MRI Magnet Design: Search Space Analysis, EDAs and a Real-World Problem with Significant Dependencies

Bo Yuan
School of Information Technology and Electrical Engineering
The University of Queensland
QLD 4072, Australia
+61-7-33651636
boyuan@itee.uq.edu.au

Marcus Gallagher
School of Information Technology and Electrical Engineering
The University of Queensland
QLD 4072, Australia
+61-7-33656197
marcusg@itee.uq.edu.au

Stuart Crozier
School of Information Technology and Electrical Engineering
The University of Queensland
QLD 4072, Australia
+61-7-33657098
stuart@itee.uq.edu.au

ABSTRACT
This paper introduces the design of superconductive magnet configurations in Magnetic Resonance Imaging (MRI) systems as a challenging real-world problem for Evolutionary Algorithms (EAs). Analysis of the problem structure is conducted using a general statistical method, which could be easily applied to other problems. The results suggest that the problem is highly multimodal and likely to present a significant challenge for many algorithms. Through a series of preliminary experiments, a continuous Estimation of Distribution Algorithm (EDA) is shown to be able to generate promising designs with a small computational effort. The importance of utilizing problem-specific knowledge and the ability of an algorithm to capture dependencies in solving complex real-world problems is also highlighted.

Categories and Subject Descriptors
J.2 [**Physical Science and Engineering**]: *Electronics*.

General Terms
Algorithms, Performance, Design, Experimentation

Keywords
EDAs, Real-World Problem, MRI

1. INTRODUCTION
In the field of Evolutionary Computation, a large number of Evolutionary Algorithms (EAs) and variations continue to appear in the literature. Typically, a new algorithm is tested on some well-known artificial benchmark problems and conclusions about its performance are then made based on corresponding experimental results. Although the ultimate goal of EAs is to solve various problems in the real world, many of them have never been tested on such problems, especially large-scale applications. An algorithm's performance on benchmark problems, while useful in validating the principles of the algorithm, provides little confidence on its performance in significantly more complex situations. As a consequence, although people in EA community do believe in the performance and potential of newly introduced EAs in solving challenging real-world problems, these algorithms have not yet played an important role in practice where optimization problems are still largely being attempted by various traditional methods.

On the other hand, although there have been more and more applications of EAs in real-world problems, many of them are focused on solving a very specific engineering problem and it can be difficult for other researchers to extract and transfer knowledge gained in solving one problem to assist in solving other problems and/or to gain insight into the performance of algorithms. For example, identifying and exploiting structural features of a problem has the potential to provide useful information for choosing an appropriate optimization algorithm and to assist in understanding the performance of an algorithm. Unfortunately, detailed analysis of problem structure is itself a challenging problem that has received relatively little attention. Hence, there are few guidelines to assist in selecting one of the many algorithms available to a given real-world problem. Furthermore, experimental factors that could influence the performance of EAs such as parameter settings, population initialization, constraint handling might not be completely specified and usually final or best results are the focus rather than reporting the complete set of experimental results obtained. All of these issues contribute to making the transfer of new EAs to real-world applications a challenging task.

In this paper, a case study on the magnet configuration design task in Magnetic Resonance Imaging (MRI) systems is conducted, which has recently been tackled by various metaheuristic algorithms [1, 4, 10]. The purpose is two-fold. Firstly, we want to evaluate a relatively new class of EAs - Estimation of Distribution Algorithms (EDAs)[8] on a real-world problem, which has not been done before. Secondly, we aim to investigate certain structural features of this problem, using techniques that can be applied to other problems, and to relate the performance of the algorithm to the problem structure.

The rest part of the paper is organized as follows. The next section gives a detailed description of the problem specification while Section 3 discusses a statistical method for estimating the number of optima in the search space and applies it to the MRI design problem. Experimental results using EDAs on the problem are presented in Section 4 and our work is concluded in Section 5 with directions for future work.

GECCO'05, June 25-29, 2005, Washington DC, USA.

2. OVERVIEW OF MRI MAGNET DESIGN

Magnetic Resonance Imaging (MRI) is an imaging technique widely used to produce high quality images of the inside of the human body. The general shape of a MRI system is roughly cylindrical, containing a deep bore in the centre where the patient is to be placed. The magnet system is used to produce an intense and homogeneous field in the region to be imaged to obtain images of good quality. A major challenge is to shorten the MRI magnet design (conventional systems are usually around 1.8 to 2 m in length) so that the perception of claustrophobia experienced by patients can be reduced. This is a significant engineering challenge as the field homogeneity is strongly dependent on the overall length of the coil structure.

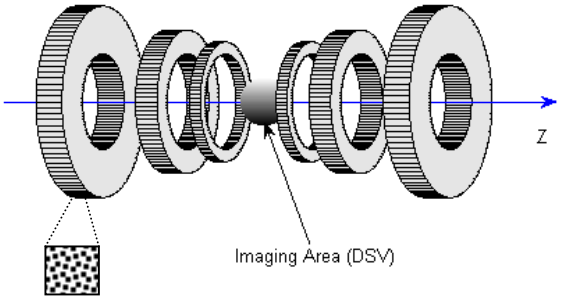

Figure 1. A six-coil symmetric magnet structure in a conventional MRI system.

