Differential Evolution for RFID Antenna Design: A Comparison with Ant Colony Optimisation

Evolutionary Multiobjective Optimization track

ABSTRACT

Differential evolution (DE) has been traditionally applied to solving benchmark continuous optimisation functions. To enable it to solve a combinatorially oriented design problem, such as the construction of effective radio frequency identification antennas, requires the development of a suitable encoding of the discrete decision variables in a continuous space. This study introduces an encoding that allows the algorithm to construct antennas of varying complexity and length. The DE algorithm developed is a multiobjective approach that maximises antenna efficiency and minimises resonant frequency. Its results are compared with those generated by a family of ant colony optimisation (ACO) metaheuristics that have formed the standard in this area. Results indicate that DE can work well on this problem and that the proposed solution encoding is suitable. On small antenna grid sizes (hence, smaller solution spaces) DE performs well in comparison to ACO, while as the solution space increases its relative performance decreases. However, as the ACO employs a local search operator that the DE currently does not, there is scope for further improvement to the DE approach.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

Differential evolution, multiobjective optimisation, RFID antenna design

1. INTRODUCTION

Since its inception in 1948 [23], radio frequency identification (RFID) has become one of the major methods for the tracking and identification of goods and items, particularly within logistics and supply chains [9]. The technology has many uses, an example of which is the tracking and identification of luggage coming through airports. Hong Kong International Airport currently processes approximately 40,000 pieces of luggage a day (for departing flights) using the RFID system. This is far more reliable than using barcodes, with the read-rates being at 97% as opposed to only 80% [7]. It is anticipated that the next large scale application of RFID will be to replace the universal price code (barcoding) system used for the purchase of goods at retail outlets.

An RFID system consists of two basic components: a reader and a tag (containing an antenna). The reader sends an RF signal which can power the receiver (the tag). This in turn will radiate back a signal to the reader [21]. This backscattered signal usually contains a number that uniquely identifies the tag/item. The major design objective for the antenna is improving the read range (the distance the signal can be sent and received). Two major factors affect the effectiveness of the system. The read range is generally inversely proportional to the resonant frequency and proportional to the antenna gain (related to the antenna efficiency). Both these factors are defined by the design of the antenna, and hence become an optimisation problem. It is desirable to minimise resonant frequency (f_0) and maximise efficiency (η). This is achieved by producing antennas that maximise the length of the antenna in a convoluted space-filling manner. RFID antennas are usually designed in such a way that they form meander lines as dipole structures. An example of such an antenna is given in Figure 1. Note that these antennas can be laid out on a Cartesian grid and are symmetrical around the dipole, so only one half of the antenna needs to be designed by an algorithm.

Producing a meander line antenna is essentially a constructive activity. Hence, much research has concentrated on the design of antennas using the constructive metaheuristic ant colony optimisation (ACO) [10, 11, 19, 25]. This paper explores the application of the iterative metaheuristic differential evolution (DE) to such a process. The problem of creating a meander line is related to two famous problems: the travelling salesman problem (TSP) [20] and that of creating self-avoiding walks (SAWs) [16, 22]. It is similar to the

Figure 1: (a) A 5×5 antenna grid. (b) A meander line RFID antenna defined on that grid. (c) The complete dipole antenna produced by mirroring the solution shown in (b); the two halves are joined by a 6mm bridge.

former as each of the points on the Cartesian grid is to be connected to make a path (although viable antennas need not use every point). However, the major difference is that for RFID each of the points connects to only four neighbouring points (those directly adjacent) in a horizontal and vertical direction. In contrast, each point on a TSP has potential connection (albeit at a cost) to every point in the graph. Thus it may be considered as a severely constrained version of the TSP. It also resembles a SAW as a meander line antenna may not cross itself. However, SAWs find common application in chemistry, where molecules may expand to infinite space, whereas RFID antennas have a specified grid size, thus restricting such activity.

