Journal of Theoretical and Applied Information Technology
10th October 2014. Vol. 68 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·

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

A MMKP BASED HEURISTIC FOR QUALITY ENHANCED SCALABLE VIDEO STREAMING OVER WIRELESS NETWORKS

[1]SWARNA PARVATHI, [2]RAJU DAS, [3]EASWARAKUMAR, [4]KALAIARASI
[1&4]Sri Venkateswara College of Engineering, Department of IT, Pennalur, India
[2]Space Application Centre, ISRO, Ahmadabad, India
[3]Anna University, CEG Campus, Chennai, India
E-mail: [1]swarna@svce.ac.in, [2]raju@sac.isro.gov.in, [3]easwara@cs.annauniv.edu, [4]kalaiarasi@svce.ac.in

ABSTRACT

Scalable Video Coding (SVC) is a very promising encoding technique that adapts to streaming video over wireless networks with bandwidth fluctuations. This paper proposes a Bandwidth Aware Layered Streaming Algorithm (BALSA), a MMKP (Multi-dimensional Multiple-Choice Knapsack Problem) based GHPMH (Gradational Hull Pareto Minimization Heuristic) to stream scalable video to heterogeneous users with different client's display under bandwidth limitations so as to maximize the average video quality over all the streams. Using extensive simulation we show that our algorithm finds solutions which are close to the optimal (within 1 db) under realistic conditions with reduced computational complexity.

Keywords: *MMKP (Multi-dimensional Multiple Choice Knapsack Problem), Scalable Video Streaming, Pareto Minimization*

1. INTRODUCTION

Video multicast is a convenient way to transmit video with heterogeneous terminals over wireless networks. Multicast applications require a low cost infrastructure to provide quality video with graceful degradation and tolerable delay to a large size of heterogeneous users with different client's display, processing capabilities,power and bandwidth limitations. It is also important to increase the throughput of the system while keeping the network maintenance cost low. Among those challenges, packet loss often has a severe effect on transmitted video quality. Thus the design of a robust, efficient and scalable delivery infrastructure is a technically challenging task. In other words, to cope with these problems, a channel-adaptive video transmission method using H.264 Scalable Video Coding (SVC) [1], [2], [3], [4], [5], is needed which better adapts to mobile environments with heterogeneous clients and time varying available capacity. SVC supports temporal, spatial and quality scalabilities at bit stream level which enables easy adaptation of video by selecting the subsets of bit streams. According to the network conditions and receiver capabilities, the pre-encoded SVC bit stream can be easily adapted by the streaming server to provide various spatial, temporal and quality (SNR) resolutions. Further, the SVC layered structure put the data of different importance into different layers. With such features, the SVC bit stream is more suitable than the non-scalable [6], bit stream when the video packets are transmitted over an error-prone channel with fluctuated bandwidth. Scalable video coding involves generating a coded representation (bit-stream) that allows decoding of appropriate subsets to reconstruct complete pictures of resolution or quality commensurate with the proportion of the bit-stream decoded. The minimum bit-stream subset that can be decoded is called base layer. The remaining bits in the bit stream are called enhancement layer(s) and by decoding the enhancement layer(s) more details are obtained to get the video at higher resolution or quality as compared to base layer.

Multiple-choice Multi-dimensional Knapsack Problem (MMKP) [7]-based optimization has been used previously in the design of an adaptive multimedia system (AMS) [8].

The contributions of this paper are twofold:

1. The formulation and implementation of a Multi-dimensional Multiple choice Knapsack Problem (MMKP)-based video streaming algorithm called Bandwidth Aware Layered Streaming Algorithm (BALSA) which is a parameterized gradational heuristic based on the principles of Pareto minimization to stream scalable video to heterogeneous users with different client's display

under bandwidth limitations. Gradational computation in the MMKP context means that (fractional) solutions for an MMKP instance can be computed by considering groups of items one at a time. This enables incremental computation of solutions; resulting in a heuristic that is very well scalable to large MMKP instances [9]. The main objective of the proposed MMKP based streaming algorithm is to exactly select one sub stream from each (SVC encoded) scalable layer such that the average quality is maximized.