As shown in Figure 1, a typical magnet design contains a number of concentric coils with regard to Z-axis, which are symmetric about the X-Y plane (i.e., vertical to Z-axis, not shown in the figure). Each coil has a ring shape with rectangular cross-section and contains many turns of superconductive wires. The imaging area (dsv) is specified by a spherical area in the origin with a typical diameter of 45 or 50 cm. The major optimization objective is to create a very homogeneous field within dsv by finding the best shapes and positions of these coils. For each coil, there are four parameters:

- axial position (distance from the origin to the middle of the coil along Z-axis)
- axial width
- number of turns (negative value represents a negatively wound coil)
- inner radius of the ring

Since coils in a typical design are arranged in symmetric pairs, for a design with N coils, the search space has 2N dimensions. Note that the value of N needs to be predefined unless the optimization algorithm in use can handle variable length candidates. If N is too small, it may be impossible to find the required solution given certain constrains (the minimum number of coils is usually not known *a priori*). On the other hand, too many coils will make the search more difficult and the design over-complicated.

The objective is to minimize the inhomogeneity of the field as measured in parts per million (ppm) with 0 ppm being the global optimum in theory (i.e., often not possible in engineering). A fast algorithm based on coil cross-section is adopted in this paper to calculate the homogeneity, allowing large numbers of candidate solutions to be evaluated within a reasonable amount of time[7]. Apart from this homogeneity requirement, in practice, it is also desirable to minimize the field outside the MRI system for the safety of other people. This secondary objective will introduce a new fitness function and could be combined with the main fitness function. This objective is not included in our current experiments but will be pursued in the future.

Finally, there are also various physical constraints embedded in the design. For example, the overall space is often restricted to a box area. Within this area, each coil cannot overlap with or be too close to others. Otherwise the design would be physically impossible to implement. Also, it is preferable not to have very thin vertical coils because they may bend due to the strong magnetic force.

3. ANALYSIS OF PROBLEM STRUCTURE

3.1 Estimation of Search Space Multimodality

One key step before applying any optimization algorithms is to have some idea of the structure of the problem to be solved. The reason is that, among the huge number of existing optimization algorithms, different algorithms often employ different heuristics and are likely to have their own strengths and weaknesses. So, having a deep understanding of the problem structure may be very helpful for choosing the appropriate algorithms. For example, if a problem is known to have a large number local optima, simple hill climbing algorithms or other gradient-based algorithms may be considered unsuitable due to their local searching behavior. On the other hand, if the problem structure is not complex, these algorithms may be expected to be more efficient and may yield more accurate results compared to population-based stochastic algorithms like EAs.

Little emphasis has been placed on investigating problem structure in previous work and algorithms have often been chosen without any clear justification. This may be partially due to the difficulty in characterizing the structure of large-scale, real-world problems. In such problems, the search spaces are often of high dimensionality. As a result, it is impractical if not impossible to perform extensive investigation. In other words, the effort required to explore the problem structure may be even greater than that needed to solve the problem itself.

Fortunately, with the help of statistical methods, it is possible to estimate some problem properties at an affordable computational cost. In this paper, we adapt the approach used in [3] to estimate the number of optima in the search space, which is one property of interest.

Suppose K is the total number of all possible unique outputs from a black-box system, each of which is equally likely to be generated at each time step. The question is how to efficiently estimate the value of K based only on observations of the output? The probability of consecutively observing N-1 unique outputs in the first N-1 steps and one duplicate output in the N^{th} step is specified by:

$$P(N,K) = \prod_{i=1}^{N-1} \frac{K-i+1}{K} \cdot \frac{N-1}{K} \qquad N \in [2, K+1] \qquad \text{Eq. 1}$$

Given a fixed K, Eq.1 could be regarded as the probability density function of N. Figure 2 (top) shows the distributions of N with different K values. It is clear that for different K values, the

distributions of N are different and the curves flatten out as K increases. The mean value of each density function can also be calculated according to:

$$\overline{N_K} = E(N \cdot P(N,K)) \qquad N \in [2, K+1] \qquad \text{Eq. 2}$$

Figure 2 (bottom) gives the mean values of N for some selected K values showing that as K increases from 300 to 100000, the mean value of N also increases but much more slowly (i.e., from around 20 to nearly 400). Note that those density functions are typically not symmetric, which means that the mean and the median are unlikely to be equal.

