Latin Hypercube Sampling and Pattern Search in Magnetic Field Optimization Problems

Kent R. Davey, *Fellow, IEEE*

Abstract—**Latin hypercube is a sampling technique for searching n dimensional space. Like Monte Carlo methods, it retains random qualities, and yet Latin hypercube is consistently more effective than Monte Carlo. Despite this fact, not a single paper has been published in IEEE Magnetics on its use. Field analysis is a long way from delivering vectorized solutions where a vector of inputs can be processed. Stochastic algorithms are exceptionally inefficient compared to their deterministic counterparts. The best optimization tool would be a deterministic method which quickly and effectively interrogates the search space. Latin hypercube sampling, combined with pattern search solutions, comes close to achieving that objective. An improved solution for the magnetic TEAM workshop problem 22 is presented using these tools.**

Index Terms—**Latin hypercube, multiple minima, optimization, pattern search.**

I. INTRODUCTION

As FIELD analysis codes become more sophisticated, attention is increasingly focused on how best to use available analysis tools to improve devices. Multi-variable optimizations are characterized by multiple local minima. Although stochastic algorithms, such as simulated annealing and genetic algorithms, have some success in avoiding local minima, their execution time is an order of magnitude longer than deterministic methods and the chance of a global minimum is only enhanced, not guaranteed. Professional stochastic algorithms offer a hybrid analysis. Here the stochastic algorithm serves only as the starting function. The switch over point is uncertain and the method does not guarantee location of the global minimum.

Latin hypercube sampling more effectively samples the search space, but provides the randomness required to investigate effectively. The technique is easily employed by all deterministic methods. In addition, the new pattern search tools can gainfully use the technique for intermediate search during mesh expansion and contraction. Latin hypercube sampling applies to both uniform bounds and infinite bounds, where the latter involves an inverse error function to get the sample points.

Manuscript received August 10, 2006.

K. R. Davey is with the Center for Electromechanics, University of Texas, 10100 Burnet Rd., Bldg. EME 133, Austin, TX 78758, USA (phone: 512-232-1603; fax: 512-475-7700; e-mail: k.davey@mail.utexas.edu).

II. LATIN HYPERCUBE SAMPLING

Latin hypercube sampling (LHS) is a form of stratified sampling that can be applied to multiple variable optimizations to circumvent the shortcomings of the best deterministic methods [1][2]. Only four journal papers have ever appeared in IEEE discussing the use of Latin hypercube, and none in IEEE Magnetics. Three of the four focus on statistical modeling [3][4][5]. Regis targets an optimization problem using Latin hypercube as a means for representing his investigation space with a multivariate function, specifically, a radial basis function [6]. He does not optimize the original function, but only the fit function obtained using a Latin hypercube approach.

The Latin hypercube method is commonly used to reduce the number of runs necessary for a Monte Carlo simulation to achieve a reasonably accurate random distribution [7]. Stratified sampling is a method of searching a population to guarantee a subpopulation spread consistent with the probability of the various subgroups. Random sampling occurs within the subgroups or strata. This procedure reduces the number of runs necessary to sufficiently sample a population of multiple variables. In its simplest form, Latin hypercube is a technique for efficiently choosing the initial guess in conjunction with a deterministic algorithm, increasing the confidence level that a global minimum has been found with minimum effort. But it can be integrated into the main function of other approaches, such as pattern search.

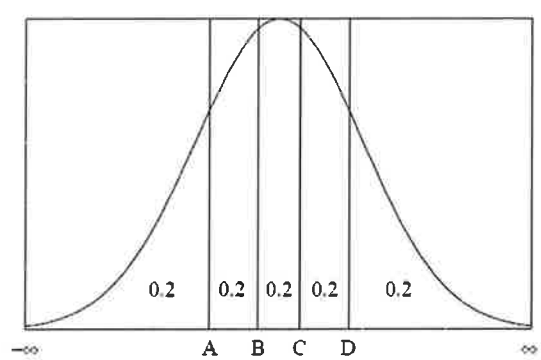

Fig. 1. Probability distribution function for a normally distributed variable x.

The points are found by randomly sampling among substrata regions and using the inverse cumulative function of the data (lower quantile tail of the sample normal distribution) to back calculate the converted sample point. A quantile is a division of any distribution into equal ordered subgroups; terciles are thirds, quartiles are quarters, quintiles are fifths, deciles are tenths, and centiles are hundredths. Sampling problems in optimization typically fit in one of two categories, normal and uniform distribution profiles. In a normal distribution, points cluster about the mean. The probability of finding a point in each of the five segments shown in Fig. 1 is 20%, so that

$$P(-\infty < x \le A) = P(A < x \le B) = P(B < x \le C) = P(C < x \le D) = P(D < x \le \infty) = 0.2. \quad (1)$$

The cumulative distribution value at point x_i is the probability of finding a point at $x<x_i$.