The proposed algorithm is shown to have the following significant advantages over the existing 0/1KP-based (Multiple Choice Knapsack Problem-MCKP) video streaming methods:

(a) The proposed MMKP-based video streaming algorithm is observed to stream video with multiple resource constraints such as varying client capabilities, varying network bandwidth.

(b) In contrast to the 0/1KP-based Video streaming methods which can satisfy only a single resource constraint at a time, the proposed MMKP-based video streaming methodology is capable of supporting multiple client-side resource constraints simultaneously.

2. Pareto Algebra [10] has recently been introduced as an algebraic framework for compositional calculation of Pareto optimal solutions in multi-dimensional optimization problems. Therefore, the proposed video streaming algorithm is mathematically derived from MMKP which is in turn proved by a Pareto Minimization algorithm, GHPMH (Gradational Hull Pareto Minimum Heuristic).

The remainder of the paper is organized as follows. Section II provides a brief review of related work. We state our problem and present the analytical formulation for it in Section III. In Section IV, we present to solve our streaming algorithm “Bandwidth Aware Layered Streaming Algorithm (BALSA)”, a MMKP based GHPMH (Gradational Hull Pareto Minimization Heuristic), which is a parameterized gradational heuristic based on the principles of Pareto minimization. Section V evaluates BALSA on various SVC encoded test data sets. Section VI concludes the paper.

2. RELATED WORK

For transporting video over wireless, there have been many proposals of adaptive approaches and services in the literature, which include an “adaptive reserved service" framework, an adaptive QoS management [9]. [11] describes the frame allocation problem for broadcasting variable bit rate video over WiMAX, but does not consider scalable video content. Using the branch and bound with linear programming (BBLP), [8], [12], [13] and [14] presented exact algorithms for MMKP, 01-KP, MCKP and MDKP respectively which were related to adaptive multimedia systems. Although BBLP is quiet effective, the time complexity is too high and unacceptable for NP-complete task. [15] consider splitting a video stream into two streams and transmitting them over two different broadcast networks. This ensures a minimum video quality at all times while maintaining the flexibility of using other applications. While this approach has its benefits, it is not very attractive from a deployment point of view since the service provider has to install and manage the infrastructure for two different kinds of networks.

Also the solutions described above evaluate the performance of video streaming as an application along with other wireless applications. In contrast, the proposed approach considers a multimedia-intensive system which uses scalable encoded video so that the system by itself becomes an adaptive multimedia system. A new MMKP based video streaming algorithm is proposed for solving this problem of video streaming.

3. PROBLEM FORMULATION

We formulate our problem of Scalable Video Streaming over wireless networks by indicating a MMKP based approach which uses Pareto minimization techniques. We devise our own heuristic to solve the above mentioned MMKP based video streaming procedure. A H.264/SVC video stream may be represented using an essential base layer (also called the main profile) and one or more optional enhancement layers (called scalability profiles). The base layer constructs the coarse or base representation of the stream, and the enhancement layers successively improve it.

Consider the scalability of H.264/SVC with one base layer and two enhancement layers with sub streams inside each. The base layer (BL) gives QCIF video with a resolution of 176×144 and specified frame rates. The enhancement layer (EL_1) improves video to CIF with a resolution of 352×288. The Enhancement layer (EL_2) is of an enhanced quality with a resolution of 704×576.This feature of SVC encoded videos may be applied for streaming by just dropping the least important levels depending on the available bandwidth and client requirements.

Consider a scenario of streaming SVC encoded video from a video server to multiple clients of different display capabilities through a wireless network. The original video sequence is encoded into '*S*' scalable streams. Each scalable stream'*s*', $1 \leq s \leq S$ has a number of layers (a base layer and one or more enhancement layers). The objective of the proposed MMKP based video streaming is to choose '*l*' total value of the collected items(sub streams), subject to *m*(here m=2 viz., Bandwidth ,Resolution)) resource constraints of the knapsack. In mathematical notation, let v_{ij} be the value of the j^{th} item of the i^{th} group, $\vec{r}_{ij}=(r_{ij1},r_{ij2}, \dots\dots r_{ijm})$ be the required resource vector for the j^{th} item of the i^{th} group, and $\vec{R}=(R_1,R_2,\dots\dots R_m)$ be the resource bound of the knapsack.

