic classes would sort program as stated by the aggregate weight paradigm.

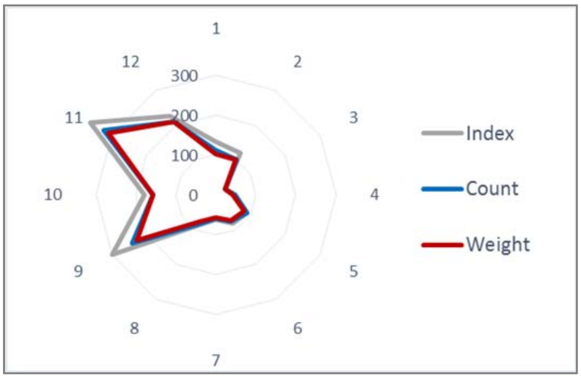

Fig. 6

Figure 6 comes about as stated by the kind criteria. Understanding of the comes about could a chance to be produced on the foundation that at first the algorithm with nearby hunt figures those time slots to the occasions of the chromatic population for the best aggregate weight, Along these lines those chance slots that have the practically huge effect on the last score of the timetable are involved Toward occasions with vast weights. Thereby, practically of the delicate requirements of those assets (students, lecturers, What's more auditoriums) would satisfied, which thus prompts a finer assessment of the timetable.

Those current ponder might be broadened in the Emulating areas:. 1) streamlining of the calculation As far as quality, complexity, Also execution time.

2) directing extra trials with that's only the tip of the iceberg data information sets for the reason for checking the effects exhibited in this article.

3) Look into of other criteria on kind the chromatic classes, for example, such that those downright span of the occasions to each population.

4) similar Investigation of the comes about gotten starting with the LSCC calculation for the comes about acquired Toward other algorithms, for example, GA, MA, Also EGB (in money house under personal satisfaction What's more execution time).

References

[1] Chen R, Shih H. Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms. 2013; 6: 227-44.

[2] Kohshori MS, Abadeh MS. Hybrid genetic algorithms for university course timetabling. International Journal of Computer Science Issues. 2012; 9(2): 446-55.

[3] Burke EK, Marecek J, Parkes AJ, Rudova H. Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research. 2010; 37: 582-97.

[4] Kralev V. Study and design of automated systems for allocation of resources. PhD Thesis. South-West University Press. DOI: 10.13140/RG.2.1.3575.9846; 2010.

[5] Kang L, White G. A logic approach to the resolution of constraints in timetabling. European Journal of Operational Research. 1992; 61:306-17.

[6] Partovi F, Arinze B. A knowledge-based approach to the faculty course assignment problem. Socio-economics planning science. 1995; 29(3): 245-56.

[7] Petrovic S, Burke E. University timetabling handbook of scheduling: algorithms, models and performance analysis. CRC Press; 2004. Chapter 45.

[8] Burke E, Petrovic S. Recent research directions in automated timetabling. European journal of Operational Research. 2002; 140(2): 266-80.

[9] Patrick K, Godswill Z. Greedy ants colony optimization strategy for solving the curriculum based university course timetabling problem. British Journal of Mathematics & Computer Science. 2016; 14(2): 1-10.

[10] Elmohamed M, Coddington P, Fox G. A comparison of annealing techniques for academic course scheduling. In proceedings of the practice and theory of automated timetabling, Toronto, Canada. Springer-Verlag. 1998 (pp. 92-112).

[11] Costa D. A tabu search for computing an operational timetable. European Journal of Operational Research. 1994; 76: 98-110.

[12] Abdullah S, Burke E, McCollum B. An investigation of a variable neighborhood search approach for course timetabling. In proceedings of the 2nd Multidisciplinary international conference on scheduling: Theory and applications, New York, USA. 2005 (pp. 413-27).

[13] Soliman SE, Keshk AE. Memetic algorithm for solving university course timetabling problem. International Journal of Mechanical Engineering and Information Technology. 2015; 3(8): 1476-86.

[14] Lewis R, Paechter B. New Crossover operators for timetabling with evolutionary algorithms. In proceedings of the 5th International conference on recent advances in soft computing (RASC 2004), Nottingham, UK. 2004 (pp. 189-95).

[15] Werra D. Extensions of colouring models for scheduling purposes. European Journal of Operational Research. 1996; 92: 474-92.

[16] Kralev V. A model for the university course timetabling problem. International Journal "Information Technologies & Knowledge". 2009; 3(3): 276-89.

[17] Kralev V. A genetic and memetic algorithm for solving the university course timetabling problem. International Journal "Information Theories & Applications". 2009; 16 (3): 291-9.

[18] Kralev V, Kraleva R. Variable neighborhood search based algorithm for university course timetabling problem. In proceedings of the Fifth international scientific conference, FMNS-2013. South-West University, Bulgaria. 2013 (pp. 202-14).

[19] Kralev V, Kraleva R, Yurukov B. An event grouping based algorithm for university course timetabling problem. International Journal of Computer Science and Information Security. 2016; 14(6): 222-9.

[20] Burke E, Elliman D, Weare R. A university timetabling system based on graph colouring and constraint manipulation. Journal of Research on Computing in Education. 1994; 27(1): 1-18.

[21] Halldorsson MM. A still better performance guarantee for approximate graph coloring. Information Processing Letters. 1993; 45: 19–23.