This paper examines how a non-constructive metaheuristic may be implemented so that it can solve the problem of producing efficient antennas that also maximise read range. Importantly, it also provides much needed comparative results to the benchmark ACO solvers that have been successfully applied to this problem. The remainder of the paper is organised as follows. Section 2 describes the various iterations of the ACO metaheuristic that have been used. From this, DE and how it is adapted to solve the constructive problem is given in Section 3. Section 4 shows the results of DE, comparing them to those of previous ACO algorithms. Finally, Section 5 contains the conclusions and outlines future research avenues.

2. RFID ANTENNA DESIGN BY ACO

By and large, the family of ACO metaheuristics use algorithms that progressively add components to an initially empty solution until a complete solution is produced. This complements how a meander line is constructed, one seg-

ment (between adjacent grid points) at a time. It is therefore not surprising that ACO was one of the first metaheuristics used to automatically design RFID antennas. Until that time, engineers would routinely design them by hand, using skill and experience to attempt to obtain good antennas. Gelehdar, Thiele and O'Keefe [6] were the first to explore the search space of the problem. However, this was done enumeratively for a 5×5 antenna (a very small size).

The initial paper by Randall, Lewis, Galehdar and Thiele [19] outlined the canonical form using the variant of ACO known as Ant Colony System (ACS). In that work, ants progressively add segments to the meander line until either a complete Hamiltonian path is produced or it becomes trapped (i.e., another segment cannot be added without crossing the meander line). In the latter case, the ant discards the solution and does not perform a pheromone update. A feature that was found to be necessary was a bias toward straight line sections of segments. This minimised the chance that ants would become stuck. Additionally, a cache was kept that held previously found solutions. This was necessary as the antenna evaluation software, known as NEC [2], requires a disproportionally large amount of computational time to evaluate a solution. Overall, very efficient antennas for a range of grid sizes (from 5×5 to 10×10 grid points) were found.

To further improve the efficiency results, a specialised form of local search was added, known as *backbite* [25]. As the antennas are tightly packed within the grid, it is difficult to alter an existing design to derive a new and, hopefully, better one. The backbite operator originated in the SAW literature and is able to do this by making slight modifications to the end of the meander line. It could take a meander line and produce a tree of slightly different alternative solutions. While the application of backbite consumed a modest amount of computational resources, it, on average, was capable of increasing efficiencies by a few percent over those of Randall *et al.* [19].

The two previous papers only considered efficiency as the objective measure. Lewis *et al.* [10, 11] added the extra objective of minimising resonant frequency. They investigated both low and high frequency antennas and relaxed the requirement that antennas had to be full Hamiltonian paths. From this, attainment surfaces of highly efficient antennas with relatively low resonant frequencies were produced.

3. DIFFERENTIAL EVOLUTION FOR RFID ANTENNA DESIGN

Differential evolution (DE) [24] is a population-based search technique that operates in continuous domains and which has been applied successfully to many different problems [18]. The operation of the algorithm in a single-objective setting is described first, followed by a survey of adaptations to multiobjective problems in Section 3.2 and subsequently details of its adaptation to the discrete RFID antenna design problem are presented in Section 3.3.

3.1 Differential Evolution

In each iteration of the DE algorithm, each member of the population of solutions is considered in turn as a *target* for replacement in the subsequent generation. A new candidate solution is generated by adding the weighted difference between two randomly chosen population members (hereafter

referred to as x_{r2} and x_{r3}, neither of which is the *target*) to a third, randomly-chosen population member, referred to here as the *base*. Typically, some form of crossover is then performed between the *target* and candidate; this further modified candidate is the one that may replace the *target*.[1]

The three main variants of DE are labelled DE/rand/1/*, DE/best/1/* and DE/target-to-best/1/*, which add (respectively) a difference vector generated from two members of the population to a randomly chosen individual, the best individual or to some point between the *target* and best individual. The most common, and frequently effective [14], variant is DE/rand/1/*. The * may be either *bin* for a uniform crossover, where the probability of mutating a component follows an approximately binomial distribution, or *exp*, where a sequence of vector components is taken, the length of which follows an inverse exponential distribution. There also exist other variants that use arithmetic crossover. In this work the DE/rand/1/exp variant is used (as initial experimental work found it to be more effective on this problem than DE/rand/1/bin), so its version of the DE mutation mechanism will be explained.