Formally, the MMKP is expressed as follows:

$$V_{max} = \text{Maximize}, \quad \sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij}v_{ij}\,; i=1\dots n,\ j=1\dots\dots l_{i.}$$

$$\text{Such that,}\quad \sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij}r_{ijk} \leq R_k\ ;\ k=1\dots\dots\dots m.$$

$$\sum_{i=1}^{n} x_{ij}=1$$

$$x_{ij}\in\{0,1\}$$

Consequently, our problem is to determine Maximize,

$$\sum_{s=1}^{S}\sum_{l=1}^{L} x_{sl}q_{sl} s=1\dots S; l=1\dots L.$$

$$\text{Subject to,}\quad \sum_{s=1}^{S}\sum_{l=1}^{L} x_{sl}r_{sl} \leq R$$

$$\sum_{s=1}^{S}\sum_{l=1}^{L} x_{sl}b_{sl} \leq B$$

$$\text{and}\quad \sum_{l=1}^{L} x_{sl}=1$$

$$x_{sl}\in\{0,1\}$$

Here x_{sl} is either 0 or 1. 0 implies a layer '*l*' of stream'*s*' is not picked. If 'x_{sl}' is 1, implying layer '*l*' of stream'*s*' is picked.

Table 1: Table of Symbols

Symbols	**Description**
S	Number of Streams
s	Scalable Video Streams
L	Number of layers
l	Substream
q_{sl}	PSNR of substream *sl*
b_{sl}	Bit rate of substream *sl*
r_{sl}	Resolution of substream *sl*
x_{sl}	0 or 1
R	Maximum Resolution
B	Maximum Bandwidth

4. HEURISTIC FOR STREAMING VIDEO OVER WIRELESS NETWORKS

Here we present our algorithm BALSA for streaming video over wireless networks.

4.1 Concepts

A preprocessing of the SVC encoded output of representing the resolutions by constants is performed. We start with finding a feasible solution. We devise an algorithm which uses the concepts of Pareto minimization by calculating fractional solutions and combine these fractional solutions iteratively to find a final feasible solution. Fractional solutions are found in order to minimize the time and space complexity. Our heuristic approach provides a computational efficiency of *O(nm)* linear time.

4.2 Definition

Consider a finite set of objectives $\{1, \dots, p\}$. A bounded cost vector $u = (u_1,\dots\dots u_p)$ is a vector of *p* components where each $u_j \in Z_+$ represents the cost with respect to objective *j* and $0 \leq u_j \leq K$, respectively. We adopt the following notation. A cost vector which has all components equal to 0 is denoted by 0, while a cost vector having one or more components equal to *K* is denoted by *K*.

A Multi-objective Constraint Optimization Problem (MO-COP) with $p > 1$ objectives is a tuple $M = \langle X,D,F \rangle$, where $X = \{X_1, \dots, X_n\}$ is a set of variables, $D = \{D_1, \dots, D_n\}$ is a set of finite domains and $F = \{f_1, \dots, f_r\}$ is a set of multiobjective cost functions. A multi-objective cost function $f_k(Y_k) \in F$ is defined over a subset of variables $Y_k \subseteq X$, called its scope, and associates a bounded cost vector $u = (u_1,\dots,u_p)$ to each assignment of its scope. The cost functions in F can be either soft or hard (constraints). Without loss of generality we assume that hard constraints are represented as multi-objective cost functions, where allowed and forbidden tuples have cost 0 and K, respectively.

The sum of cost functions in F defines the objective function, namely $F(X) = \sum_{k=1}^{r} f_k(Y_k)$. A solution is a complete assignment of the variables $\overline{x} = (x_1, ..., x_n)$ and is characterized by a cost vector $u = F(\overline{x})$, where u_j is the value of $\overline{x}$ with respect to the j^{th} objective. Hence, the comparison of solutions reduces to the comparison of their cost vectors. The set of all cost vectors attached to solutions is denoted by S. We next present some definitions related to Pareto dominance concepts.

Definition 1 (Pareto dominance). Given two cost vectors u and $v \in Z_+^p$, we say that u dominates v, denoted by $u \preccurlyeq v$, if $\forall i\ u_i \leq v_i$. We say that u strictly dominates v, denoted by $u \prec v$, if $u \preccurlyeq v$ and $u \neq v$. Given two sets of cost vectors U and V, we say that U dominates V, denoted by $U \preccurlyeq V$, if $\forall\ v \in V$, $u \in U$ such that $u \preccurlyeq v$.

