

# GRADIENT ASCENT PAIRED-COMPARISON SUBJECTIVE QUALITY TESTING

*Stephen Voran and Andrew Catellier*

Institute for Telecommunication Sciences  
325 Broadway, Boulder, Colorado, USA 80305  
{svoran,acatellier}@its.bldrdoc.gov

## ABSTRACT

Subjective testing is the most direct means of assessing audio, video, and multimedia quality as experienced by users and maximizing the information gathered while minimizing the number of trials is an important goal. We propose gradient ascent subjective testing (GAST) as an efficient way to locate optimizing sets of coding or transmission parameter values. GAST combines gradient ascent optimization techniques with paired-comparison subjective test trials to efficiently locate parameter values that maximize perceived quality. We used GAST to search a two-dimensional parameter space for the known region of maximal audio quality as proof-of-concept. That point was accurately located and we estimate that conventional testing would have required at least 27 times as many trials to generate the same results.

**Index Terms**— Audio Quality, Gradient Ascent, Golden Section Line Search, Multimedia Quality, Subjective Testing, Video Quality

## 1. INTRODUCTION

Subjective testing is arguably the most basic and direct way to assess the user-perceived quality of audio, video, and multimedia presentations. Through careful selection of signals, presentation environments, presentation protocols, and test subjects, one can approximate a real-world scenario and acquire a representative sample of user perceptions for that scenario. This generally requires specialized equipment, software, laboratory environments, skills, and numerous human test subjects [1]. These elements equate to significant expenses and weeks or months of work. Objective estimators of perceived quality can eliminate many expenses and complications inherent in subjective testing but they produce only estimates of perceived quality [1].

Between these poles lies another option: subjective testing with improved efficiency. That is, gathering more information using fewer experimental trials.

One quality assessment task that can be particularly intensive is optimization of one or more coding or transmission parameters. Given a bit-rate constraint, one might seek to optimally partition those bits between different signals (e.g., au-

dio, video, data), between basic signal coding and redundancy that improves robustness to transmission errors or losses (e.g., multi-descriptive coding or forward error correction), or between the different components of a reduced-rate signal representation for an individual signal (e.g., quantized scale factors and quantized transform coefficients in an audio coder).

Such optimization problems can be solved by an exhaustive search (ES) of a discretized version of the parameter space using an absolute category rating (ACR) subjective test to evaluate each point in the space. But this can require the evaluation of a very large number of points and it also requires one to guess at how to best discretize the parameter space.

We propose gradient ascent subjective testing (GAST) as an efficient alternative to ES ACR testing. GAST can efficiently and adaptively select a subset of points in the space to evaluate, without the need to impose an arbitrary discretization on the space. GAST can incorporate the ACR approach but is particularly well-matched to paired-comparison (PC) testing. Using PC testing is an added bonus because comparing two stimuli is easier (and thus less noise is introduced into results) than providing absolute ratings for two stimuli presented in isolation from each other.

In Section 2, we describe the GAST algorithm. Section 3 details an initial proof-of-concept experiment using the GAST algorithm. Discussion and observations are provided in Section 4.

## 2. GRADIENT ASCENT PAIRED-COMPARISON SUBJECTIVE TESTING ALGORITHM

Finding the point in  $n$ -dimensional space that approximately maximizes (or minimizes) an objective function defined on that space is a classic problem and many different avenues to its solution have been offered over the years [2], [3]. A unifying key idea is to evaluate the objective function at a small number of intelligently selected points, use those results to select more points, and thus continue to better locate the desired maximal point. Key attributes of solutions include convergence properties and efficiencies.

We adopt this basic idea to the problem of optimizing perceived quality on an  $n$ -dimensional parameter space. In this

case, the objective function is perceived quality and is evaluated by human subjects. Thus a GAST algorithm implementation platform includes a computer and one or more human subjects. Software calculates a pair of points in the parameter space where the objective function (perceived quality) should be evaluated, and then facilitates the presentation of stimuli associated with this pair of points. The subject evaluates this pair of stimuli and the software uses the response to then calculate the next pair of points to evaluate. The software and the subject continue this interplay until termination criteria indicate it is likely that point of maximum quality has been located.