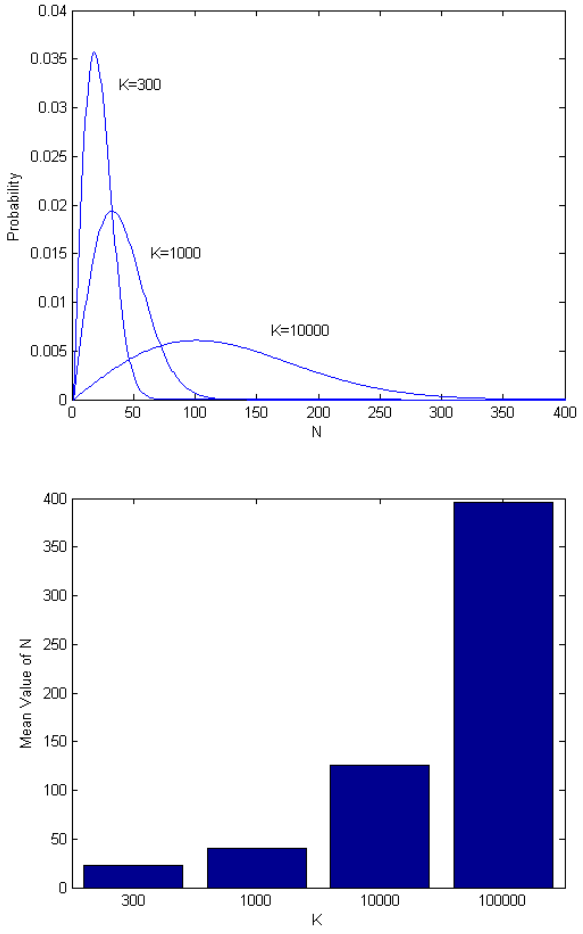

Figure 2. Probability distributions (top) and mean values of N (bottom) for different K values.

One approach to the estimation of K based on the above analysis is to conduct M independent sampling experiments. In each experiment, the outputs are monitored and the value of N (i.e., the number of outputs observed till the first duplicate) is returned as the experimental result. Given a set of such N values, a principled way to estimate a value for K is to use the maximum likelihood method by searching for the K whose corresponding probability density function is most likely to support the observed N values:

$$\max_K \sum_{i=1}^{M} \log P(N^i, K) \qquad \text{Eq. 3}$$

In practice, a number of experiments may be conducted in order to improve the accuracy of estimation. However, for large M values, it may be time-consuming to calculate Eq. 3, especially when there are a huge number of possible K values to be examined. A simpler method is to only utilize the sample mean of the N values and search different K values for the best mean value match. The advantage is that Eq. 2 is usually easier to calculate and can be pre-calculated as shown in Figure 2 (bottom).

The efficiency of this method is easily demonstrated. For example, assume that the unknown value of K=100,000, so that $N_K \approx 400$. If 10 of the experimental trials described above are conducted, leading to the observation of (ideally), around 4000 outputs, then this will provide a good estimation of the true value of K after observing only 4% of the total number of possible outputs. As K goes up, the efficiency could improve even further. If K is scaled up by a factor of 10 (i.e., K=1,000,000), N_K will only be scaled up by a factor of around 3 (i.e., $N_K \approx 1200$). In this case, the portion of all optima that need to be explored is down to around 1.2%. One thing that needs to be noted is that as K increases, the variance of the probability density function also increases. According to the Central Limit Theorem, the number of trials also needs to be increased in order to reduce the randomness of N_K.

Similarly, each optimization problem could be regarded as a black-box and each optimum corresponds to a unique output. By randomly sampling optima and counting the number of samples required to find the first duplicate, it is also possible to use the approach discussed here to perform estimation. An obvious question is how to sample optima in an optimization problem? Typically, this could be done by running a multi-restart hill climbing algorithm until it gets stuck with each time using a randomly chosen starting point. In an idealized situation, this hill-climbing algorithm should be able to find one optimum at the end of each trial. One issue is that since different optima are likely to have different basin sizes, the chance of finding each optimum may differ, which violates the assumption of equally probable outcomes. In this case, values of N produced will tend to be smaller and the true value of K may be underestimated.

To sample local optima using a hill climbing algorithm, the first choice that needs to be made is the definition of neighborhood, which is essential in determining whether a point is an optimum. There is another issue about the step size. In continuous spaces, with a large step size, an algorithm could often jump over some optima with small basins of attraction and thus the number of optima found may be less than the number of optima found by using a smaller step size. In other words, a large step size could effectively smooth the search space. Furthermore, in continuous spaces, the exact location of an optimum is related to the floating-point precision of the computer used. This creates another difficulty in determining whether two hill climbing trials getting stuck at different but close locations actually correspond to the same optimum, especially when there is no *a priori* knowledge about the problem structure.

Due to the above issues, the estimated K value is typically lower than its true value. Also, due to the inherent randomness in sampling, this estimation could not be expected to be very accurate. However, from a practical point of view, this method should still be useful as an indicator of the multimodality of the search space. Please refer to [5, 6] for related works on estimating the number of optima.