Definition 2 (Pareto frontier). Given a set of cost vectors U, we define the Pareto or efficient frontier of U, denoted by $ND(U)$, to be the set consisting of the non-dominated cost vectors of U, namely $ND(U) = \{u \in U \mid \nexists v \in U \text{ such that } v \prec u\}$. A cost vector $u \in ND(U)$ is called Pareto optimal.

4.3 Procedure

In this section, we present our main contribution, BALSA i.e., a new heuristic for streaming scalable video over wireless networks. The Pseudo code for the Heuristic is presented **Table 2**.

BALSA defines configuration sets "l_i" of "n" video layers, with "j" sub streams, j=1.........m. The Pareto-minimization procedure is invoked to calculate the dominated configuration set which returns the feasible solution of maximized video quality. Gradational Hull Pareto Minimization Heuristic essentially takes a tensor product of the configuration sets and then removes infeasible and dominated configurations. Steps 1 to 4, are used to compute the tensor product of the configuration set and remove all points which do not satisfy the resource constraints.

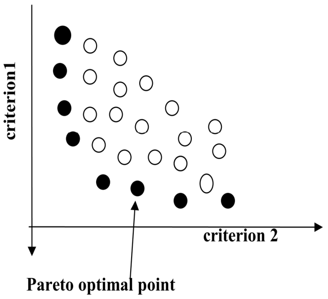

Figure 1: Pareto Optimal Point

Step 5 performs Pareto Minimization procedure - Recall the definitions of Pareto dominance and Pareto optimum which says that the efficient frontier and thereby the Pareto optimal points consist of the non-dominated configurations. In the context of our problem of video streaming, the dominated configurations can never contribute to the optimal solution of the MMKP instance. Therefore we eliminate the dominated configurations by performing the Pareto minimum procedure defined as follows:

In the array of configuration sets, if any candidate in the fractional solution is not better in any dimension than some other candidate, then it may be discarded. ie., if any candidate in the partial solution has a same or lower resource cost than some other candidate, then this may replace the candidate which is said to be a dominated configuration. All points which are non-dominant configurations are termed as Pareto points. Continue to iterate over this procedure until the new configuration set obtained is itself dominated. The resultant set is called the set of feasible solutions.

We slightly modify the convex hull algorithm to linear time for implementing the above minimisation procedure which works as follows: It uses sequential ordering of a simple polygon's edges using a "deque" (a double-ended queue) of a convex hull. It sequentially processes each of the polyline vertices in order. Let the input polyline be given by the ordered vertex set:{P_0, P_1,….., P_m}. At each stage, the algorithm determines and stores (on a double-ended queue) those vertices that form the ordered hull for all

Table 2: Pseudo Code For Heuristic Of Scalable Video Streaming

```
Procedure 1: Bandwidth Aware Layered Streaming Algorithm (BALSA)
I/p: 'l' configuration sets of size 'i', 'rs' result set of min($C_i$), 'b', a set of vector
resource constraints.
O/p: 'result', highest value of MMKP instance
       for all $l_i$; i =1 to n do
             min($l_i$)
         return rs;
         result=max(rs)
                end
Procedure 2: Gradational Hull Pareto Minimization Heuristic to compute min(li)

I/p: '$l_1$', a configuration set of video layers with substreams ranging from 1 to lx, '$l_2$',
a configuration set of video layers with substreams ranging from 1 to ly, 'b', a set of
vector resource constraints, 'p[m]', a set of all layers which do not violate the resource
constraints.
O/p: result set 'rs', a minimized, combined configuration set with only feasible
configurations.

     1. Initialize a result set (rs) = $\varphi$
          // Read the inputs of configuration set.
     2. for all $l_1 \leftarrow 1$ to $l_x$ , $l_2 \leftarrow 1$ to $l_y$ do
     3. Compute the tensor product ,l
               $l = l_1 \otimes l_2$
     4. Compute p[m] for all $l \leq b$
     5. Minimize the current feasible p[m]
         // Minimization Procedure
        // Check for dominating configurations of Pareto Points in p[m]
        dominance:= false;
        Repeat
        for all p[m] index from 1 to n do
        if ((item at index) $\prec$ (item at index+1))
        // Remove the item at index and replace with item at index+1
       // Store the current value to rs
       rs $\leftarrow$ p[m];
  dominance true;
       until dominance:= false;
  return rs;
```

polyline vertices considered so far. Then, the next vertex P_k is considered. It satisfies one of two conditions: either (1) it is inside the currently constructed hull, and can be ignored; or (2) it is outside the current hull, and becomes a new hull vertex extending the old hull.

