Annals of Pure and Applied Mathematics
Vol. 7, No. 1, 2014, 97-103
ISSN: 2279-087X (P), 2279-0888(online)
Published on 9 September 2014
www.researchmathsci.org

Annals of
Pure and Applied Mathematics

Selection of *k* Sets of Disjoint Channels for Advertising with an Aim to Maximize Viewers' Count and Minimize Cost Constrained by Budget

Sumana Bandyopadhyay[1] and Rajat Kumar Pal[2]

[1]Department of Computer Science, Prabhu Jagatbandhu College,
Howrah – 711 302, West Bengal, India; E-mail: emlsmn_bnrj1981@yahoo.co.in
[2]Department of Computer Science and Engineering, University of Calcutta,
92, A. P. C. Road, Kolkata – 700 091; E-mail: pal.rajatk@gmail.com

Received 10 July 2014; Accepted 25 August 2014

Abstract. A production unit sponsors a number of programmes in different television channels. Aim of the unit is to spread the product related message to a maximum number of targeted viewers. However, to make the venture a successful one, a sufficient amount of budget must have to provide. So, this paper attempts to maximize number of viewers, while keeping cost of purchasing the programmes within a specified amount of budget.

Keywords: *Advertisement, television rating, budget, comparability graph, maximum weighted k-clique.*

AMS Mathematics Subject Classification (2010): 05C78

1. Introduction

In this paper our objective is placement of advertisements (or ads) in different television channels in such a way that the ad related message reach to the maximum number of viewers as well as the cost incurred in the process must be constrained by the capability of the production unit who is interested in successful marketing of the product. Hence, we have to maximize the number of viewers (termed as viewers' count) and minimize total cost involved in this process.

The implication of optimizing both the parameters is very much realistic. The production units promote their goods through advertisements that are broadcasted as commercial breaks in between programmes running throughout the day in different television channels. The motive of the production units is to reach out to as many customers as possible but they are bound to follow some planning and marketing constraints of which the total budget is one of the most important ones. The cost of procuring break points is liable to be optimized. With the help of the other parameter, i.e., viewers' count, we are trying to approximate the popularity of programmes. This parameter is derived from television ratings published by various national and international rating systems or by some surveys. So, in this paper, we are keen to optimize both the parameters while keeping the total cost below some predefined budget.

In this context we may reiterate the idea of *multi-objective optimization*. Incidentally, several algorithms are available in literature [1, 6] those like to maximize viewers' count only. On the contrary, in this paper, we desire to optimize both the parameters, i.e., viewers' count of programmes (giving this parameter the highest priority) followed by the cost required to purchase the break points. The graph theoretic formulation results a comparability graph from the problem domain.

2. Preliminaries

In this section we are going to make the reader familiar with some graph theoretic terms and invariants. For ease of the reader, we include some vital terms in the form of corollaries.

Corollary 1. A clique consists of a maximum number of vertices present in a graph, each pair of which is connected by an edge in the given graph. Such a clique is usually a maximum clique as it helps to declare the clique number of a given graph. However, a maximal clique is also a clique wherein none of the remaining vertices of the graph could be included to make it larger.

The graph obtained from the problem domain is a kind of perfect graph; more specifically, this graph is a *comparability graph* [5]. The edges of a comparability graph can always be transitively oriented [5]. Computation of maximum weighted k-clique of the comparability graph modeling the problem affords us the solution of the problem sought here.

Corollary 2. A *maximum weighted k-clique* of a graph is a collection of k-disjoint *maximal cliques* that weigh the maximum overall possible *k-disjoint maximal cliques* in the graph. In general, the *maximum weighted k-clique* computation problem is NP-complete [3], but when restricted to comparability graphs it becomes tractable [4].

3. Formulation of the problem

Now, we are going to impose some constraints that should be followed by the graph theoretic formulation of the problem addressed herein. The constraints are listed below:

1. The standard television ratings are available for a particular programme of any channel based on some surveys and national and/or international rating system.
2. In general, programme slots are non-intersecting if their corresponding broadcasting time spans are also non-intersecting or non-overlapping.
3. Programmes on different television channels begin at a specific time (here midnight). In case a programme overlaps the boundary, we may split it into two sub-slots; one that terminates at midnight and another that starts broadcasting at the midnight, both having the same viewers' count and cost.

