A Kriging Based Optimization Approach for Large Datasets Exploiting Points Aggregation Techniques

Yinjiang Li, Song Xiao, Mihai Rotaru, and Jan K. Sykulski, *Fellow, IEEE*

Electronics and Computer Science, University of Southampton, Southampton, UK, jks@soton.ac.uk

***Abstract*—A kriging based optimization approach is proposed for problems with large datasets and high dimensionality. Memory usage is maintained via model centering aided by minimizing the impact of information loss on accuracy of new point prediction using points aggregation techniques. The 8-parameter TEAM problem 22 is revisited in the context of computational efficiency and accuracy.**

***Index Terms*—kriging, surrogate optimization, clustering, large datasets.**

I. Introduction

Surrogate modelling techniques are helpful tools in design optimization, especially when the underlying problem is computationally expensive. This situation frequently arises in the design of electromagnetic devices where time consuming finite element simulations may be necessary to ensure accurate performance prediction. Kriging based methodologies have been shown to be particularly useful and offer good accuracy while reducing the number of required objective function calls. Unfortunately, the complexity of the algorithm increases as applying kriging involves the inversion of a correlation matrix, resulting in O(n^3) computation cost and O(n^2) storage cost. Consequently, the otherwise efficient kriging is often limited to smaller scale problems. Attempts have been made to address this bottle-neck of kriging methods when applied to large datasets, including zooming-in modeling [1], moving-window kriging [2], covariance tapering [3] and fixed rank kriging [4].

In this paper points aggregation is proposed. The method locates the most interesting search area for the next infill point, aggregates points outside this area, and finally builds a kriging model for infill point search within the identified center area.

II. Center Positioning

The objective is to locate a center around which a prescribed number of 'interior' points will not be aggregated and where the next infill point will be added inside this region. For an arbitrary location inside the design space a corresponding $Criterion_c$ can be introduced based on (1) below, then the location with the maximum $Criterion_c$ will be defined as the model center

$$Criterion_c(x) = \begin{cases} C_1 + C_2, & random(0,1) < v \\ C_3, & otherwise \end{cases} \quad (1)$$

$$C_1 = {R_{C_1}}^{-1} \times \sum_{i=1}^{k} \|\boldsymbol{x_i} - \boldsymbol{c}\|^{\frac{1}{2}}, \quad R_{C_1} = k \times \max\left\{\|\boldsymbol{x_i} - \boldsymbol{c}\|^{\frac{1}{2}}\right\} \quad (2)$$

$$C_2 = {R_{C_2}}^{-1} \times \sqrt{\frac{\sum_{i=1}^{k}(y_i - \mu)^2}{k}}, \quad R_{C_2} = \frac{range(\boldsymbol{y})}{2} \quad (3)$$

$$C_3 = {R_{C_3}}^{-1} \times \left(max(\boldsymbol{Y}) - \frac{\sum_{i=1}^{k} -w_i y_i}{\sum_{i=1}^{k} w_i}\right), \quad (4)$$

$$R_{C_3} = range(\boldsymbol{Y}), \text{ and } w = e^{-v^{-5}(\boldsymbol{x_i} - \boldsymbol{c})} \quad (5)$$

where $\boldsymbol{c}$ denotes the center, $\boldsymbol{x}_i$ is the location of its i^{th} closest point, k defines the number of closest neighborhood points around $\boldsymbol{c}$, y_i denotes the objective function values of the i^{th} closest neighborhood point, w_i is the weight term which has an inverse relationship with the distance from point i to the center $\boldsymbol{c}$, μ is the mean of $\boldsymbol{y}$, and v is the calculated probability. C_1 in (1) is the sum of square roots of Euclidean distances between the hypothetical center $\boldsymbol{c}$ and k nearest points around it. The value of C_1 is a measure of a sample rate within the region; it determines how close a hypothetical center $\boldsymbol{c}$ is in relation to its nearest k points, while the square root deemphasizes the influence of remote points. C_2 is a weighted standard deviation of the objective function values of all the neighborhood points. Finally, C_3 is the weighted mean of the objective function values of all the neighborhood points. Each point is weighted by an exponential function, whose gradient is controlled by the parameter v. The smaller values of v apply less weight on remote points. The C_1 and C_2 terms will encourage exploration of the under-sampled and rough areas, respectively, while C_3 focuses on exploitation of the current optimum region.