However, in case (2), vertices that are on the list for the old hull, may become interior to the new hull, and need to be discarded before adding the new vertex to the new list. Each vertex of the vertex set can be put on the deque atmost twice (once at each end). Thereafter, elements on the deque can be removed at most once. Each of these events, to potentially add or remove a vertex to/from the deque, is associated with exactly one test. Thus, the worst case behaviour of the procedure is bounded by 3m tests and queue operations. The best case behaviour would have 2m tests and only 4 queue operations (when the initial triangle $\Delta P_0P_1P_2$ is the final hull).Thus, the procedure has a *O(m)* time and space complexity. After minimization applied to the upper bound 'n'of the configuration sets, finally find the candidate with the maximum or the highest quality value from the set of feasible solutions.

4.4 Computational Complexity

The computational complexity of our heuristic BALSA shown in table 2 is *O(mn),* where '*n*' is the number of layers in the video stream and '*m'* is the number of sub layers in each layer.

Proof:

Let us assume that there are '*n*' layers each having '*m'* items corresponding to sub layers in it, in case of different number of items per group, let ‘*p*’ be the maximum number of items in the group with '*b*' number of resource vectors. For simplicity, suppose, let $l_i=l_1 \ldots \ldots l_n$. The time complexity of the steps 1 to 4 for computing the tensor product of the configuration set and removing all points which do not satisfy the resource constraints is *O(mn)* as the loops iterate over *'n'* times for *'m'* items.

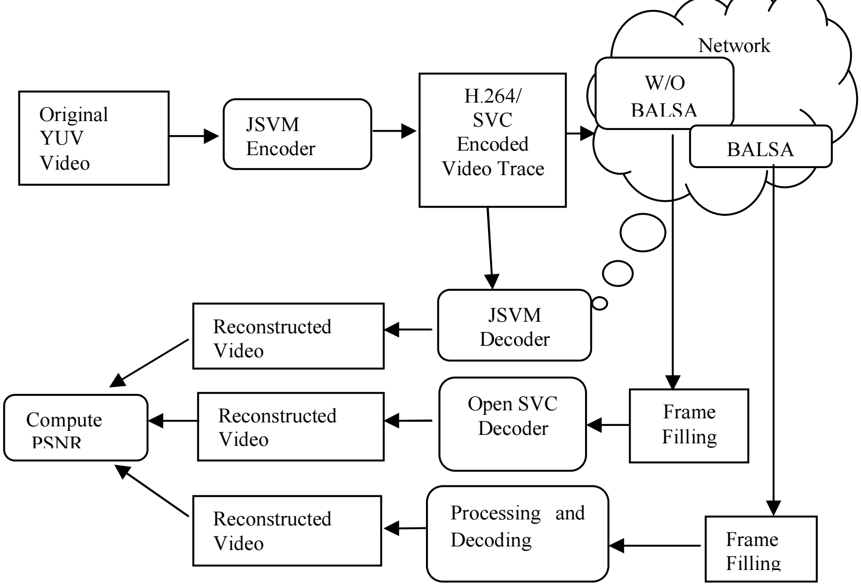

Figure 2: Simulation Setup

Finally, step 5 is a minimization procedure GHPMH which is bounded by *'m*' iterations. Therefore it requires *O(m)* running time. So the overall running time of the procedure is linear which can be deduced as follows:

$$O(nm)+O(m)=O(nm)$$

5. TRACE DRIVEN EVALUATION

5.1 Simulation Setup

Our algorithm is tested over the simulation setup provided in figure 2. For generating video traffic, raw video files are retrieved from the video trace repository of Arizona State University [16]. These files are commonly in the YUV 4CIF (704 × 576), YUV CIF (352 × 288), or QCIF (176 × 144) formats).