According to the formalism, we have devised a scheme to optimize the ratio of the television rating and cost keeping budget as a constraint. The said ratio is used as weight on the vertices of the graph as obtained from the graph theoretic formulation of the problem.

Next we like to show (with the help of a suitable instance) the graph theoretic formulation that clarifies how programme slots are laid on the real time line and using graph theoretic modeling how a comparability graph can be constructed from the instance.

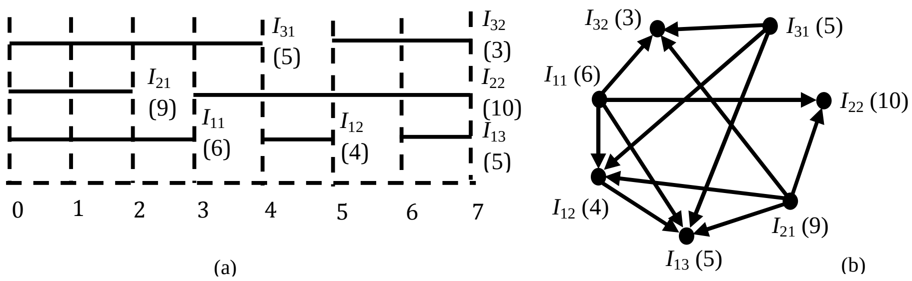


Figure 1: (a) A set of programme slots on three different channels. I_{XK} (P) represents the kth programme slot (or interval) on channel X and P represents the weight, which is the ratio of viewers' count to the cost of purchasing ad breaks. (b) The comparability graph structure obtained for the set of intervals there in (a), which is oriented as the (chronological) programme slots occur from left to right in non-overlapping fashion.

In Figure 1(a), we have taken an appropriate instance (ranging from 0 to 7 hours, in some duration of time scale) with programme slots running in different channels and Figure 1(b) shows the corresponding weighted comparability graph. While converting an instance into the equivalent graph structure, each vertex I_{XM} represents the Mth programme interval of channel X (after midnight). Each vertex is weighted as explained previously. Edges are introduced between all such pairs of vertices I_{XM}, I_{YN} that represent non-intersecting time slots. For instance, the spans of I_{12}, I_{32}, or I_{13} are completely disjoint with respect to the span of interval I_{11}. Edges are directed from an interval I_{XM} to another interval I_{YN}, if I_{XM} terminates broadcasting before or exactly when I_{YN} begins to broadcast. In the above example, edges have been introduced and oriented from the vertex analogous to I_{11} to the vertices matching to I_{12}, I_{13}, I_{22}, and I_{32}. No edges are introduced between intervals I_{11}, I_{21}, and I_{31}, as the intervals have nonempty intersections in the spans of their broadcasting times; therefore, related vertices in the graph of Figure 1(b) are independent to each other.

We can now propose an appropriate graph based definition for the programme interval associations. We call this graph a *broadcasting graph*. Let G be the graph structure that models the interval associations. Then G is a 2-tuple (V, E), where V represents a set of weighted vertices and E represents a set of directed edges between non-overlapping intervals modeled as vertices. Each vertex $I_{XM} \in V$ is weighted by W_{XM}, where W_{XM} is defined as follows:

W_{XM} = *Television rating (or Viewers' count) / Cost of purchasing breakpoints*

Also, an edge e_{XY} is directed from I_{XM} to I_{YN}, if the programme M in channel X terminates on or before the beginning of programme N in channel Y, or vice versa. Sometimes, it may so happen that $Y = X$ and $N = M+1$, where certainly Y and X are nonoverlapping to each other.

Lemma 1. *The broadcasting graph is a comparability graph* [1].

In graph theoretical terms, the problem is stated in the form of following lemmas.

Lemma 2. The maximum weighted path based on the orientation from a vertex corresponding to a programme broadcasted earlier to a vertex corresponding to a programme broadcasted later induces a maximum weighted clique in a comparability graph [1].