3.2 The Multimodality of the MRI Problem

In order to apply a hill-climber to our problem, the neighborhood could be defined in a number of different ways. One possibility is to allow all parameters to be changed simultaneously. In this situation, a hill climber needs to examine 3^n-1 neighbors in order to decide where to move next (i.e., n is the dimensionality). Obviously, this could quickly become intractable as n goes up. Alternatively, the hill climber could be restricted to moving along one dimension at a time, searching up to 2n neighbors. However, the issue is that if there is some dependence in the local space around an optimum, changing one parameter a time may make the algorithm get stuck when it is still distant from the optimum, which means that simply knowing a hill climber has stopped does not necessarily imply that it has found an optimum or is even very close to one (i.e., the first method could face the same issue if the step size is not appropriate). One solution is to use a very small step size but the algorithm may then need a significant number of steps to stop, which could be very time consuming. The approach used in our work is to start the hill climbing algorithm with a moderate step size and once it gets stuck, a very small step size is then used to conduct fine searching to make sure that the algorithm could finally be as close to the optimum as possible.

A 10-coil symmetric design was chosen as the target and only those 5 coils on the positive part of Z-axis were optimized, which created a 20D search space. The boundaries of the search space were set (for each coil) as:

- axial position: [20, 700]
- axial width: [20, 100]
- number of turns: [-5000, 5000]
- inner radius: [480, 520]

These values represent a typical feasible search space. For each hill climbing trial, the starting point and all points visited were restricted to the above area with the additional constraint that all coils must be within the positive part of Z-axis. If any variable exceeds its corresponding boundary, its value will be set to the boundary value. Furthermore, if a coil moves into the negative part of Z-axis, it will be moved back by increasing its axis position. The maximum number of steps was set to 10,000 (i.e., this is a fairly large number compared to the step size and the size of the search space) and each hill climbing trial started with step size 5 mm and then changed to 0.05 mm to do fine searching after it got stuck for the first time.

Five independent experiments were conducted with each one containing up to 1,000 hill climbing trials starting from random positions. At the end of each trial, the location and the fitness value of the best solution were recorded. The Manhattan distances between this solution and all existing solutions found in previous trials were calculated and if the distance between any pair of solutions was less than a threshold (100 mm), a duplicate was claimed to have been found and the experiment was terminated.

The difficulty of this problem is clearly evident from the experimental results. No duplicate could be found within the five experiments, which means that it is very likely that on average more than 1,000 samples are needed in order to find a duplicate. Using the method described in the last section, a conservative estimate of the number of optima in this problem is around one million. The distribution of fitness for the 1,000 best solutions (dsv=50 cm) from a certain experiment is shown in Figure 3 (top). This indicates that the hill climbing algorithm even had difficulty in finding solutions close to 1000 ppm let along 0 ppm. The distance distribution, taking into account the permutations of coils and current direction, is shown in Figure 3 (bottom) from which it is clear that all solutions were well separated from each other (i.e., the smallest distance was more than 650 mm). Based on these results, it is reasonable to say that this problem is extremely difficult for hill climbing and gradient-based algorithms unless they are lucky enough to start very close to the global optimum.

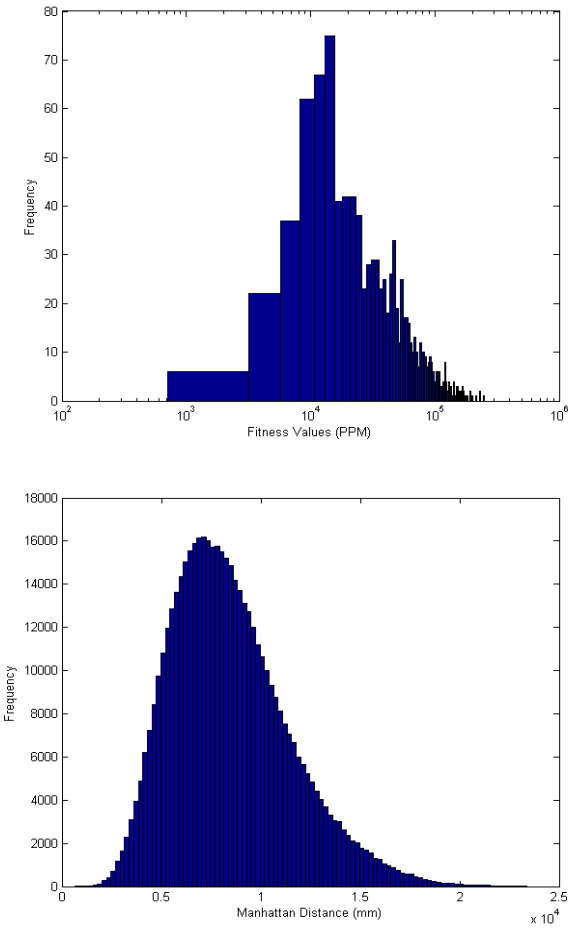

Figure 3. Distributions of fitness values (top) and Manhattan distance (bottom) of best solutions.

4. EXPERIMENTS

4.1 Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) [8] refer to a new class of population-based metaheuristic algorithms, which roughly follow the general framework of EAs. Instead of using traditional generator operators such as crossover and mutation, in EDAs, a probabilistic model of current promising individuals is maintained and all new individuals are generated by sampling from this model (Table 1). One of the major advantages of EDAs compared against other algorithms is that they can explicitly model the dependencies among problem parameters and utilize this information to conduct efficient searching. In optimization,

dependence means that it is impossible to optimize a problem by decomposing it into a set of one-dimensional problems and solving them independently. As a result, in order to solve problems with dependence structure efficiently, it is desirable to generate new individuals whose parameter values are chosen based on the problem structure.