The probability of exploration and exploitation is controlled by a parameter v, whose value is related to the root-mean-square deviation (RMSD) of the kriging model. Instead of a deterministic mixture of exploration and exploitation terms, a stochastic approach has been applied to eliminate the risk of the criterion function being trapped in a local optimum.

The predictor deviation d in iteration $iter$ is defined as

$$d_{iter} = f(\boldsymbol{x}_{iter}) - Pred_{iter-1}(\boldsymbol{x}_{iter}) \quad (4)$$

where $\boldsymbol{x}_{iter}$ is the location of the infill point in the $iter^{th}$ iteration, $f(\boldsymbol{x})$ is the evaluated objective function at location $\boldsymbol{x}$ and $Pred_{iter-1}(\boldsymbol{x})$ is the predicted objective function value at location $\boldsymbol{x}$ in iteration $iter - 1$.

Manuscript received November ?, 2016; revised ?? ??, 201? and ?? ?, 201?; accepted ?? ?, 201?. Date of publication ?? ?, 201?; date of current version ?? ?, 201?. Corresponding author: J. K. Sykulski (e-mail: jks@soton.ac.uk).

Digital Object Identifier (inserted by IEEE).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMAG.2017.2665703, IEEE Transactions on Magnetics

The deviation d_{iter} is calculated and recorded whenever a new infill point is defined. Finally, the historical root-mean-square deviation (RMSD) is

$$RMSD = \sqrt{\frac{\sum_{iter=1}^{m} {d_{iter}}^2}{m}} \quad (5)$$

where m is the most recent iteration.

To obtain a generalized weight term, an exponentially weighted RMSD is applied in this case in order to put more weight on recent results; the aim is to emphasize the recent prediction error to reflect on the optimization progress. The exponentially weighted RMSD is calculated using the formula

$$RMSD_{weighted} = \sqrt{\frac{\sum_{iter=1}^{m}(1-\alpha)^{m-iter} \times {d_{iter}}^2}{\sum_{iter=1}^{m}(1-\alpha)^{m-iter}}} \quad (6)$$

where α is the decay parameter and $0 < \alpha < 1$. A larger α will put less weight on past prediction errors and vice versa. When $\alpha = 0$, $RMSD_{weighted}$ is identical to $RMSD$. The parameter α is defined by considering the smoothness of the underlying function and the 'allowance' for the number of function calls. Generally, a smaller value provides a smoother decay of the parameter v, thus the solver has a better chance to locate the global optimum. Based on our experiments, the algorithm preforms best when α is set within the range of 0.01 to 0.1.

A generalized weight term that represents the current optimization progress in terms of model prediction deviations may be defined by taking a ratio of the exponentially weighted RMSD and regular RMSD of historical prediction errors

$$v = weighted\_RMSD / RMSD \quad (7)$$

The parameter v can be regarded as a measure of model quality at any stage, v often ranges between α and $1+\alpha$, and α controls the gradient of the exponential weight function; as model deviation decreases v will gradually move towards zero.

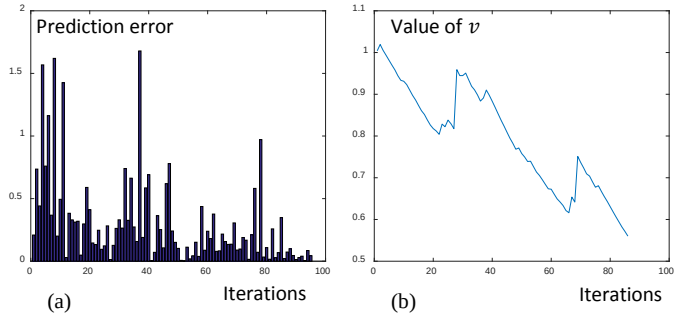


Fig. 1. (a) The prediction error, (b) the value of v, as iterations progress.