This approach could be applied to any number of optimization algorithms. We have elected to start with a very basic gradient ascent algorithm because it seems well-matched to expected properties of actual applications (i.e., smooth, slowly varying objective functions with fairly broad maxima that can only be imprecisely evaluated). The GAST algorithm iterates between two main steps: finding the direction of steepest ascent, and searching a line for a maximum. Each of these steps requires PC scores from a test subject.

## 2.1. Subjective Scores

The GAST algorithm described here uses PC scores. This means that two stimuli are presented and a subject indicates any preference between the two. For visual stimuli, either sequential or side-by-side presentations are possible. Another option is to employ an A/B switch that allows the subject to switch between the two stimuli at will. For auditory stimuli, the options are sequential presentation and A/B switching.

Our initial experiment was auditory. We used sequential presentation and allowed for five possible responses: “The audio quality of the second recording is much better than, better than, the same as, worse than, or much worse than, the first recording.” The algorithm associates these five responses with the integers 2, 1, 0,  $-1$ , and  $-2$ , respectively. These integers are called subjective scores.

Without loss of generality, we adopt the language of sequential presentation for the remainder of this paper. We use  $S(\mathbf{x}, \mathbf{y})$  to represent the subjective score resulting from a presentation of the signal parameterized by the vector  $\mathbf{x}$  (representing a point in  $n$ -dimensional space), followed by presentation of the signal parameterized by the vector  $\mathbf{y}$ . Thus positive values of  $S(\mathbf{x}, \mathbf{y})$  indicate that the  $\mathbf{y}$  signal was preferred to the  $\mathbf{x}$  signal, negative values indicate the opposite, and zero indicates that there was no preference.

## 2.2. Direction Finding

Consider a point in an  $n$ -dimensional space represented by a column vector  $\mathbf{x}$ . We seek to find the direction in which the objective function increases most rapidly. The direction-finding algorithm finds an approximate solution using between  $n$  and  $2 \cdot n$  finite differences. Let

$$\mathbf{x}_k^\pm = \mathbf{x} \pm \Delta_d \cdot \mathbf{I}^k, \quad k = 1, 2, \dots, n, \quad (1)$$

indicate a point near  $\mathbf{x}$  differing from  $\mathbf{x}$  in only the  $k^{\text{th}}$  dimension. In (1),  $\Delta_d$  is a fixed scalar direction-finding step size and  $\mathbf{I}^k$  is the  $k^{\text{th}}$  column of the  $n \times n$  identity matrix.  $\Delta_d$  needs to be large enough to cause detectable changes in perceived quality, but small enough to provide accurate localized information about those changes.

The direction-finding algorithm gathers subjective scores  $S(\mathbf{x}, \mathbf{x}_k^\pm)$  for each dimension  $k$ , as allowed. If the parameter space is bounded,  $\mathbf{x}_k^+$  or  $\mathbf{x}_k^-$  could be outside the parameter space, the corresponding signal would not exist, and the corresponding subjective score would not exist. If only one subjective score exists for dimension  $k$ , then the corresponding element  $\delta_k(\mathbf{x})$  of the direction vector  $\boldsymbol{\delta}(\mathbf{x})$  is given by

$$\delta_k(\mathbf{x}) = \frac{S(\mathbf{x}, \mathbf{x}_k^\pm)}{\pm \Delta_d}. \quad (2)$$

For dimensions where both subjective scores exist,  $\delta_k(\mathbf{x})$  is given by

$$\delta_k(\mathbf{x}) = 0, \text{ when } S(\mathbf{x}, \mathbf{x}_k^-) < 0 \text{ and } S(\mathbf{x}, \mathbf{x}_k^+) < 0, \quad (3)$$

$$\delta_k(\mathbf{x}) = \frac{S(\mathbf{x}, \mathbf{x}_k^+) - S(\mathbf{x}, \mathbf{x}_k^-)}{2\Delta_d}, \text{ otherwise.} \quad (4)$$