Let $S = \{x_1, x_2, \ldots, x_{|S|}\}$ be the population of solutions. In each iteration, each solution in S is considered as a *target* for replacement by a new solution; denote the current *target* by x_i. A new point v_i is generated according to

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{1}$$

where x_{r1} (i.e., the *base*), x_{r2} and x_{r3} are distinct, randomly selected solutions from $S \setminus \{x_i\}$ and F is the scaling factor, typically in $(0, 1]$ although larger values are also possible. The candidate solution u_i is produced by performing crossover on v_i, controlled by the parameter $Cr \in [0, 1]$, by taking a (wrapping) sequence of L components from v_i starting at a randomly chosen index k. The length L is determined by generating, for each component after k, a uniform random number and comparing it against the crossover parameter Cr; if the random number is greater than Cr the remaining components are taken from the *target*. The probability that a sequence will have a particular length h is given by $P(L = h) = (1 - Cr)Cr^{h-1}$ [18].

$$u_i^j = \begin{cases} v_i^j & \text{if } j \in \{k, \langle k+1 \rangle, \ldots, \langle k+L-1 \rangle\}, \\ x_i^j & \text{otherwise} \end{cases} \tag{2}$$

In single-objective DE the *target* is replaced if the new solution is as good or better.

3.2 Multiobjective Differential Evolution

The following assumes some familiarity with multiobjective optimisation; good introductions to the topic can be found in Coello Coello [3] and Deb [4]. In typical applications of DE to single-objective problems the population of solutions converges to a single location; indeed, this convergence is often a necessary feature of the algorithm that allows it to automatically scale the magnitude of moves it makes in solution space [15]. However, this behaviour is not desirable in a multiobjective application where a diverse range of solutions spread along the non-dominated front is sought. Mezura-Montes, Reyes-Sierra and Coello Coello [13] categorise DE algorithms for multiobjective optimisation into non-Pareto-based approaches, Pareto-based

approaches and combined approaches, the last of which is not discussed here. Non-Pareto-based approaches may treat some objectives as constraints or use aggregation functions to collapse several objectives into one. As the precursor studies to that described here do not take this approach, such DE algorithms are not considered further. Pareto-based approaches maintain a collection of solutions representing one or more non-dominated fronts. Mezura-Montes *et al.* further divide these into approaches that use *Pareto dominance* to select the better solution from *target* and candidate, and those that use *Pareto ranking*, where dominance is used to rank generated individuals by the front in which they appear. The first is effectively a translation of the original DE into the multiobjective realm, where each candidate solution competes only with one of its parents, namely the *target*. The second is most similar to other multiobjective evolutionary algorithms, such as Deb *et al.*'s NSGA-II [5], involving a $(\mu + \lambda)$-selection after all candidate solutions have been produced. Key exemplars of this approach are Madavan's [12] Pareto-Based Differential Evolution (PBDE), which uses a DE/current-to-rand/1/bin algorithm and the non-dominated sorting and ranking of NSGA-II, Xue *et al.*'s [26] Multi-Objective Differential Algorithm (MODE), and Iorio and Li's [8] Nondominated Sorting Differential Evolution (NSDE), which is effectively identical to NSGA-II except the mutation operator is replaced by DE/current-to-rand/1.[2]

The algorithm described in this work most closely follows NSDE. Many previous Pareto-based DE algorithms use the current-to-rand DE variant, as they are applied to continuous domains where that operator has some advantages (e.g., it is rotationally invariant). As this work describes a novel adaptation of DE to a discrete problem, more traditional DE mutation operators have been investigated first: DE/rand/1/bin and DE/rand/1/exp.