Regardless of which criterion function is used, this stage involves finding the number of closest neighborhood points k around $\boldsymbol{c}$ and calculating the parameter v; this requires extra computation resources, thus finding the hypothetical center $\boldsymbol{c}$ using exhaustive search is not practical. It may be seen as a global optimization problem with the input $\boldsymbol{c}$, the output $Criterion_c$, and the objective function $Criterion_c = (C_1 + C_2)$ or $Criterion_c = C_3$. Because the center only defines an area for the new infill point, an approximate solution will suffice; here a stochastic sequential global optimization method of simulated annealing has been utilized. It is argued that as the precision of estimating this intermediate optimum is not that important (some inaccuracy may be tolerated), a sequential method has the advantage of higher efficiency and very short computing times over population based methods.

The response surface of a 2D function is plotted in Fig. 2, with the red crosses at the bottom marking the location of existing design points. Figs. 3 (a) and (b) illustrate the criterion function for exploration and exploitation terms at the 90th iteration, respectively. As can be seen from the figures, the C_1+C_2 term encourages search in less sampled and non-smooth areas around x=[0.92,0.79], while the C_3 function suggests exploration of the area around the minimum at x=[0.20 0.27].

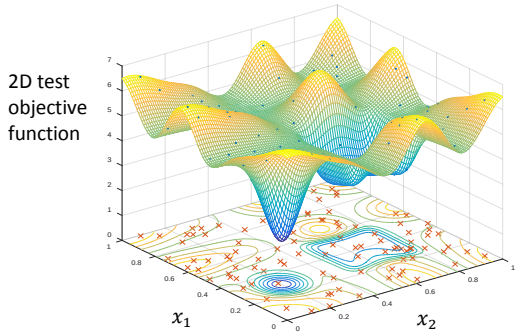
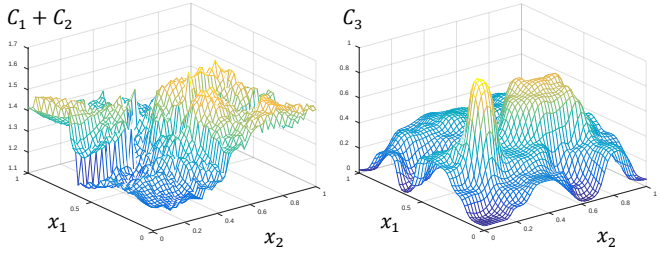


Fig. 2. A 2D test function and existing design points (with normalized axes).



(a) (b)

Fig. 3. (a) Exploration functions $C_1 + C_2$, (b) exploitation function C_3.

III. Outside Points Aggregation

The objective of this step is to aggregate design points further from the model center into a smaller number of nodes (the 'knots'), so that the total number of nodes and points in the model can be fitted into the memory. The problem of outside points aggregation involves hierarchical cluster analysis (HCA) [5] and a single variable optimization design. The objective of HCA is to group outsider points into a set of clusters so that the number of clusters is equal to the number of nodes. The value of the node is equal to the weighted mean of aggregated points.

There is rich literature related to cluster analysis, in particular in the field of data science, and many algorithms have been published. The points aggregation can be treated as a k-mean clustering problem where there are significantly more clusters to be identified compared to conventional clustering problems. In this paper we developed a sequential algorithm for weighted points clustering, the pseudo code of which is given below

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMAG.2017.2665703, IEEE Transactions on Magnetics


```
for m = 1: sample size
    for n = 1: number of clusters
        · calculate new cluster centroid
        · calculate weighted Euclidean distance
          to centroid
    end
    · find the cluster(x)with minimum weighted
      Euclidean distance e
    if e < weighted dissimilarity
        · add point(m) to cluster(x)
    else
        · create a new cluster
    end
end
```

The cluster's centroid $\boldsymbol{o}$ of a set of m points $\boldsymbol{x}$ is given by

$$\boldsymbol{o}(\boldsymbol{x}) = \frac{1}{p} \times \sum_{i=1}^{p} \boldsymbol{x}_i \tag{8}$$

where p is the number of points to be considered.

The Euclidean distance is weighted by the distance between the cluster's centroid $\boldsymbol{o}$ and the model's center $\boldsymbol{c}$, based on a correlation function, while the dissimilarity is weighted too, as a larger distance results in lower correlation and therefore information loss due to points aggregation will have a smaller impact on the prediction result within the center area. The design space is normalized and each cluster's centroid is weighted by the Gaussian function.