Table 1. The Framework of EDAs.

```
Initialize and evaluate the population P
  While stopping criteria not met
    Select some individuals P^sel from P
    Estimate the density function θ^sel
    Create P' by sampling from θ^sel
    Evaluate individuals in P'
    Combine P and P' to form the new P
  End While
```

A continuous EDA named EDA$_{mvg}$ based on a single Gaussian distribution with full covariance structure was used in this paper. In this algorithm, the probabilistic model is fully specified by the mean vector and the covariance matrix where off-diagonal elements represent the dependence information, which can be calculated according to their maximum likelihood estimates. New individuals are generated from this multivariate Gaussian distribution using Cholesky decomposition [9]. The advantage of EDA$_{mvg}$ is that it can be implemented very efficiently and still has the capability of capturing complex structure. EDAs belonging to this type such as EMNA [8] and RECEDA [9] have shown comparable or even better performance on a number of test problems compared to more sophisticated EDAs.

One characteristic of many real-world problems is that they often have a large search space in terms of the range of each parameter and the number of parameters (dimensionality). As a consequence, a large population is often required to achieve satisfactory performance. Furthermore, the fitness functions may also involve some intensive computation. Fortunately, since EDAs are population-based algorithms, it is easy to apply parallel computation to significantly speedup the optimization process.

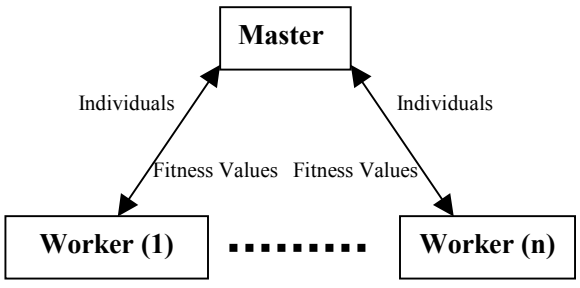

Figure 4. The framework of parallel computation with Master/Worker structure.

The approach used in our work is to, in each generation, separate the population into a number of subsets and assign each of them to a different process running on a different CPU/computer, which is similar to the structure in [2]. In Figure 4, the Master process contains the complete code of the EDA while each Worker process is dedicated to the evaluation of the fitness function. At each generation, the Master process sends individuals to each Worker process for evaluation, and Workers send the corresponding fitness values back to the Master process. The advantage of this framework is that it has a simple hierarchical structure and any modification of the optimization algorithm itself in the Master process has no influence on those Worker processes. Since the communication could conveniently be realized through shared files, there is no need to have a supercomputer with multiple CPUs, which is usually very expensive. Instead, a number of inexpensive PCs connected to a local network could be utilized with each one running a single process. In the mean time, since the volume of data that needs to be exchanged is often trivial compared to the speed of modern networks, the improvement of speed is very significant especially when the fitness function is very time-consuming and thus it is possible to employ a very large population and/or to run the algorithm for a large number of generations.

4.2 Experiments with EDA$_{mvg}$

Some preliminary trials were conducted to investigate the performance of EDA$_{mvg}$ on the magnet design task with 10 coils. The search space and constraint handling were the same as those in the hill climbing experiment above. For EDA$_{mvg}$, there are three parameters to be set: population size, number of generations and the selection operator. The population size was set to 1,000 with maximum number of generations equal to 100, allowing 100,000 fitness function evaluations in total. Truncation selection with selection ratio 0.3 was used to choose the promising individuals (i.e., top 30% individuals were chosen to build the model). The initial population was generated in the standard way, by choosing each parameter of each individual randomly within the feasible search space. Unfortunately, no satisfactory results were found with the above setting and most solutions found had homogeneity more than 2,000ppm (dsv=50 cm). Another issue is that there were often some large coils in those solutions, resulting in very strong magnetic field, which is not desirable (i.e., the field strength should be maintained at less than 10T). Certainly, by increasing the population size and/or the number of generations, one can expect to get better results. However, our major interest here is not to simply solve this problem but to solve it efficiently.

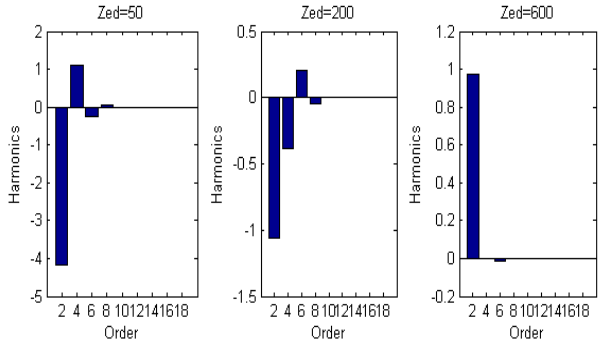

Figure 5. Even order harmonics of a positive coil at different axial positions: [50, 200, 600].