Equation (3) treats the special case where  $\mathbf{x}$  is located at a maximum in dimension  $k$ . Equation (4) treats the general case where two subjective scores are available and uses them together to approximate an average local slope in dimension  $k$ . Once  $\delta_k(\mathbf{x})$  has been calculated for all  $n$  dimensions, the resulting direction vector  $\boldsymbol{\delta}(\mathbf{x})$  is scaled to have unit norm:

$$\hat{\boldsymbol{\delta}}(\mathbf{x}) = \frac{\boldsymbol{\delta}(\mathbf{x})}{|\boldsymbol{\delta}(\mathbf{x})|}. \quad (5)$$

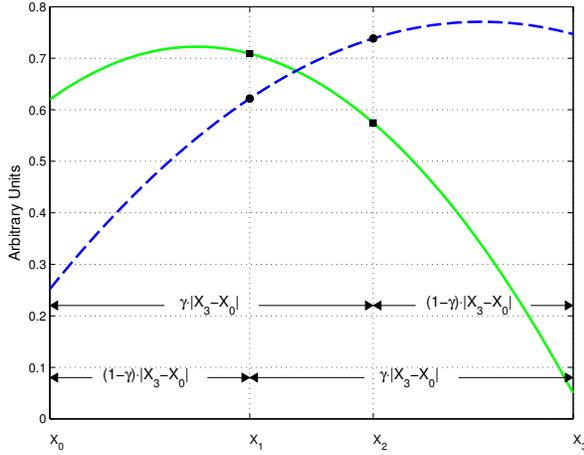
The result is a unit-norm vector  $\hat{\boldsymbol{\delta}}(\mathbf{x})$  that provides an approximate indication of the direction in which the objective function increases most rapidly. It is an approximate result because it is based on finite differences in the parameter space, and because the subjective scores are constrained to five distinct values. The impact of this approximation will depend on the specific context in which GAST is used. Our proof-of-concept experiment appears unhindered by this approximation.

## 2.3. Golden Section Line Search

Given an arbitrary line segment in parameter space, the iterative line search algorithm in GAST finds the point on that line segment that approximately maximizes the objective function. The algorithm is initialized by a point represented by the column vector  $\mathbf{x}_0$ , a unit-norm direction vector  $\hat{\boldsymbol{\delta}}(\mathbf{x}_0)$ , and a boundary definition for the parameter space. The first step is to find the line segment (or “line” for brevity) that runs in

the direction  $\hat{\delta}(\mathbf{x}_0)$  from  $\mathbf{x}_0$  to the boundary of the parameter space. We call the second end of this line  $\mathbf{x}_3$ .

This line is the input to the iterative portion of the algorithm. Each iteration results in a new, shorter line that is evaluated on the next iteration. This evaluation is based on the comparison of the objective function at two interior points that lie on this line. These points are called  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and are ordered as shown in Figure 1. If  $S(\mathbf{x}_1, \mathbf{x}_2) < 0$  (consistent with the example of the solid line) then the new line to search on the next iteration is the line between  $\mathbf{x}_0$  and  $\mathbf{x}_2$ . If  $0 < S(\mathbf{x}_1, \mathbf{x}_2)$  (consistent with the example of the broken line) then the new line to search is the line between  $\mathbf{x}_1$  and  $\mathbf{x}_3$ .



**Fig. 1.** Example relationships for four points in the line search.

Motivated by a desire for predictable convergence, we add the constraint that each iteration must scale the line down by a constant value  $0 < \gamma < 1$ , regardless of which interval is chosen as the new interval. This means that

$$|\mathbf{x}_2 - \mathbf{x}_0| = |\mathbf{x}_3 - \mathbf{x}_1| = \gamma |\mathbf{x}_3 - \mathbf{x}_0| \quad (6)$$

and

$$|\mathbf{x}_1 - \mathbf{x}_0| = |\mathbf{x}_3 - \mathbf{x}_0| - |\mathbf{x}_3 - \mathbf{x}_1| = (1 - \gamma) |\mathbf{x}_3 - \mathbf{x}_0| \quad (7)$$