The Gaussian correlation function in the kriging model is used to calculate the weight w; the original function is given by

$$f(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\theta \boldsymbol{d}_{ij}{}^2} \tag{9}$$

Because the hyperparameter θ needs to be tuned during the model construction, but is unknown when outside points are aggregated, we recommend $\theta = 2$ (based on experience) as this provides a smoother decay in correlation and gives generally good results when the underlying problem is unknown.

The optimization problem is defined as $OF(d) = (n-q)^2$, where d is the input variable dissimilarity, n is the number of nodes/clusters generated during the clustering process, q is the number of outside nodes that can be fitted into the memory. The pseudo code provided above shows a basic workflow of the clustering process; to speed up the process, clusters with the minimum value of $n-c$ are kept in memory and a new clustering iteration starts with these existing clusters. The clustering process is terminated when the sum of the number of existing clusters and the number of unclassified points is less than the number of nodes calculated previously.

The following example illustrates outside points aggregation applied to a 2D problem. Fig. 4(a) shows the clustering without Gaussian weights, while Fig. 4(b) illustrates the clustering with Gaussian weight terms applied. The problem consists of 500 observations, assuming that the memory can build a kriging model up to 250 design points. We specify that 40% of the memory will be used to store the interior points within the model's central area, while the remaining 60% of memory is used to store nodes related to outside points. The 400 points outside the center area are aggregated into 150 nodes.

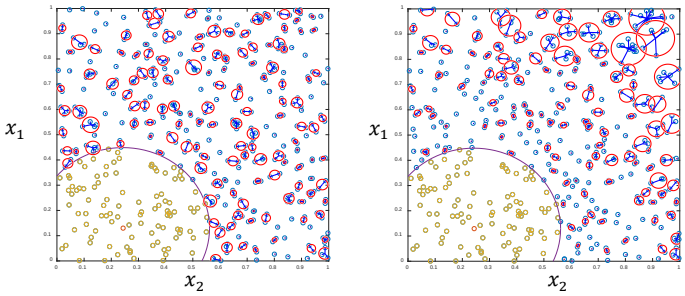


Fig. 4. (a) Clustering without Gaussian weight functions, (b) Clustering with Gaussian weight functions (with normalized axes).

IV. A 2D Example

The point aggregation technique is illustrated in Figs. 5 and 6 by a 2D example. The kriging model in Fig. 5 is built using 100 design points, while the model in Fig. 6 contains 60 nodes, including 20 points inside the center area and 40 nodes outside, where 62 original points have been aggregated. A slight loss of information has resulted – mainly away from the important region – but, as the size of the correlation matrix depends on the square of the number of points, the memory requirement has been reduced by 64%. The test function is given as follows

$$f(x_j) = c + a \prod_{j=1}^{j=2} cos\left(w(x_j - p_j)\right) e^{-\left(w(x_j - p_j)\right)^m} + b \prod_{j=1}^{j=2} sin\left(e^{-\left(v(x_j - q_j)\right)^k}\right) + f_{wave} \tag{10}$$

where $c = 4.5$, $a = 3.5$, $w = 8.4$, $p_1 = 0.2$, $p_2 = 0.3$, $m = 2$, $b = 2.8$, $v = 6.4$, $q_1 = 6.5$, $q_2 = 8$, $k = 6$, and f_{wave} is an interpolation function of a set of randomly generated points.

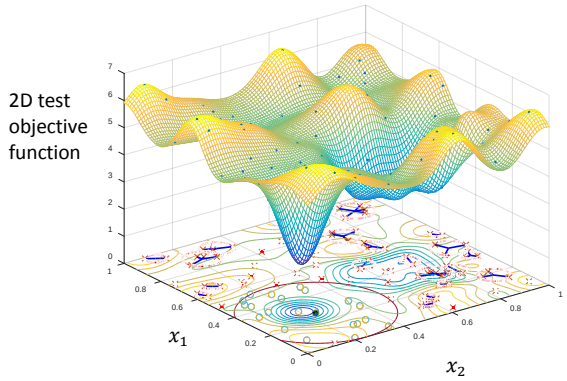


Fig. 5. Kriging estimate of a 2D test function (100 design points).