An investigation into the fitness function (and confirmed by our colleagues in MRI) reveals that the overall homogeneity is largely determined by the sum of the harmonics of each coil. Figure 5 shows the even order harmonics from 2 to 18 of a positive coil at

different axial positions (for symmetric designs, odd order harmonics are zero) from which some interesting patterns could be discovered immediately. For example, the second order harmonic is negative when the coil is close to the origin and gradually becomes positive when the coil is at the far end. In general, the relationship between the harmonics and the distance to the origin is nonlinear but quite smooth. Since the objective of optimization is to make the sum of harmonics as close to zero as possible with reasonable field strength, it is possible to come up with some heuristics about the layout of coils for good designs.

The heuristic adopted in our work is simple. A small positive coil is placed close to the origin while a large positive coil is placed at the far end so that the second order harmonics generated by the two coils tend to cancel against each other. The other three coils are put in between with each one restricted to a sub-interval, to try to avoid overlaps in the designs generated. In fact, this heuristic is used to identify the area where good designs are likely to exist, so that the actual space that needs to be searched can be significantly reduced.

The actual boundaries used are listed in Table 2 in which the axial length and inner radius were the same as before and the major change was the boundaries of axial position. Unlike in previous experiments where each coil was allowed to move freely along the Z-axis, all coils were restricted to some smaller ranges with possible overlapping (i.e., it is difficult to precisely determine the boundaries of each coil). Furthermore, the size of coil #1 was restricted to a small value and the size of coil #5 was intended to be large due to their locations with respect to the origin.

Table 2. Boundaries of five coils

#	Axial Position	Axial Length	Number of Turns	Inner Radius
1	[10, 50]	[20, 100]	[200, 500]	[480, 520]
2	[100, 200]	[20, 100]	[-1000, 1000]	[480, 520]
3	[200, 400]	[20, 100]	[-1000, 1000]	[480, 520]
4	[300, 600]	[20, 100]	[-1000, 1000]	[480, 520]
5	[600, 700]	[20, 100]	[2000, 5000]	[480, 520]

New experiments were then conducted using this heuristic. The effect was dramatic and EDA$_{mvg}$ could often find some good solutions. Figure 6 shows the cross-section of a design with 20.9 ppm (dsv=50cm) / 6.3 ppm (dsv=45cm) and the main field is around 6 T. The overall length of the magnet is around 1.41m, which is significantly less than conventional systems.

Table 3. Boundaries of four coils

#	Axial Position	Axial Length	Number of Turns	Inner Radius
1	[10, 50]	[20, 100]	[200, 500]	[480, 520]
2	[100, 400]	[20, 100]	[-1000, 1000]	[480, 520]
3	[300, 600]	[20, 100]	[-1000, 1000]	[480, 520]
4	[600, 700]	[20, 100]	[2000, 5000]	[480, 520]

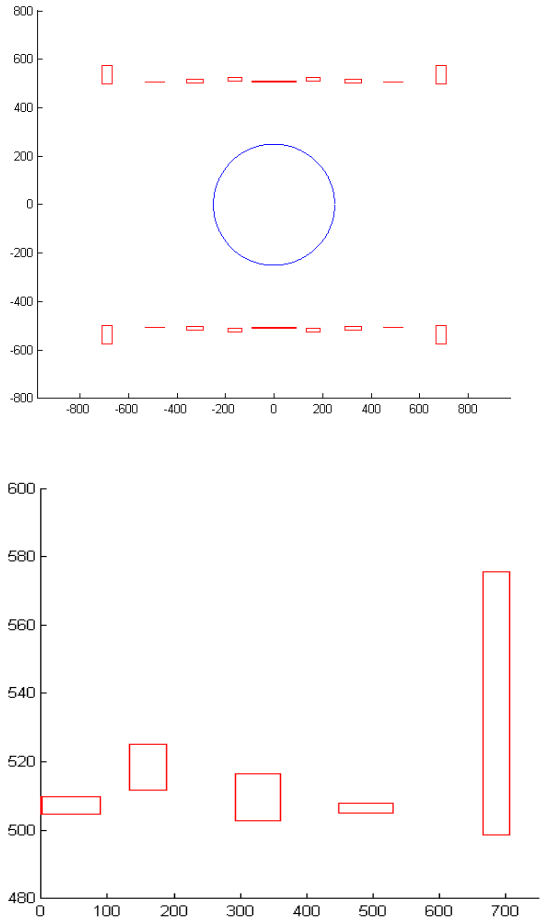

Figure 6. A 10-coil design (top) and a detailed view of the upper-right corner (bottom).