3.3 Adapting DE for RFID Antenna Design

Adapting continuous solvers to discrete problem domains is generally a non-trivial task (see Onwubolu and Davendra [17] for several examples). A naïve approach could be to treat each discrete decision variable in the problem as a dimension of the continuous space, with continuous values mapped to discrete values by rounding or truncation. Yet in constrained problems this may produce a complex search space where many positions are infeasible. Rather than encode antennas directly in a continuous space, the proposed approach describes the antenna construction process. As noted above the problem has some similarities with the TSP in that a meander line antenna is a sequence of visited nodes. A common approach taken with permutation problems is to order vector components by their value, thereby producing a permutation of the components. Yet while a meander line antenna can be described as a permutation of the nodes visited, given the restrictions on which nodes may be connected to each other this approach is considered to be inappropriate here.

Similar to prior work with ACO [10, 11, 19, 25], antennas are built via a series of decisions concerning which edge to add next. However, given the similarity of the antenna design problem to that of producing self avoiding walks, the

---

[1]At a minimum, one randomly chosen vector component will always come from the initially generated candidate point.

[2]The identifiers *current-to-rand* and *target-to-rand* are synonyms; the latter is used by the authors here to match the use of the label *target*.

approach here uses relative rather than absolute directions, as such a representation has previously been found to be more effective in evolutionary algorithms for the SAW [1]. All solutions are constructed from node 1 (top-left in antenna diagrams used here) with an assumed initial direction pointing down, and proceed by moving along edges either *(L)eft*, *(S)traight* or *(R)ight* from successive nodes, until no further edges can be constructed. The maximum number of such instructions on an $m \times m$ grid is $m^2 - 1$. Thus, each DE solution is a vector in $n = m^2 - 1$ space, where the first dimension describes the relative direction to move from node 1, the second dimension the direction to move from the next node, and so on. Each dimension is over the (arbitrary) range $[0, 3]$, and is divided into three areas corresponding to L, S and R, respectively. To encourage the construction of longer antennas, the interpretation of a component's value is altered adaptively such that only those feasible directions are represented. For example, if only the directions L and S are possible from a given node, the corresponding dimension's range is considered to be divided in two, the lower half representing the direction L and the upper half the direction S. Consequently, a value in $[0, 1)$ represents a *tendency* to go left at that point, a value in $[1, 2)$ a tendency to go straight, etc. In this way the solution representation in continuous space has an intuitive correspondence with its discrete counterpart. Figure 2 illustrates this solution encoding and how it is interpreted during antenna construction.

Initial testing revealed that antennas with high efficiency, η, but poor (high) resonant frequency, f_0, are created by the algorithm relatively easily. Because these solutions are non-dominated they are maintained in the fixed-size population, which may prevent the algorithm from properly exploring the space of solutions with lower f_0. To encourage exploration of this space the DE may be run with a constraint on the minimum length of antennas; the minimum length was set to half the maximum length. When this constraint is active (denoted DE_{minL}) an antenna below half the maximum length is considered to be dominated by any antenna that is at or above that threshold. The standard dominance relation is applied between antennas that are either both 'too short' or both acceptable.

To improve the time efficiency of the algorithm a solution cache was maintained.[3] When a previously evaluated solution is produced its quality is obtained from this cache. In some runs on the 10×10 grid (which has the largest solution space and longest solution evaluation time), up to 25% of generated solutions have been previously seen, resulting in a substantial decrease in runtime.

This proposed approach was found to be the most effective found among a number of competing schemes. These alternatives are evaluated and compared in a work currently in preparation and are not discussed further here.

4. COMPUTATIONAL EXPERIENCE

As this is a novel application of DE to the RFID antenna design problem a sensitivity analysis was performed to select appropriate settings for its control parameters. This analysis is described in the next section, with the experimental set up and results described in subsequent sections.