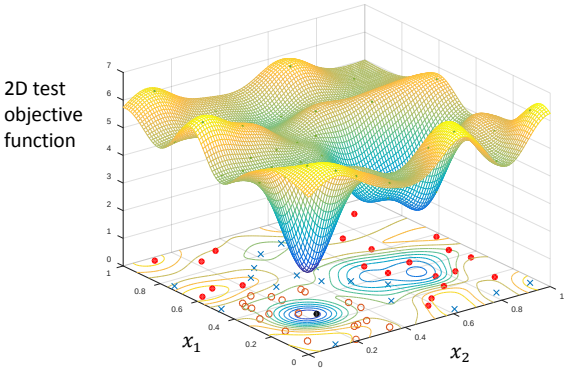


Fig. 6. Kriging estimate after point aggregation (60 nodes).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMAG.2017.2665703, IEEE Transactions on Magnetics

TABLE I
PERFORMANCE COMPARISON BETWEEN DIFFERENT ALGORITHMS

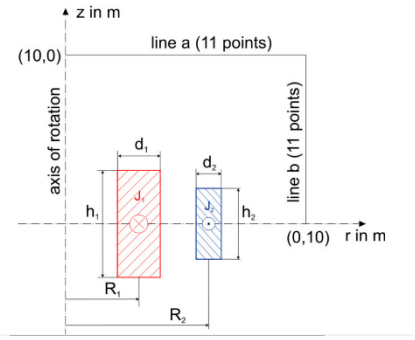
Algorithm	R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm^2)	J_2 (A/mm^2)	Objective function	Constraints penalty [2]	No. of FEM calls
PSO	1	2.2647	1.1076	1.7766	0.5225	0.3442	28.1779	–5.4921	1.5673	85.0413	~6000
Q-PSO	2.2947	2.6126	0.39	2.2704	0.3967	0.204	30	–21.293	2.4016	13.3456	~6000
E-QPSO	1	1.8	0.38	3.6	0.5155	0.2851	19.9975	–6.3571	0.3464	0.3685	~6000
GSA	1.939	2.823	0.37	1.101	0.399	0.195	22.5	–22.5	1.5547	0.195	17150
ES	1.99	2.931	0.421	0.94	0.29	0.188	26.6	–26.6	0.4103	1.69235	4200
SAA	1.694	2.907	0.394	0.882	0.323	0.207	20.9	–20.9	1.0087	1.09395	14000
Kriging standard	1	1.8	1.56	1.39	0.4	0.15	30	–30	1.4065	32.9056	449
Kriging proposed [1]	3.272	3.573	1.819	1.106	0.195	0.154	26.932	–23.259	0.0383	0.0159	500 [3]
	1.103	2.318	3.193	0.288	0.259	0.734	22.5	–22.5	0.0014	0.0003	829 [4]
Original answer	1.296	1.8	2.178	3.026	0.583	0.195	16.955	– 18.91	0.0033 [5]	0	–

PSO: particle swarm optimization [6], Q-PSO: quantum-behaved particle swarm optimization [6-8], E-QPSO: QPSO with exponential probability distribution [9], GSA: global search algorithm [10], ES: evolution strategy [10], SAA: simulated annealing algorithm [10]. Results for PSO, Q-PSO, E-PSO, GSA, ES and SAA taken from [6] to [10]. The comparison is for the 8 parameter continuous case [11]. Notes:

1) The new kriging algorithm offers significant savings in memory related to the correlation matrices; this has been achieved by aggregating the outside points.
2) Solutions from some previously published methods have violated the quench condition; the degree by which this constraint has not been met is given by the 'penalty' (high values indicate severe violation). In some cases the geometrical or current density constraints have not been met either.
3) For a fairer comparison of memory usage between standard kriging and the proposed kriging method, the maximum number of iterations was set to 500, while maintaining a maximum of 375 nodes; a memory saving on correlation function of ~50% was achieved and – as a bonus – a better optimum was found.
4) The proposed enhanced kriging method may be allowed to continue the search with the number of nodes maintained at 500; improved results have been achieved (better value of objective function and lower constraint violation) after more iterations, at the modest expense of more FEM calls.
5) The value of the objective function in the original specification was a little different; it was recalculated here using a consistent FEM model for comparison.