Since the minimum number of coils needed is typically not known, additional experiments were conducted with 8 coils to find out whether it is possible to simplify the design. The boundaries used for this 8-coil design are listed in Table 3. Compared to Table 2, the only change is that a single coil #2 replaced coils #2 & #3. Experimental results show that reducing the number of coils led to increasing performance of EDA$_{mvg}$. With four coils to be optimized, the search space was reduced as the dimensionality decreased from 20 to 16 and the convergence speed and accuracy of EDA$_{mvg}$ all improved. Figure 7 shows a design with 9.0ppm (dsv=50 cm)/2.0ppm (dsv=45 cm) and the main field is also around 6 T. The overall length of the magnet is around 1.42 m.

Although EAs/EDAs are general purpose optimization techniques, they may not be able to solve real world problems efficiently when used in a straightforward way. As shown by the results in this section, it is very important to utilize problem-specific knowledge whenever possible in complex situations. The heuristic used here, which is about the boundaries of each coil and specifies the spatial distribution of coils, has proven to be effective in identifying the general nature of a good solution so that EDA$_{mvg}$ could concentrate on promising area. In the meantime, this heuristic does not need to be very strict due to the global optimization ability of population-based algorithms.

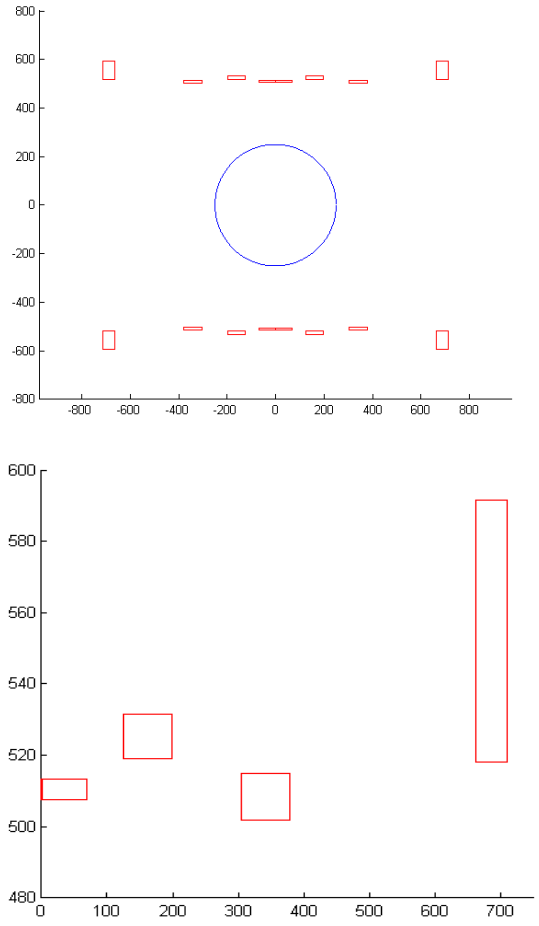

Figure 7. An 8-coil design (top) and a detailed view of the upper-right corner (bottom).

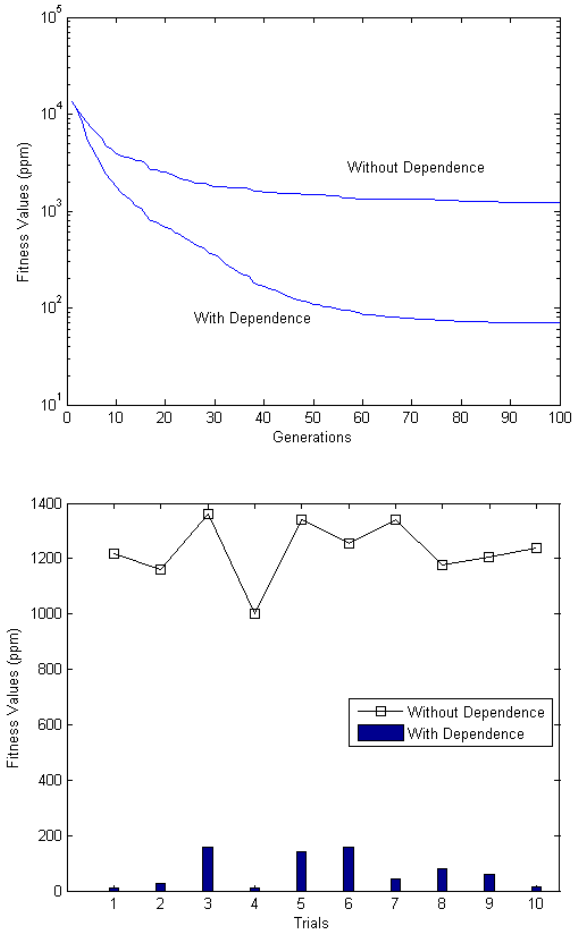

Figure 8. Comparison of the performance of EDA$_{mvg}$ with and without dependence.

4.3 Dependence vs. Independence

It is clear from the above results that significant dependencies exist among the problem variables. It is interesting to know whether EDA$_{mvg}$ is able to learn and utilize this dependence information. In EDA$_{mvg}$, the probabilistic model is represented by a multivariate Gaussian. Previously, a full covariance matrix was employed to explicitly capture the dependence among variables. A simplified version of EDA$_{mvg}$ (equivalent to UMDAc [8]) is to only use diagonal elements when generating new individuals. By doing so, no dependence is taken into account and each variable is generated independently of others.