Lemma 3. The maximum weighted k-clique computation of a comparability graph is polynomial time computable for any value of k [4].

4. Development of the algorithm *MWkQ* constrained by budget

The algorithm described in this section aspires to compute the *maximum weighted k-clique* of the broadcasting graph, constrained by a budget. The algorithm developed here is based on the graph traversal method *depth first search* (*DFS*) [2]. We call this modified DFS algorithm *DFS-Variant.* However, the algorithm maintains the corresponding path information, and cumulative ratio and cumulative cost of the path. Here a *path* denotes a sequence of vertices from one of the source vertices to one of the sink vertices that is reached by tracing the algorithm. The cumulative ratio means the sum of the weight ratios W_{XM} of the vertices belonging to this path. Similarly, the cumulative cost is simply the sum of the cost of the vertices belonging to the path. While checking for inclusion of a new vertex within the path, an alternative path is rejected, if the following conditions as briefed below occur.

(i) If the new path produces same / lower cumulative ratio and same / higher cumulative cost.

(ii) If the new path produces higher cumulative ratio and higher cumulative cost but increase in cumulative ratio is *sufficiently less* than the increase in cumulative cost.

(iii) If the new path produces lower cumulative ratio and cumulative cost but decrease in cumulative ratio is *significantly high* with respect to decrease in cumulative cost.

The terms "*sufficiently less*" and "*significantly high*" can be formally defined by comparing to some threshold values. Threshold values are computed as functions of the available budget, average ratio, average cost, cumulative ratio, and cumulative cost at the current time instant. On the contrary, for inclusion as a better alternative (new) path our algorithm checks whether one of the following conditions occurs in practice.

(i) If the new path produces same / higher cumulative ratio but lower / same cumulative cost.

(ii) If the new path produces higher cumulative ratio and higher cumulative cost but increase in cumulative ratio is *sufficiently higher* than increase in cost and the cumulative cost is still under budget.

(iii) If the new path produces lower cumulative ratio and cumulative cost but decrease in cumulative cost is *much more* with respect to decrease in cumulative ratio.

The same logic (or threshold values) as stated above is applied for determining when increase in cumulative ratio is "*sufficiently higher*" or decrease in cumulative cost is "*much more*".

While executing the algorithm, we maintain the vertices in the current path under consideration in a specific list called the *Path List* (*PL*). To implement our approach, the entries of *Path List* are vertices ordered primarily by the associated costs in decreasing order, and secondly by the associated ratios in increasing order. Suppose there are n vertices in the current path list, $PL_1 = \{v_1, v_2, ..., v_n\}$. Also, suppose the inclusion of a new vertex v_{n+1} arises one of the situations for rejection or simply the cumulative cost of the path goes above the budget. The cumulative ratio of the path also increases. Let this intermediate path configuration be denoted by $PL_2 = \{v_1, v_2, ..., v_n, v_{n+1}\}$.

Now, in such a situation, we pick up the first vertex v_1 from PL_2 removing which from the path may ultimately result in the cumulative cost to go below (or equal to) the budget. The removal of v_1 also decrements the overall cumulative ratio. Let this new configuration of the path list be denoted by $PL_3 = \{v_2, v_3, ..., v_n, v_{n+1}\}$. Now, if PL_3 has cumulative ratio still greater than PL_1, then this path is accepted as the newly generated path. The removal of vertices from the ordered *Path List* can be continued as long as the cumulative ratio of the resulting path is higher than the cumulative ratio of PL_1 and the cumulative cost of the newly generated paths remain within the available budget. If a path configuration PL' exists such that the cumulative ratio of PL' is greater than the cumulative ratio of PL_1 and the cumulative cost of PL' is less than the available budget, then the modified (or new) *Path List* is accepted to reflect the change in the path, i.e., $PL = PL'$.