---

[3] The previous ACO for this problem also maintains a solution cache, although the implementations are quite different.
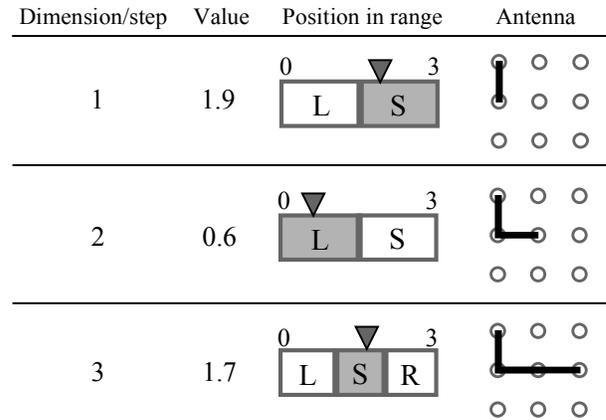

Figure 2: Decoding a vector in continuous space to construct an RFID antenna. The shaded portion of each range indicates the direction chosen. The range of each dimension $[0, 3]$ is independent of the grid size.

4.1 Sensitivity Analysis

DE's crossover rate Cr has a large impact on the algorithm's behaviour. Thus an initial sensitivity analysis was performed to assess the algorithm's performance at low and high values. The *exp* and *bin* crossover variants were compared as part of the analysis, with $Cr \in \{0.05, 0.9\}$ for *bin* and $Cr \in \{0.63, 0.99\}$ for *exp* so that the probability of mutating a component was equivalent to *bin* (see Zaharie [28] for suitable equations). The scale factor F was fixed at 0.8; low values lead to convergence [27], which is undesirable in multiobjective optimisation. The different combinations of low or high Cr and *exp* or *bin* were compared in terms of the quality of the final front achieved as well as the expected change in antenna length given that of the parent (*target*). A common heuristic in RFID antenna design is that longer antennas are better—certainly there is a correlation between antenna length and read distance—so measuring the ability of different settings to promote or maintain long antennas is important.

Conceivably low values of Cr would be expected to cause the smallest change to the resultant antenna structure and thus be less likely to "break" a long structure, but also less likely to rapidly generate longer antennas. Conversely, high values of Cr produce larger changes in solutions and thus would be expected to promote longer antennas with the attendant risk of generating much smaller structures from larger ones. Results from the sensitivity analysis confirm these suppositions: high values of Cr produce the largest positive changes to initially short antennas while low values produce smaller changes and are more likely to preserve long structures. Initial, randomly generated solutions are typically fairly short, so this could suggest that Cr should be changed dynamically with antenna length. However, the quality of the final fronts achieved after a fixed number of function evaluations is similar, regardless of the value of Cr used. Further, fine tuning of solutions may be better achieved by the use of a local search heuristic based on the previously successful *backbite* operator; this is yet to be inte-

grated with DE. Similarly, as solutions represent a sequence of construction instructions it was believed that *exp* may help preserve contiguous blocks of instructions more faithfully than *bin* and hence perform better overall, yet both variants achieve comparable results. The speed with which these results are achieved is, however, not the same, with *exp* crossover and $Cr = 0.99$ producing good results most quickly. This variant and value of Cr were used in all subsequent experiments.

4.2 Experimental Set Up

The DE algorithm was applied to grid sizes from 5×5 to 10×10. As in Lewis *et al.* [11] the track width was fixed at 1mm and all grids have dimensions 25×25mm. DE/rand/1/exp's control parameters were $Cr = 0.9$, $F = 0.8$ with a population size of 100. While it has been suggested by DE's creators that population size should be 10 times the number of dimensions—which would correspond to populations between 240 and 990 here—both initial experiments and previous findings concerning the behaviour of DE with smaller populations (see, e.g., Montgomery [15]) suggest that a population of 100 provides good performance on this problem. Note that DE's fixed population also fixes the maximum number of solutions that can be held in a single front.