In order to demonstrate the importance of capturing dependence in this optimization task, both versions of EDA$_{mvg}$ were run for 10 trials with the same setting as in the last experiment (dsv=50 cm). The mean fitness values of the best solutions found at each generation are plotted in Figure 8 (top) showing that EDA$_{mvg}$ found solutions of much lower quality when dependencies were not used. A pair-wise comparison (i.e., each pair of trials used the same random number seed) in terms of the quality of the final solutions is shown in Figure 8 (bottom) from which we can see that four trials out of 10 trials produced satisfactory results when capturing dependence. As a contrast, when no dependence was taken into account, all results were an order of magnitude worse.

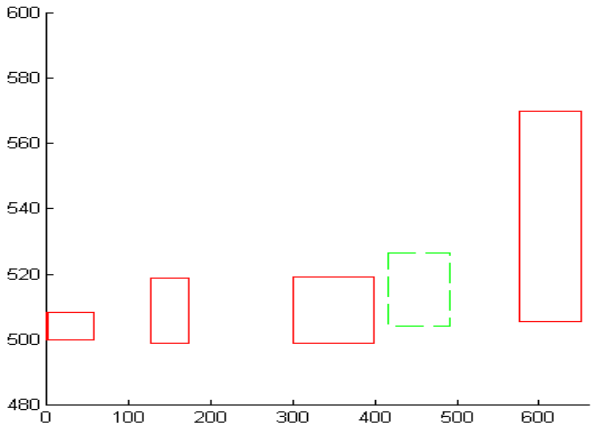

Figure 9. A 10-coil design with overall length 1.30 m (dashed line indicates a negatively wound coil).

4.4 Extensions

Additional experiments were also conducted to further test the potential of our method. With a larger population size (e.g., 2000) and some slightly different heuristics (e.g., the position of the right-most coil), even shorter magnet designs could be found. Figure 9 shows such a design with overall length 1.30 m and homogeneity 9.6ppm (dsv=45cm).

5. CONCLUSION

In this paper, the magnet design task in MRI systems was introduced as a challenging real-world problem for EAs. A continuous EDA was used to find the appropriate configuration of coils in a short magnet design. It has shown that within only a moderate number of fitness evaluations, various designs of high homogeneity with length around 1.4 m could be found reliably. Certainly, our current results are still not comparable with the best available designs, which were often based on very large-scale experiments. A straightforward direction for future work is to conduct further experiments to investigate the possibility of more advanced designs such as short magnets with length close to 1 m and asymmetric designs with dsv close to the open end. Usually, many more coils will be needed in such situations and a much larger population would be required accordingly.

This paper has discussed the application of a specific kind of algorithm to a specific real-world problem. Nevertheless, we have attempted to emphasize some important general issues such as problem characterization, dependence capture, using problem-specific knowledge and parallel computation, which we hope could be helpful to solving a wider range of real-world problems.

6. ACKNOWLEDGEMENT

This work was supported by an Australian Postgraduate Award granted to Bo Yuan.

7. REFERENCES

[1] Ansorge, R.E., Carpenter, T.A., Hall, L.D., Shaw, N.R. and Williams, G.B. Use of Parallel Supercomputing to Design Magnetic Resonance Systems. *IEEE Transactions on Applied Superconductivity*, *10*, 1 (2000), 1368-1371.

[2] Bäck, T. *Evolutionary algorithms in theory and practice*. Oxford University Press, New York, 1996.

[3] Caruana, R. and Mullin, M. Estimating the Number of Local Minima in Big, Nasty Search Spaces. In *Proceedings of IJCAI-99 Workshop on Statistical Machine Learning for Large-Scale Optimization*, 1999.

[4] Crozier, S. and Doddrell, D.M. Compact MRI Magnet Design by Stochastic Optimization. *Journal of Magnetic Resonance*, *127*, 2 (1997), 233-237.

[5] Eremeev, A. and Reeves, C.R. Non-parametric Estimation of Properties of Combinatorial Landscapes. In *Proceedings of EvoWorkshops 2002*, 2002, 31-40.

[6] Eremeev, A.V. and Reeves, C.R. On Confidence Intervals for the Number of Local Optima. In *Proceedings of EvoWorkshops 2003*, 2003, 224-235.

[7] Forbes, L.K., Crozier, S. and Doddrell, D.M. Rapid Computation of Static Fields Produced by Thick Circular Solenoids. *IEEE Transactions on Magnetics*, *33*, 5 (1997), 4405-4410.

[8] Larrañaga, P. and Lozano, J.A. (eds.) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

[9] Paul, T.K. and Iba, H. Real-Coded Estimation of Distribution Algorithm. In *Proceedings of The Fifth Metaheuristics International Conference*, 2003.

[10] Shaw, N.R. and Ansorge, R.E. Genetic Algorithms for MRI Magnet Design. *IEEE Transactions on Applied Superconductivity*, *12*, 1 (2002), 733-736.