Regardless of the subjective score, the new shorter line (between  $\mathbf{x}_0$  and  $\mathbf{x}_2$  or between  $\mathbf{x}_1$  and  $\mathbf{x}_3$ ) always inherits an interior point from the longer line ( $\mathbf{x}_1$  in first case and  $\mathbf{x}_2$  in the second case). Motivated by a desire to use paired comparisons efficiently, we add the constraint that this inherited (from iteration  $i$ ) interior point must be one of the two interior points evaluated in iteration  $i + 1$ .

Consider the case where the result of iteration  $i$  is the line between  $\mathbf{x}_0$  and  $\mathbf{x}_2$  (consistent with the solid line in the example of Figure 1). That new shorter line inherits the interior point  $\mathbf{x}_1$ . In iteration  $i + 1$  a second interior point must be added. If this new point is inserted to the left of  $\mathbf{x}_1$ , then  $\mathbf{x}_1$

would now (iteration  $i + 1$ ) serve the role that  $\mathbf{x}_2$  played in iteration  $i$ . Using (6) we conclude that

$$|\mathbf{x}_1 - \mathbf{x}_0| = \gamma^2 |\mathbf{x}_3 - \mathbf{x}_0| \quad (8)$$

Comparing (7) and (8) we conclude that

$$\gamma^2 = (1 - \gamma) \text{ so } \gamma = \frac{-1 + \sqrt{5}}{2}. \quad (9)$$

Finally,

$$\frac{1}{\gamma} = \gamma + 1 = \frac{1 + \sqrt{5}}{2} = \varphi \approx 1.618. \quad (10)$$

If the new point is inserted to the right of  $\mathbf{x}_1$ , then  $\mathbf{x}_1$  would now (iteration  $i + 1$ ) serve the same role that it played in iteration  $i$ . Using (6) and (7) we conclude that

$$|\mathbf{x}_1 - \mathbf{x}_0| = (1 - \gamma) |\mathbf{x}_3 - \mathbf{x}_0| = (1 - \gamma) \gamma |\mathbf{x}_3 - \mathbf{x}_0|, \quad (11)$$

but this can only be solved by  $\gamma = 1$  which violates the allowed range on  $\gamma$ . Thus the new point must be inserted to the left of  $\mathbf{x}_1$ .

If iteration  $i$  produces the line between  $\mathbf{x}_1$  and  $\mathbf{x}_3$  (consistent with the broken line in the example of Figure 1), an analogous set of results will follow. Thus  $\gamma = 1/\varphi$  is the only value to use in equations (6) and (7) to locate  $\mathbf{x}_1$  and  $\mathbf{x}_2$  so that the uniform-scaling-per-iteration constraint and the interior-point-reuse constraint are satisfied. The line to search scales by  $\gamma = 1/\varphi$  at each iteration. The irrational number  $\varphi$  is called the golden section or golden mean. It defines an aesthetically pleasing rectangle that has been used widely in architecture and art, and also lends its name to this line search algorithm [2].

In GAST this golden section line search iterates until  $S(\mathbf{x}_1, \mathbf{x}_2) = 0$  and  $|\mathbf{x}_2 - \mathbf{x}_1| < \Delta_t$ , where  $\Delta_t$  is a termination parameter. This condition indicates there is no preference between two signals whose parameterizations are sufficiently close to each other. The algorithm returns  $\frac{1}{2}(\mathbf{x}_2 + \mathbf{x}_1)$  as the approximation to the point on the original line where the objective function is maximized. Our proof-of-concept experiments indicate that the approximation is a good one. If  $S(\mathbf{x}_1, \mathbf{x}_2) = 0$  when  $\Delta_t \leq |\mathbf{x}_2 - \mathbf{x}_1|$ , then  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are moved apart in increments until a non-zero vote is returned. This is a special case that breaks from the golden section constraints.