We encode the video data into H.264 SVC format using the open-source JSVM (Joint Scalable Video Model) reference software [17] into several layers with different encoding parameters (temporal encoding, spatial encoding, SNR encoding or combined encoding).The data rates(kbps) and the PSNR values(db) can be obtained as a separate metadata for each stream by using the PSNR static command [JSVM 2009], after passing through the Bit stream Extractor to produce the original NALU (Network Abstraction Layer Unit) trace file and extracting sub-streams of the SVC stream. The substreams represent streams with a reduced spatial and/or temporal resolution and/or a reduced bit-rate. These sub streams are then prepared to be transmitted over the network. i.e, they are streamed to heterogeneous clients from a server through a wireless network which is simulated using our own simulator configured using Java. Finally the open SVC decoder [18], a open source decoder is used to decode the streamed SVC encoded video and reconstruct the raw video sequence. As a next step we apply our proposed algorithm BALSA while streaming the video from the server to the client and compare its performance with the system without application of BALSA.

5.2 Performance Analysis

The performance of the system is analyzed under various circumstances with the metric PSNR which is a measure of the quality of video as stated below:

5.2.1 Performance Metric - Video Quality (PSNR)

Several objective image and distortion/quality metrics have been identified in literature [19]. Here we adopt the commonly used one PSNR (Peak signal-to-noise ratio).Despite it is well known that PSNR only provides an approximate measure of the quality as subjectively perceived by human observers [20], it is widely used because it is simple to calculate and has clear physical meanings. In particular, PSNR is defined as:

$$\mathrm{PSNR}=20\log_{10}(MAX_I/\sqrt{MSE})$$

where,

$$\mathrm{MSE}=\frac{1}{\mathrm{mn}}\sum_{i=0}^{\mathrm{m}-1}\sum_{j=0}^{\mathrm{n}-1}[\mathrm{I}(\mathrm{i},\mathrm{j})-\mathrm{K}(\mathrm{i},\mathrm{j})]^2.$$

$\mathrm{MAX_I}$ is the maximum possible pixel value of the image, e.g., when the pixels are represented using 8 bits per sample, $\mathrm{MAX_I}$ is 255.

5.2.2 Analysis

A comparative analysis of three different measurements of video quality is performed as elaborated below:

As H.264/SVC video encoding is an encoding scheme that results in degradation of the original quality because of the lossy compression, PSNR is computed after encoding, but before transmission, in order to appreciate the specific contribution to video distortion caused by the network. This is indicated as RRA (Reconstructed Raw Video) in figures 3a to 3c. We evaluate the performance of the system by computing the PSNR of the reconstructed erroneous video obtained by streaming over the network without application of our algorithm. This is indicated as without BALSA (w/o BALSA) in figures 3a to 3c. To prove that our algorithm improves the quality of the streamed video ,we measure the video quality of the system by computing PSNR of the streamed video over the network after application of our algorithm BALSA indicated as BALSA in figures 3a to 3c.

Our analysis show that the video quality of the resultant streamed video computed by measuring the PSNR value after applying our algorithm BALSA is higher than the video quality obtained for the streamed video without application of BALSA. This indicates the tremendous potential of using BALSA to stream scalable video over wireless networks. Also, the comparison of the above two videos of streaming with and without BALSA with the raw decoded video(RRV) shows that video distortion is caused due to errors introduced in the network.