V. PRACTICAL EXAMPLE TEAM 22

The superconducting magnetic energy storage device in TEAM problem 22 consists of two superconducting coils. The design objective is to minimize the stray magnetic field while maintaining the stored energy at 180 MJ (see Fig. 7), subject to specified quench conditions and geometrical constraints [11].



$$OF = \frac{B_{stray}^2}{B_{norm}^2} + \frac{|E - E_{ref}|}{E_{ref}}$$

$$E_{ref} = 180\ MJ,$$
$$B_{norm} = 200\ \mu T$$

$$B_{stray}^2 = \frac{\sum_{i=1}^{n=22} |B_{stray,i}|}{22}$$

Fig. 7. The superconducting magnetic energy storage device (TEAM 22) [11].

The results are summarized in Table I. As reported before, kriging has shown its superiority by dramatically reducing the number of function calls and thus avoiding excessive use of the computationally expensive finite element software. Moreover, points aggregation has capped the number of active points in the design space with the benefit of reducing the memory requirements without sacrificing the accuracy. Finally, the iterations may be allowed to continue to achieve an even better design. The computational overhead associated with the proposed algorithm is very modest and considerable savings in overall simulation times may therefore be achieved.

VI. CONCLUSIONS

A kriging based optimization approach for large datasets has been proposed and its efficiency demonstrated using the TEAM 22 problem. The model center positioning algorithm balances exploration and exploitation assisted by the use of a stochastic approach, which eliminates the risk of a deterministic criterion function being trapped in a local optimum. It has been found that the size of the correlation matrices can be greatly reduced by applying points aggregation techniques. It is shown that the proposed approach can fit a large set of data into a limited size of memory and whereas some loss of information about remote points may be experienced this is alleviated by the use of points aggregation incorporating a new weighted clustering algorithm.

REFERENCES

[1] S. Xiao, M. Rotaru, and J. K. Sykulski, "Adaptive weighted expected improvement with rewards approach in kriging assisted electromagnetic design," *IEEE Trans. Magn.*, vol. 49, no. 5, pp. 2057-2060, 2013.
[2] T. C. Haas, "Lognormal and moving window methods of estimating acid deposition," *Journal of the American Statistical Association*, vol. 85, no. 412, pp. 950-963, 1990.
[3] R. Furrer, M. G. Genton, and D. Nychka, "Covariance tapering for interpolation of large spatial datasets," *Journal of Computational and Graphical Statistics*, 15:3, pp. 502-523, 2006.
[4] N. Cressie and G. Johannesson, "Fixed rank kriging for very large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, pp. 209-226, 2008.
[5] F. Murtagh, "A survey of recent advances in hierarchical-clustering algorithms," *The Computer Journal*, vol. 26, pp. 354 -359, 1983.
[6] L. S. Coelho and P. Alotto, "Global optimization of electromagnetic devices using an exponential quantum-behaved particle swarm optimizer," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1074-1077, 2008.
[7] T. Hogg and D. S. Portnov, "Quantum optimization," *Inform. Sci.*, vol. 128, (3–4), pp. 181-197, 2000.
[8] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," *Proc. Congress Evolution of Computation*, vol. 1, pp. 325-331, 2004.
[9] R. A. Krohling and L. S. Coelho, "PSO-E: particle swarm with exponential distribution," *Proc. IEEE Congresss Evolution Computation*, Vancouver, BC, Canada, pp. 5577-5582, 2006.
[10] P. Alotto, A. V. Kuntsevitch, C. Magele, G. Molinari, C. Paul, K. Preis, M. Repetto, and K. R. Richter, "Multiobjective optimization in magnetostatics: a proposal for benchmark problems," *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 1238-1241, 1996.
[11] P. Alotto, U. Baumgartner, F. Freschi, M. Jaindl, A. Kostinger, C. Magele, W. Renhart, and M. Repetto, "SMES optimization benchmark: TEAM workshop problem 22," http://www.compumag.org/jsite/team.html