## 2.4. Entire Algorithm

To start the GAST algorithm, one must select a starting point  $\mathbf{x}_0$  in the  $n$ -dimensional parameter space. (In Section 3 we successfully use both deterministic points on the boundary of the space and randomly selected interior points.) The direction-finding algorithm is applied to find  $\hat{\delta}(\mathbf{x}_0)$  indicating the direction of steepest ascent from  $\mathbf{x}_0$ . Next,  $\mathbf{x}_0$  and  $\hat{\delta}(\mathbf{x}_0)$

are provided to the line search algorithm, which searches in the direction  $\hat{\delta}(\mathbf{x}_0)$  from  $\mathbf{x}_0$  to the boundary of the search space and returns the maximizing point  $\mathbf{x}_1$ .

The direction-finding algorithm is then used to find  $\hat{\delta}(\mathbf{x}_1)$ , which shows the direction of steepest ascent from  $\mathbf{x}_1$ . Line searching and direction finding continue to alternate in this fashion until a terminating condition is satisfied. At any iteration, the output of the last line search is the best approximation to the point in the parameter space that maximizes the objective function.

One terminating condition is  $\hat{\delta}(\mathbf{x}_i) = \mathbf{0}$  since this indicates that there no is direction to move from  $\mathbf{x}_i$  to increase the objective function. Equations (2) through (4) show that this could be due to subjective scores of zero (no differences detected), a local maximum, or a local minimum that is judged to be perfectly symmetrical in all  $n$  dimensions. Terminating in a local minimum is not desirable, so if this is deemed a possibility one should test for it (the test is analogous to the one in equation (3)) and restart the GAST algorithm from a new starting point as necessary. The algorithm also terminates if the distance between the input and output points of a line search is less than  $\Delta_t$  since future iterations will be unlikely to move the result outside that neighborhood.

The GAST algorithm climbs the surface of the objective function to find a maximal value. If multiple local maxima exist, the algorithm will find one of them but there is no guarantee that it will be the global maximum. If multiple local maxima are suspected, then multiple trials using multiple starting places could help to identify them. Other more sophisticated algorithms might be considered as well.

### 3. PROOF-OF-CONCEPT EXPERIMENT

To test the GAST concept we devised an experiment using reference conditions and a 2-dimensional parameter space with a known region of maximal audio quality.

#### 3.1. Experiment Description

This experiment used eight diverse five-second musical segments taken from compact discs. A sample rate of 44,100 samples/second was maintained through this experiment. The segments were passed through two reference conditions in sequence. The first reference condition was the modulated noise reference unit (MNRU) [4]. This condition adds signal-correlated Gaussian noise to the audio signal at the specified SNR of  $Q$  dB:

$$y_k = x_k + x_k \cdot n_k \cdot 10^{-\frac{Q}{20}} = x_k \cdot \left(1 + n_k \cdot 10^{-\frac{Q}{20}}\right), \quad (12)$$

where  $x_k$ ,  $y_k$  and  $n_k$  are input, output, and unit-variance zero-mean Gaussian noise samples, respectively. The noise added by the MNRU sounds like that produced by some waveform coders.

The second reference condition was modeled after the T-Reference described in [5, 6]. We used a frame size of 256 samples (5.8 ms). If frames are labeled 1 through  $N$ , then the T-Reference applies temporal compression to frames numbered  $1 + 3 \cdot k$ , it does not change frames numbered  $2 + 3 \cdot k$ , and it applies temporal expansion to frames numbered  $3 + 3 \cdot k$ ,  $k = 0, 1, 2, \dots$ . Temporal compression is accomplished by deleting every  $T^{th}$  sample, and the complementary temporal expansion is accomplished by interpolating a sample between every  $T^{th}$  and  $T + 1^{st}$  sample. Since  $\lfloor 256/T \rfloor$  samples are deleted from the first frame in the group and the same number of samples are interpolated into the third frame in the group, the total number of samples in each group of three frames is preserved at  $3 \cdot 256$ .

Valid values of the unit-less parameter  $T$  are integers in the range from 2 to 256. The distortion introduced by the T-Reference is described as “warbling” or “burbling” and is similar to that produced by some parametric coders. Larger values of  $T$  correspond to less distortion.