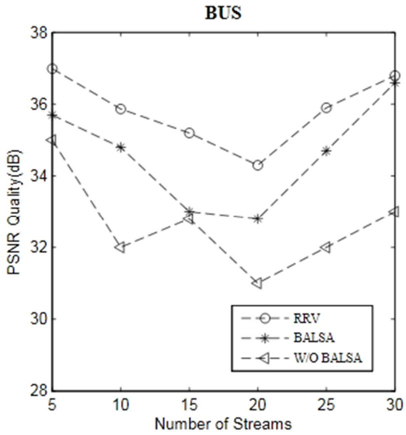

Figure 3a: Video Quality Analysis-BUS Video

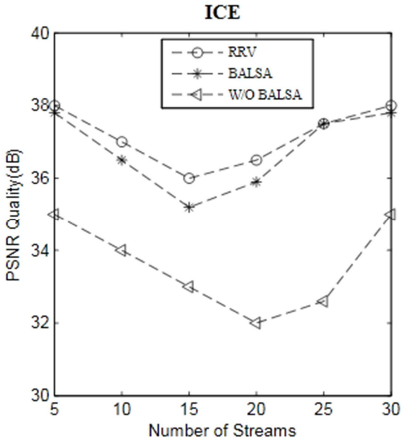

Figure 3b: Video Quality Analysis-ICE Video

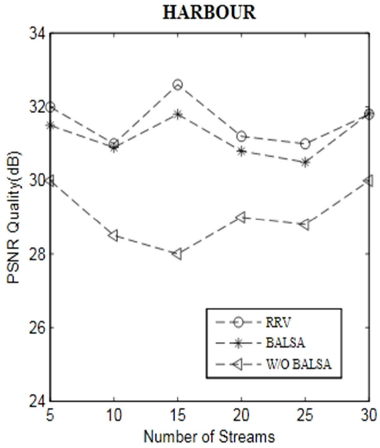

Figure 3c: Video Quality Analysis-HARBOUR Video

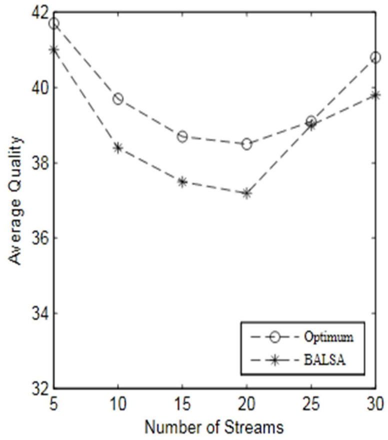

Figure 4: Average Video Quality of Test Case Videos

5.2.3 Comparison with Optimal method

To prove the efficiency of our algorithm with respect to the optimal method, we compare the performance of our algorithm with the optimal. Optimal solutions are computed using the optimization software LP solver ILOG CPLEX [21]. We note that although there are other methods of computing the optimal solution (e.g. branch and bound) they too are not suitable for real-time computation of the solutions. A plot of the same is shown in figure 4. This shows that our algorithm is 1 db near optimal.

All the above computation is performed by varying the number of video streams from 5 to 25. As indicated, Figure 4 shows the comparison of average quality across all the video streams for our algorithm and the optimal method, where as Figure 3a Figure 3b, and Figure 3c show the comparative quality analysis of the reconstructed raw video, streamed video without BALSA and with BALSA for the Bus,Ice and Harbour video sequences respectively.

5.2.4 Efficiency

We evaluate the running time of our algorithm for the input video sequences viz., Harbour, Ice, Foreman, Mobile, Soccer, News and Bus and compare the results with that of the optimum. The execution times of these sequences are measured in a Intel Pentium core i3 processor with 4 GB memory. The results of the experiment for our algorithm BALSA for various test sequences are shown in figure 5. Also a plot of the execution time of BALSA and the optimal method is shown in figure 6. The results indicate that computing the optimum solution using ILOG CPLEX takes much longer time as the number of streams grow compared to our proposed algorithm BALSA.

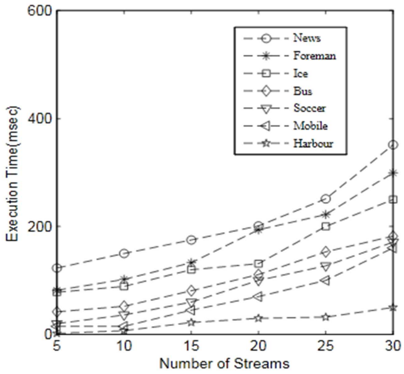

Figure 5: Running Time of Test Case Videos

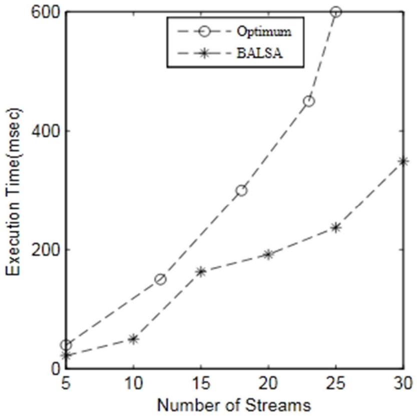

Figure 6: Running Time of the Algorithm

6. CONCLUSION

This paper presents BALSA, a parameterized gradational heuristic for streaming video over wireless networks. It is proved that GHPMH can find good quality solutions for streaming video over wireless networks with multiple resource constraints such as varying client capabilities, varying network bandwidth.Also, our analysis show that the computational complexity of our algorithm is O(nm) linear time which makes it suitable for real time applications like video streaming. Simulation results show that our algorithm finds solutions which are close to the optimal (within 1db) under realistic conditions.