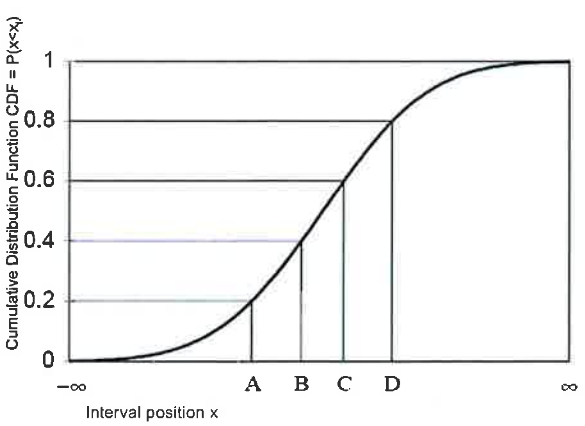

Fig. 2. Cumulative distribution function for a normally distributed variable x.

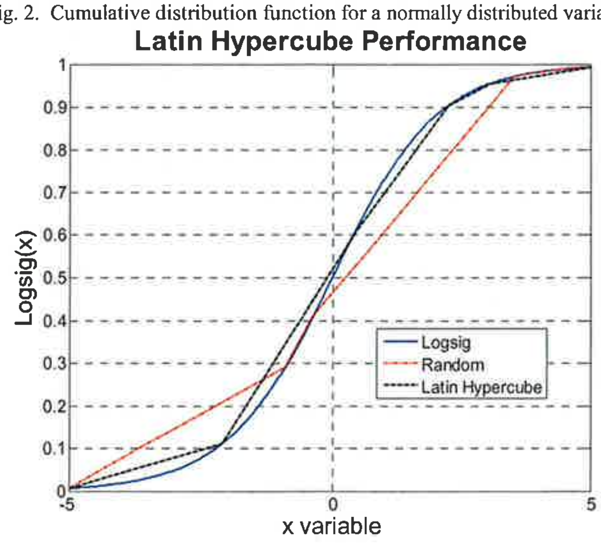

Fig. 3. Reconstruction of a log sigmoid function from 7 sample points.

$$\sqrt{\int \left(f_{MonteCarlo} - \log sig\right)^2 dx} - \sqrt{\int \left(f_{sample} - \log sig\right)^2 dx} .$$

A Latin hypercube distribution function with n samples is found by randomly assigning points to one of n intervals (illustrated as one of the five intervals on the vertical ordinate of Fig. 2) and then solving the inverse probability function for the value on the horizontal abscissa. The error function is defined as

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt . \quad (2)$$

For a normally distributed random distribution with a mean $\overline{x}$ and standard deviation σ, this value comes through the inverse error function,

$$x = \overline{x} + \sqrt{2}\,\sigma\, erf^{-1}\left(2 \cdot cdf - 1\right) . \quad (3)$$

A one variable example helps illustrate how this works. Consider reconstructing a log sigmoid function from seven sample points including the ends. Latin hypercube sampling delivers a significantly better sampling fit nearly all the time, as shown in Fig. 3. One indicator of this improvement can be gauged by integrating the squared difference of the sample prediction and the actual function.

A. Two Variable Sampling

There is an interesting two variable function in MATLAB® called "peaks," obtained by translating and scaling Gaussian distribution functions shown in the upper inset of Fig. 4. The middle inset is obtained from a two dimensional Latin hypercube sampling of the peaks function. Each of the variables x and y are built using a uniform probability function. The lower inset is obtained from a Monte Carlo sampling of the function. The power of the Latin hypercube sampling is easily detected by a visual comparison of these two sampled functions with the original function.