We developed GAST software to work in a normalized  $[0, 1]$  parameter space. Thus, we mapped this range to  $Q$  and  $T$  values according to

$$Q = -85 \cdot p_1^2 + 100 \cdot p_1 \quad (13)$$

and

$$T = 1 + \left\lceil 2^{(-15 \cdot p_2^2 + 13 \cdot p_2 + 2)} \right\rceil \quad (14)$$

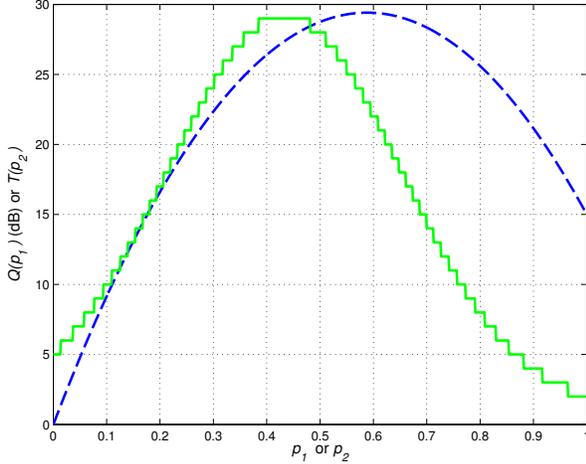
where  $\lceil \cdot \rceil$  denotes rounding to the nearest integer. These relationships are displayed in Figure 2. They were selected to smoothly traverse a wide range of  $Q$  and  $T$  values, have different shapes, asymmetric slopes, and a single interior maximum for both  $Q$  and  $T$ .

From Figure 2 we can conclude that in the two-dimensional space  $(p_1, p_2)$ , there is a line segment of numerically maximal audio quality extending from the point (0.60,0.39) to the point (0.60,0.48). This segment is shown as a solid vertical line in Figures 3 and 4.

#### 3.2. GAST Algorithm Implementation

The direction finding and the golden section line search algorithms were coded inside objects called “tunes” such that all calculations take place transparently to an outer algorithm that facilitates subject interaction. The outer algorithm needs only to instantiate said tunes by specifying  $\mathbf{x}_0$ ,  $\Delta_d$  and  $\Delta_t$ , request parameter pairs associated with the signal pairs that are played, submit subjective scores, and keep track of all tune objects that it instantiated.

The outer algorithm is also responsible for drawing a graphical user interface to be used by the subject, as well as instantiating, polling, and updating necessary tune objects, presenting audio to subjects, handling subject votes, randomizing tune play order, and ensuring that each search terminates. The MNRU and T-Reference algorithms execute



**Fig. 2.**  $Q$  as a function of  $p_1$  (dashed), and  $T$  as a function of  $p_2$  (solid).

quickly, so it was possible to generate the required audio signals just before they were played.

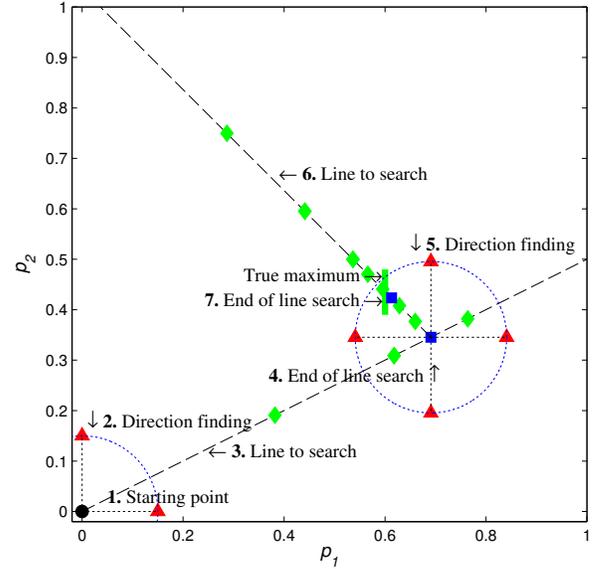
Subjects heard signals through headphones and submitted votes using a PDA. After the presentation of each pair of signals, a subject could submit a vote (as described in Section 2.1) or request to hear the pair played again. Our GAST software is available at [www.its.bldrdoc.gov/audio](http://www.its.bldrdoc.gov/audio) for those who wish to experiment with the GAST technique.