A random control was implemented and works as follows: at each iteration all candidate solutions are produced randomly (i.e., as if a new initial population were being created) and then non-dominated sorting is performed to keep the best individuals from those produced in previous iterations and the new solutions. This provides an indication of how easily certain parts of the front are attained with the chosen representation and can also indicate if the DE algorithm and its mutation operator confers some benefit on the search. In tables and charts this approach is indicated by Rand or Rand$_{minL}$ when the minimum length constraint was used.

The DE and Rand algorithms were each allowed to produce 10,000 solutions in addition to the initial, randomised population of 100 individuals. This is 1% more than the number produced by the prior ACO, which produced 10,000 solutions in total. However, the total number of (costly) function evaluations is considerably less than 10,000 given the use of the solution cache. As noted, both ACO and DE made use of a solution cache to reduce computational cost.

Results from the DE were compared with those from the earlier ACS algorithm used in [10, 11] run with three different levels of greediness: $q \in \{0.1, 0.5, 0.9\}$.[4] These three variants are designated ACO$_{q1}$, ACO$_{q5}$ and ACO$_{q9}$, respectively. It has been common practice to run each of the three ACO variants concurrently and aggregate the results. As may be noted by inspection of the results obtained, this confers some slight improvement in quality of results obtained, particularly for larger grids.

4.2.1 Implementation Differences Affecting the Comparison

Although the comparison presented in the next section is intended to be fair to both DE and prior ACO there are some key differences in the approaches and results that will affect the comparison:

---

[4]In ACS, the parameter q is the probability that a greedy decision is made instead of a probabilistic one.
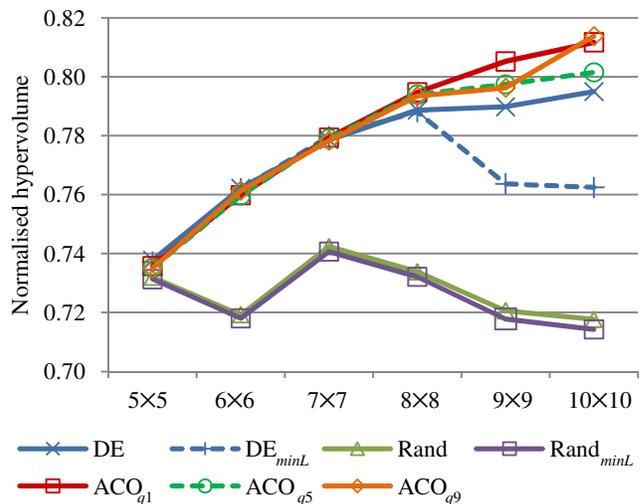

Figure 3: Hypervolumes produced by each algorithm across antenna grid size. Values exist only at markers; lines have been added as a guide to trends between grid sizes.

- In the current DE all antennas start from node 1, while in the previous ACO antennas may start from any node on the edge of the grid; antenna designs are then rotated such that the starting point is in on the midline of the mirrored dipole antenna structure. This limitation of the current DE will impact the range of solutions it can explore. A future version will support antennas that start from other nodes.

- Available ACO results were produced by an ACO algorithm that makes use of the backbite local search operator, which is not yet employed by the DE. Local search will be incorporated into a future version of the DE.

- The NEC evaluation software is under active development, with the DE-produced antennas evaluated against the version available in late October 2010. ACO results were produced using an earlier version that evaluates the same antenna structures slightly differently. To ensure a fair comparison the solutions obtained from Lewis *et al.* [11] have been re-evaluated against the same version of NEC as used by the DE, with these revised values used in the next section. In some cases re-evaluated solutions are now dominated by other front members; these newly-dominated solutions have been removed from the comparisons.