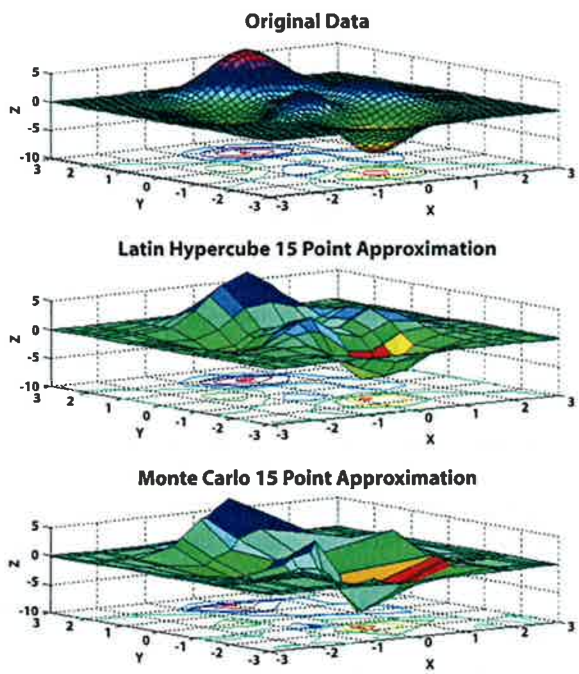

Fig. 4. Representation of a two dimensional function with Latin hypercube and Monte Carlo representations.

Latin hypercube can be used both with stochastic and deterministic algorithms, but is perhaps better suited to the latter. All deterministic methods are initial condition dependent. Restarting these algorithms with a Latin hypercube spread saves computation time and increases the confidence level of finding a global minimum. It can also be used with novel algorithms, such as pattern searches, to speed computation as the algorithm progresses.

B. Magnetic Field Optimization

It is difficult to find a better magnetic field optimization problem than international TEAM problem #22 [8][9][10]. Two concentric coils carry current with density 22.5 MA/m^2, as shown in Fig. 5. This superconducting magnetic energy storage device (SMES) must store 180 MJ, but not exceed a certain critical field within the conductor itself. With B in Tesla and J in A/mm^2, this relation is

$$|B| < (54 - |J|)/6.4. \quad (4)$$

With B_{norm} set to 3 mT and E_{ref} set to 180 MJ, the objective function ξ is defined as

$$\xi = 1/22\left(\sum_{i=1}^{22}\left|B_{stray_i}\right|^2\right)/B_{norm}^2 + \left(\left|Energy - E_{ref}\right|\right)/E_{ref}. \quad (5)$$

The objective is to determine the best parameters in Fig. 5 to minimize (5) subject to the constraint in (4). The problem has both a three variable and an eight variable version.

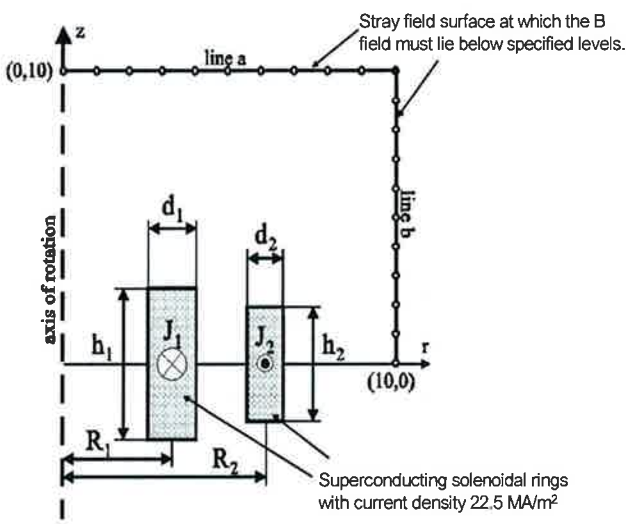

Fig. 5. Team problem 22 uses two solenoidal coils to store 180 MJ of energy while minimizing the field along stray field surfaces.

The author has written a MATLAB® code using elliptic integrals to represent the vector potential and B field of the coil represented as filaments [11], followed by Gauss Legendre integration over the domain of the coil.

Sequential quadratic programming (SQP) algorithms are among the more powerful deterministic methods that use gradient information [12][13]. The response surface method attempts to build an n dimensional surface of the problem and then to examine how that surface varies with the system unknowns [14]. The three variable version of TEAM problem 22 has about 14 local wells if the optimization is treated as a response surface. The results show that, except for very small run number groups, a Latin hypercube is virtually assured of always improving the probability of finding the minimum over a Monte Carlo approach, and especially as the run number increases.

C. Latin Hypercube and Pattern Search

Pattern search algorithms were first introduced by Hooke and Jeeves in 1961 [15]. This author was able to find only seven papers in IEEE Transactions on Magnetics where the technique was employed. In [16], a generic pattern search was combined with simulated annealing, attempting to avoid localized wells. In [17], a pattern search was used as a secondary step for a gradient-based algorithm. A standard form of the pattern search algorithm was used in magnet design, along with the response surface method in [18]. Most recently, Dilettoso [19] and Kim [20] used a genetic algorithm to identify the radius of local wells, what they calls "niches", and then used a pattern search to find the optimum within the niche. Whereas Dillettoso and Kim started with a stochastic genetic algorithm, Drago used the same two-step optimization approach, but began with simulated annealing [21]. Rao [22] did similar work is using a variant of simulated annealing followed by a pattern search to determine an optimal design. His annealing step was interesting in that he broke the problem into a family of convex optimization problems with different weighting functions.