Following the logic of DFS, the algorithm ultimately backtracks to the source vertex from where it started traversing and computes a *maximum* or near to the *maximum weighted path* from that source vertex. The same method is applied for each of the source vertices. At the end, a path with the maximum weight is obtained by simple comparison of the paths starting from different source vertices. By tracing out a path, actually a *maximum* or near to the *maximum weighted clique* of the broadcasting graph is being computed.

Suppose, after one iteration of the algorithm, the path being outputted has cumulative cost sufficiently less than the budget, i.e., there exist(s) other path(s) within the broadcasting graph whose cumulative cost is less than or equal to the remaining budget. Then the same algorithm can be applied repetitively on the (remaining) graph. In such a case, the vertices in the *clique* computed in the previous iteration are deleted (along with their adjacent edges) from the original graph. Thus, the same algorithm can be iterated a fixed k number of times for complete utilization of the budget. According to Gavril, the *maximum weighted k-clique* can be computed successively by applying the *maximum weighted clique* computation algorithm on a comparability graph for k times [4]. Although being constrained by budget (as a practical issue) we have deviated from the pure *maximum weighted k-clique* computation of comparability graphs, and applied some greedy logic.

In some cases, after completion of a fixed k number of iterations of the algorithm, a very small amount of the budget may still remain unutilized. Here, no further *clique* may be computable, using our algorithm, whose cumulative cost is less than or equal to the remaining budget. To cope with the situation, we use a greedy approach. Once the *maximum weighted k-cliques* have been removed (in succession) from the original graph, we sort the remaining vertices (programme slots) in decreasing order of their weight ratios W_{XM}. We pick the first vertex with the highest weight ratio whose additional cost does not result in the cumulative cost to exceed the proposed budget. This process may be repeated until the budget is totally utilized or the remaining budget is not sufficient to purchase any of the existing (remaining) programme slots.

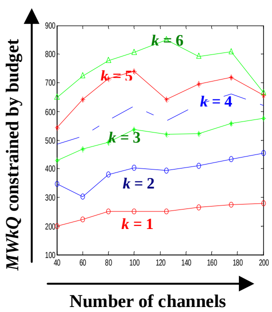
5. The computational complexities of algorithm *MWkQ* constrained by budget

One can observe that algorithm *MWkQ constrained by budget* primarily uses the graph traversal method *Depth First Search* (*DFS*) with some extra operations that keep the path found out (or clique computed) to be within the budget while maximizing cumulative ratio [2]. Therefore, we may claim that the complexities (both time and space) of algorithm *MWkQ* constrained by budget are O(n+e), where n is the number of vertices and e is the number of edges of the comparability graph instantiated.

6. Experimental results

We have implemented the algorithm, developed in Section 4, and produced a huge amount of data in support of our invention. Instances are generated (pseudo-randomly) for different number of channels, ranging from 40 to 200. For each channel number 20 instances have been generated and an average value is received as an acceptable data. Here we have taken into consideration from 12:00 at noon to 4:00 PM, where mostly housewives, old men or those having night duty watch television. We have summed up the experimental results made by us in Table 1. In this paper, we have strived to optimize two parameters, i.e., maximizing the cumulative viewers' count and minimizing the cumulative cost. But ultimately we have outputted the viewers' count in place of the ratio as production units are interested in the number of viewers.

Table 1: Implemented results showing the performance of *MWkQ* for an afternoon session during 12:00 at noon through 4:00 PM.



Number of Channels	Budget						
	Number of Parallel Sessions						
	k = 1	**k = 2**	**k = 3**	**k = 4**	**k = 5**	**k = 6**	**k = 7**
40	73.900	73.900	75.210	82.670	86.000	87.000	94.000
60	70.150	74.240	76.125	79.140	84.265	86.400	87.000
80	66.500	66.500	70.750	75.800	81.600	84.000	88.000
100	68.900	70.210	76.000	77.840	89.600	89.600	-
125	72.550	72.550	76.650	84.570	89.270	93.000	-
150	68.850	68.850	71.590	78.080	84.140	92.500	92.500
175	66.900	66.900	69.440	76.080	79.830	82.670	-
200	66.650	67.800	70.800	73.300	81.000	88.000	88.000