### 3.3. Experiment Results

Six persons participated in the experiment. Each ran the GAST algorithm on four of the eight musical selections, using two different starting places per selection. One starting place was the origin of the parameter space, the other was randomly chosen for each musical selection and each subject. Thus, each subject started eight different GAST tasks, and in each trial the subject made one step of progress on one task randomly selected from the eight. We used the direction-finding step size  $\Delta_d = 0.15$  and the terminating condition  $\Delta_t = 0.20$ .

Some tasks ended prematurely due to operational errors, subject time limitations, or saturation of perceived quality (at the low end) near the corners of the parameter space. Beyond these special cases, the GAST algorithm behaved as expected.

Figure 3 shows an example GAST task trajectory. The region of numerically maximal audio quality is shown with a bold vertical line. The circle at the origin indicates the starting location. The triangles connected to that circle indicate the two points used in the first direction-finding step. The audio signal parameterized by the triangle at  $(0.15, 0)$  was voted "much better" than the signal associated with the origin, so  $S((0, 0)^T, (0.15, 0)^T) = 2$ , where  $(\cdot)^T$  indicates the transpose operator. Similarly  $S((0, 0)^T, (0, 0.15)^T) = 1$ .

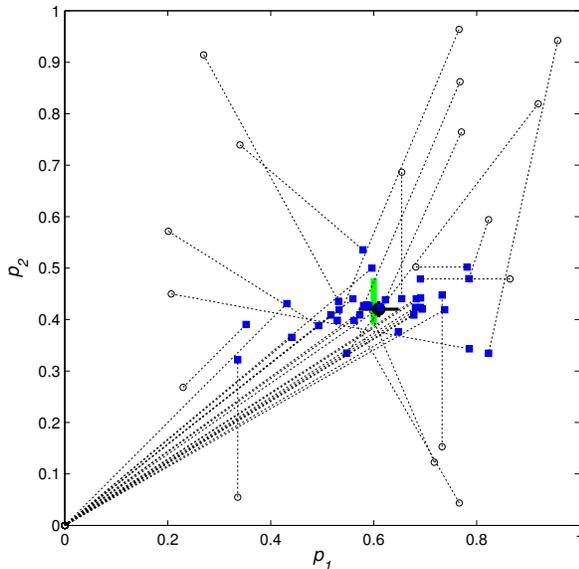


**Fig. 3.** Example trajectory of a GAST trial; details in text.

These two scores yielded the normalized direction vector  $\hat{\delta}(\mathbf{x}) = (1/\sqrt{5}) \cdot (2, 1)^T$  and this led to a search of the line that runs up and to the right. Points played on this line are shown with diamonds and the result of the line search is shown with a square. The four points connected to that square were played as part of the second direction-finding step. This led to a search of the line that runs toward the upper left corner of the figure. Again, points played are shown with diamonds and the final result is shown with a square. This result is very close to the location of numerically maximum audio quality. This task required 13 votes.

Different musical selections can reveal or mask distortions in different ways, and these distortions may be perceived differently by individual subjects. Thus, perceived quality is a function of signals and subjects as well as the device under test. Averaging results over a representative sample of relevant signals and subjects gives the most meaningful perceived quality results.

Figure 4 shows the GAST algorithm start (open circles) and end (solid squares) points for the 35 GAST tasks that ran to completion. An average 15.6 votes were required per task. The end points cluster around the line segment of numerically maximal audio quality (the solid vertical line), as expected. The mean and 95-percent confidence intervals for the  $p_1$  and  $p_2$  dimensions are shown with a circle and a cross. For the 35 combinations of subjects and musical selections, we are 95 percent confident that the mean location of maximal perceived audio quality is between 0.571 and 0.649 in  $p_1$  dimension, and between 0.404 and 0.436 in the  $p_2$  dimension. This result is consistent with the known location of numerically maximal audio quality and required  $15.6 \times 35 = 546$  PC presentations (not including any replays) and 546 votes.