A pattern search/Latin hypercube algorithm is slower than an SQP algorithm, but when the restart penalty is added to increase the confidence level, the computational advantage of the pattern search algorithm becomes evident (Fig. 6). The Latin hypercube/pattern search algorithm is a clear winner. This algorithm was also compared to a hybrid genetic algorithm – SQP algorithm; if both are non-vectorized, the genetic hybrid proves to be about 10 times slower than a pattern search with Latin hypercube approach.

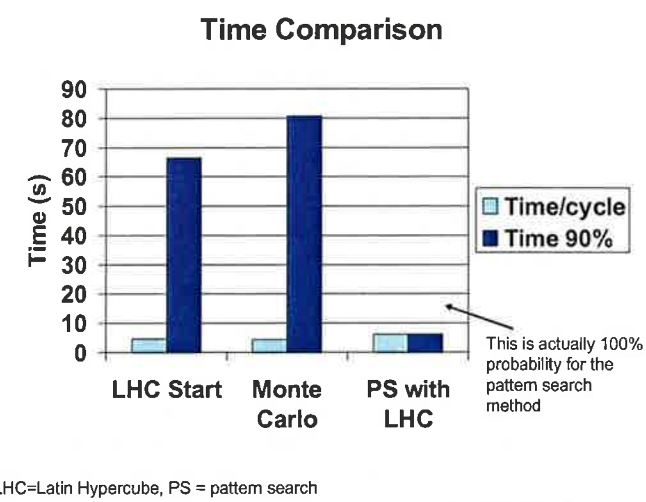

Fig. 6. Time required to get a 90% confidence level of finding the global minimum with three techniques.

III. Results

TEAM problem #22's eight-parameter optimization is challenging. To speed up processing time, an accelerated mesh convergence was employed in which the grid contraction is advanced as the algorithm approaches convergence. This eight-parameter optimization requires approximately six hours on a 3 GHz Pentium IV. TABLE I shows the restraints and best dimensions, as reported by the international team originator [8]. TABLE II shows the reported "best" web solution results and those derived using the discussed Latin hypercube pattern search algorithm. The calculation of the B field and the energy involves the integration of elliptic integrals, and so is likely to vary slightly depending on how those integrals are performed numerically. The third row of TABLE II gives the results of this author's calculation using the old result parameters in TABLE I, and shows that the two methods of evaluating the key indices are reasonably close. The new objective function is about one order of magnitude smaller than the present best solution, and the parameters are quite different. By chance, the algorithm reversed the radial placement of the positive and negative coils, a reversal which has no consequence.

TABLE I
PARAMETER BOUNDS AND RESULTING DIMENSIONS

	R_1 [m]	R_2 [m]	$h_1/2$ [m]	$h_2/2$ [m]	d_1 [m]	d_2 [m]	J_1 [MA/m^2]	J_2 [MA/m^2]
min	1	1.8	0.1	0.1	0.1	0.1	10	-30
max	4	5	1.8	1.8	0.8	0.8	30	-10
old	1.6	2.1	0.79	1.42	0.59	0.26	17.337	-12.57
new	2.8	2.34	1.55	0.78	0.12	0.39	23.381	-19.55

TABLE II
PUBLISHED "BEST" SOLUTION WEB RESULTS AND THOSE COMPUTED USING THE PATTERN RESEARCH

	ξ [-]	B_{stray}^2 [T^2]	Energy [MJ]
Old Results	5.52E-03	2.19E-10	179.9924
New Results	2.21E-04	1.89E-09	179.998

IV. Conclusions

Latin hypercube algorithms will increase the performance of all deterministic algorithms by increasing the probability of finding the global minimum over Monte Carlo counterparts. This is the most straightforward use of Latin hypercube sampling features. However, Latin hypercube searches also increase the performance of pattern search algorithms within the algorithm by speeding the polling direction search. A proper choice of mesh expansion and contraction can sometimes yield a 100% confidence level for finding the global minimum. The reader should not expect this to be the norm. Unless considerable time is spent selecting mesh alteration rates, the algorithm will be shown to yield many local minima. A superior solution is found for the rather challenging eight parameter workshop TEAM problem #22 using this method.