Number of Channels	Viewers' Count, Cost						
	Number of Parallel Sessions						
	k = 1	**k = 2**	**k = 3**	**k = 4**	**k = 5**	**k = 6**	**k = 7**
40	199.70, 34.550	346.48, 50.950	428.56, 60.470	484.59, 67.330	543.59, 76.780	649.28, 86.800	748.41, 89.000
60	223.20, 36.350	303.15, 47.240	468.49, 59.125	513.50, 68.670	641.37, 77.750	723.02, 84.500	836.00, 87.000
80	250.40, 35.950	380.47, 49.000	492.63, 59.680	568.23, 69.180	714.61, 77.800	776.20, 82.000	817.20, 88.000
100	251.30, 38.500	402.90, 51.950	536.77, 64.470	617.16, 72.076	738.80, 85.200	804.60, 89.500	-
125	251.60, 35.500	392.62, 48.400	520.84, 60.710	566.41, 69.130	641.90, 78.550	850.00, 88.500	-
150	265.40, 36.150	420.25, 49.050	540.20, 58.294	620.10, 68.750	698.90, 78.750	895.00, 81.500	897.00, 88.000
175	274.55, 33.850	433.35, 47.550	557.19, 58.900	661.08, 67.920	717.67, 73.670	808.00, 82.000	-
200	278.20, 38.000	453.56, 50.900	576.89, 60.900	621.20, 67.300	655.74, 71.670	664.74, 74.000	724.32, 81.000

Figure 2: Performance of *MWkQ* for different parallel sessions versus number of television channels for broadcasting programmes during an afternoon session from 12:00 at noon onwards for four hours.

Here, the number of channels is plotted along x-axis (as shown in Figure 2), and the probable *maximum weighted k-clique* is plotted along y-axis. We have plotted the curves for various values of k that means the number of parallel sessions that can be fitted within the budget. However, an increase in number of parallel sessions increases with the number of channels. In Figure 2, the plots for k = 1, 2, or 3 have nearly equal spacing. But when the value of k increases above 3, more and more individual instances get terminated. It should be remembered that the values used in the tables or curve plotting are actually average values. So for larger values of k, curves are mostly near to each other and for sufficiently large values of number of channels, they nearly try to converge to some small range of values (see Figure 2). For small number of channels, there may create some spurious data generating sudden rise of the corresponding curve. But as the channel number increases, the curves turn to be nearly saturated. However, we have generated 20 instances for each of the channel numbers. If the number of generated instances is increased, then the corresponding average values may produce smoother curves, we hope.

7. Conclusion and future work

Production units launch and promote their items through electronic or print media. There are different parameters for both types of media that when properly set can result in effective advertisement of the product. Production units have a very vague idea about these parameters. Ad agencies take this responsibility and provide production units with a variety of solutions. Each of these solutions bears a cost and, depending on their budget, production units may select the scheme of their choice. The cost of each solution is governed by the parameters which decide the efficiency of the advertising media. There are various approaches available in literature regarding the selection of ad slots (commercial breaks) within programme slots, but this paper is unique in the sense that the totality of the problem has been taken care of. Our approach has great market value and it targets two different entities, viz., ad agencies and production units, of the advertisement industry.

REFERENCES

1. S.Bandyopadhyay and R.K.Pal, An algorithm for selecting programme slots to broadcast advertisements in parallel in different television channels, *Proc. of IEEE International Conference on Computer Science and Automation Engineering (CSAE 2011)*, session c2, Shanghai, China, Vol. 2, (2011), pp. 399-403, June 10-12.
2. T.H.Cormen, C.E.Leiserson, R.L.Rivest, and C.Stein, *Introduction to Algorithms*, Prentice Hall of India, New Delhi, 2002.
3. M.R.Garey and D.S.Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.
4. F.Gavril, Algorithms for maximum k- colorings and k-coverings of transitive graphs, *Networks*, 17 (1987) 465-470.
5. M.C.Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
6. M.Pal, A.Saha and T.K.Pal, selection of programme slots of television channels for giving advertisement: a graph theoretic approach, *Information Sciences*, 177 (2007) 2480-2492.