**Fig. 4.** Start and end points for 35 GAST trials shown with open circles and solid squares, respectively. The cross shows mean and 95-percent confidence intervals for all end points. Bold vertical line represents region of numerically maximal audio quality.

To locate this point with the same resolution using ES ACR testing, one would need about 13 samples  $((0.649 - 0.571)^{-1} = 12.8)$  in the  $p_1$  dimension and 32 samples  $((0.436 - 0.404)^{-1} = 31.3)$  in the  $p_2$  dimension, resulting in a 416 sample grid on the parameter space. Evaluating each point with all 35 combinations of musical selections and subjects would require  $416 \times 35 = 14,560$  ACR presentations (not including any replays) and votes. This is a lower bound. If 35 trials per point in the parameter space does not result in statistically significant differences between adjacent parameter space samples in the neighborhood of the quality maximum, then additional trials would be required to locate the maximum with a resolution that matches GAST.

Thus, we find that the number of votes required is reduced by at least a factor of  $14,560/546 = 26.7$ . Two signals must be played for a PC presentation, and one is played for an ACR presentation, so the number of presentations is reduced by at least a factor of 13.4.

#### 4. DISCUSSION AND OBSERVATIONS

We have demonstrated proof-of-concept for the GAST approach in a simple, controlled 2-dimensional case with a known region of maximal audio quality. The correct result was obtained. Compared with the hypothetical comparable ES ACR subjective test, votes and presentations were reduced by factors of at least 27 and 13, respectively. One would expect these savings to increase in higher dimensional

problems. Additionally, GAST eliminates the need to select step sizes to use in ES. An option between ES and GAST would be to perform a coarse, broad ES that identifies a region where one should follow up with a fine, narrow ES. But this requires manual intervention and manual selection of multiple step sizes and search ranges.

This initial experiment admittedly uses a well-behaved quality surface, and GAST results will likely vary depending on the shape of the parameter space and attributes of the quality surface over that space. This paper presents only an initial effort. There are many potential paths to improved GAST performance, efficiency, and robustness.

Given additional time and data, one might undertake refinement of the terminating conditions, possibly making them adaptive. One might make line lengths adaptive, searching shorter lines as the algorithm progresses, since the start of the line should be getting closer to the sought-after point of maximal quality. The direction finding step size  $\Delta_d$  might be advantageously adapted as the algorithm progresses (larger early on or when in flatter regions, smaller later or in steeper regions). When finding points of minimal quality is of interest, one can simply multiply all votes by  $-1$  and the GAST algorithm will locate minima instead of maxima.

GAST can be used with naïve or expert subjects. Expert subjects might benefit from additional information as the test progresses. Since the end point of each line search is the current approximation to the point of maximal quality, experts might be able to improve their GAST efficiency if they can respond to a message that says, “You have just completed the your  $n^{th}$  line search for this task. Would you like to begin another one?”

#### 5. REFERENCES

- [1] S. Voran, “Estimation of speech intelligibility and quality,” in *Handbook of signal processing in acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds., vol. 2, chapter 28, pp. 483–520, Springer, New York, 2008.
- [2] B. Gottfried and J. Weisman, *Introduction to optimization theory*, Prentice Hall, Englewood Cliffs, NJ, 1973.
- [3] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, second edition, 2006.
- [4] *ITU-T Recommendation P.810, Modulated noise reference unit (MNRU)*, Geneva, 1996.
- [5] B. Cotton, “New reference condition for very low bit rate voice coder evaluation,” CCITT SGXII Contribution D.108, September 1991.
- [6] S. Voran, “Observations on the t-reference condition for speech coder evaluation,” CCITT SGXII Contribution SQ 13.92, February 1992, available: [www.its.bldrdoc.gov/audio](http://www.its.bldrdoc.gov/audio).