References

[1] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143-151, 1987.

[2] G. Wyss and K. Jorgensne, "A user's guide to LHS: Sandia's latin hypercube sampling software," SAND report 98-0210, Risk Assessment and Systems Modeling Department, Sandia National Laboratories, PO Box 5800, Albuquerque, NM, 87185-0747, February 1998.

[3] I. C. Kizilyalli, T. E. Ham, K. Singhal, J. W. Kearney, W. Lin, and M. J. Thoma, "Predictive worst case statistical modeling of 0.8-µm BICMOS bipolar transistors: a methodology based on process and mixed device/circuit level simulators," *IEEE Trans. Electron Devices*, vol. 40, no. 5, pp. 966-973, May 1993.

[4] K. Singhal and V. Visvanathan, "Statistical device models from worst case files and electrical test data," *IEEE Trans. Semi. Manufacturing*, vol. 12, no. 2, pp. 470-484, Nov. 1999.

[5] J. F. Swidzinski and K. Chang, "Nonlinear statistical modeling and yield estimation technique for use in Monte Carlo simulations [microwave devices and ICs]," *IEEE Trans. Microwave Theory and Techniques*, vol. 48, no. 12, pp. 2316-2324, Dec. 2000.

[6] R. G. Regis and C. A. Shoemaker, "Local function approximation in evolutionary algorithms for the optimization of costly functions," *IEEE Trans. Evolutionary Comp.*, vol. 8, no. 5, pp. 490-505, Oct. 2004.

[7] J. Cheng and M. Druzdzel, "Latin hypercube sampling in Bayesian networks," *Proc. 13th International Florida Artificial Intelligence Research Society Conference*, J. Etheredge and B. Manaris, Eds. 2000, pp. 287-292.

[8] http://www.igte.tugraz.at/archive/team_new/description.php.

[9] P. Alotto, A. Bertoni, G. Molinari, M. Nervi, B. Brandstaetter, Ch.Magele, K. R. Richter, C. Ragusa, and M. Repetto, "A combined approach for the stochastic optimization of multiminima problems using adaptive fuzzy sets and radial basis functions," *IEEE Trans. Magn.*, vol. 34, pp. 2837-2840, Sept. 1998.

[10] K. R. Davey, "Examination of various techniques for the acceleration of multivariable optimization techniques," *IEEE Transactions on Magn.*, vol. 39, no. 3, pp. 1293-1296, May 2003.

[11] W. R. Smythe, *Static and Dynamic Electricity*, 3rd ed. New York: Hemisphere, 1989, pp. 290-291.

[12] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Computer J.*, vol. 6, pp. 163-168, 1963.

[13] D. Goldfarb, "A family of variable metric updates derived by variational means," Mathematics of Computing, Vol. 24, pp. 23-26, 1970.

[14] G. E. P. Box and N. R. Draper, *Empirical Model Building And Response Surface*, New York: John Wiley and Sons, 1987.

[15] [R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. of ACM*, vol. 8, 1961.

[16] E. Costamagna, A. Fanni, and M. Marchesi, "Modified TEM cell design using mixed simulated annealing-deterministic optimization," *IEEE Trans. Magn.*, vol. 32, no. 3, part 1, pp. 1202-1205, May 1996.

[17] A. Aarkadan and S. Subramaniam-Silvanesan, "Identifying an inaccessible electrostatic source with gradient-based inverse problem methodology and boundary elements," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1578-1581, May 1999.

[18] P. Alotto, P. Girdinio, P. Molfino, and M Nervi, "Mesh adaption and optimization techniques in magnet design," *IEEE Trans. Magn.*, vol. 32, no. 4, pp. 2954-2957, July 1996.

[19] E. Dilettoso and N. Salerno, "A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1203-1206, April 2006.

[20] J.-K. Kim, D.-H. Cho, H.-K. Jung, and C.-G. Le, "Niching genetic algorithm adopting restricted competition selection combined with pattern search method," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 1001-1004, March 2002.

[21] G. Drago, A. Manella, M. Nervi, M. Repetto, and G. Secondo, "A combined strategy for optimization in nonlinear magnetic problems using simulated annealing and search techniques," *IEEE Trans. Magn.*, vol. 28, no. 2, pp. 1541-1544, March 1992.

[22] L. Rao, W. Yan and R. He, "Mean field annealing (MFA) and optimal design of electromagnetic devices," *IEEE Trans. Magn.*, vol. 32, no. 4, pp. 1218-1221, July 1996.