REFRENCES:

[1] H.Schwarz, D. Marpe, and T. Weigand, “Overview of the Scalable video coding extension of the H.264/AVC standard”, *IEEE Trans. Circuits Syst. Video Technol*, Vol. 17, No.9, Sept. 2007, pp. 1103-1120.

[2] M. Wien, H. Schwarz, and T. Oelbaum, “Performance analysis of SVC”, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 17, No.9, Sep. 2007, pp. 1194-1203.

[3] T. Schierl, and T. Wiegand, “Mobile video transmission using SVC”, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 17, No.9, Sep. 2007, pp. 1204 - 1217.

[4] C.A. Segall and G. J. Sullivan, “Spatial scalability within the H.264/AVC Scalable video coding extension,” *IEEE Trans. Circuits Syst. Video Technol.*, Vol.17, No. 9, Sep. 2007, pp. 1121-1135.

[5] H. Schulzrinne, S. Casner, R. Fredrick, and V. Jacobson, “RTP: A Transport for Real-Time Applications,” *IETF STD 0064, RFC 3550*, Jul. 2003.

[6] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard”, *IEEE Trans. Circuits Syst. Video Technol.* Vol. 13 No. 7, 2003, pp.560-576.

[7] S. Martello, P. Toth, “Algorithms for knapsack problems,” *Annals of Discrete Mathematics*, Vol. 31, 1987, pp. 70–9.

[8] S. Khan, “Quality adaptation in a multi-session multimedia system: model and architecture,” PhD dissertation. Department of Electrical and Computer Engineering, University of Victoria, 1998.

[9] MM Akbar, M. S Rahman, M. Kaykobad, E.G. Manning, and G.C. Shoja, “Solving the multidimensional multiple-choice Knapsack problem by constructing convex hulls,” *Comput Oper Res.*, Vol. 33, 2006, pp. 1259–1273.

[10] M. Geilen, T. Basten, B.D. Theelen, and R. Otten, “An algebra of Pareto Points,” Fundamenta Informaticae, Vol. 78, No. 1, 2007, pp. 35-74.

[11] P. Hosein, “Broadcasting VBR traffic in a WiMAX network,” in *Proc. of IEEE Vehicular Technology Conference (VTC’08) Calgary, Canada*, September 2008, pp. 1–5.

[12] P. Koleser, “A branch and bound algorithm for knapsack problem,” *Management Science*, Vol. 13, 1967, pp. 723–35.

[13] R. Nauss, “The 0-1 knapsack problem with multiple choice constraints,” *European Journal of Operation Research*, Vol. 2, No. 2, 1978, pp. 125–31.

[14] W. Shih, “A branch and bound method for multiconstraint knapsack problem,” *Journal of the Operational Research Society*, Vol. 30, 1979, pp. 369–78.

[15] V. Reguant, F. Prats, R. de Pozuelo, F. Margalef, and G. Ubiergo, “Delivery of H.264 SVC/MDC streams over WiMAX and DVB-T networks,” in *IEEE International Symposium on Consumer Electronics (ISCE’08), Algarve, Portugal*, April 2008, pp. 1–4.

[16] M. Reisslein, YUV Video sequences. Video Traces Research Group, http://trace.eas.asu.edu/yuv/ .

[17] “H.264/SVC Reference Software JSVM,” 2006, http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm.

[18] M. Blestel, Open SVC Decoder Website," http://sourceforge.net/projects/opensvcdecoder/, 2009.

[19] Y. Wang, “Survey of Objective Video Quality Measurements,” *Tech report, Worcester Polytechnic Institute*, June 2006.

[20] Definition of Quality of Experience (QoE), Reference: TD 109rev2 (PLEN/12), ITU – International Telecommunication Union.

[21] ILOG, Ilog cplex 10, http://www.ilog.com/products/cplex/ (2006).