4.3 Results

Attainment surfaces (i.e., the final fronts) were compared using hypervolume (also called the S-metric) and the C-metric (see, e.g., Zitler and Thiele [29]). Hypervolume measures the volume (area in this instance) of dominated space under an attainment surface. The C-metric, denoted $C(A, B)$, describes the proportion of points in the surface produced by B that are dominated by at least one point produced by A. In hypervolume calculations here the objective of minimising f_0 was transformed into a maximisation objective by
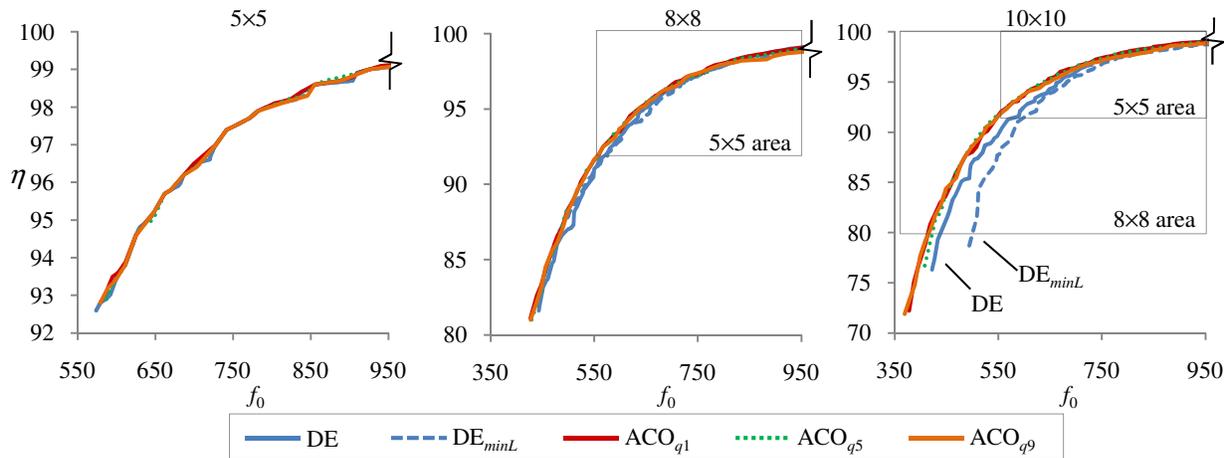
Figure 4: **Lower portions of fronts produced by DE and ACO for 5×5, 8×8 and 10×10 grids; regions not shown for poor resonant frequency but very high efficiency are largely indistinguishable between algorithms.**

subtracting values from $2,250$.[5] In addition, η values, which have a natural upper bound of 100, were multiplied by 20 to produce modified objective values of similar magnitude. Reported hypervolumes were normalised by the maximum possible—but not practically achievable—area determined by these upper bounds of 2,250 for η and $20 \cdot 100 = 2,000$ for f_0.

Hypervolume results are shown in Figure 3. The upward trend with grid size is because larger grids allow for longer antennas, which expands the solution space and, consequently, dominated objective space for f_0. Both hypervolume and C-metric results indicate that DE outperforms Rand, with DE's attainment surfaces typically dominating many or all points produced by Rand or $Rand_{minL}$ (i.e., $C(DE, Rand) = 0.4$ for the 5×5 grid and $C(DE, Rand) > 0.85$ for 6×6 and larger grids; $C(Rand, DE) < 0.01$ across all grid sizes). This strongly suggests that the DE mutation mechanism *is* able to generate novel and improving solutions for this problem using existing solutions. This is an important finding, given that the proposed solution encoding with DE search is not a 'natural' approach to this problem, in particular when compared with the direct construction of solutions used in the earlier ACO algorithms.

On grids 5×5 through 7×7, DE performed similarly to ACO in terms of hypervolume achieved, with poorer performance on larger grids. C-metric values comparing DE and ACO, presented in Table 1, show that DE compares favourably on grids 5×5 and 6×6, but its solutions tend to be dominated on larger grids, even though the attainment surfaces are very close. This is further illustrated in Figure 4, which shows the outline of the attainment surfaces on grids 5×5, 8×8 and 10×10.

If only the hypervolume and C-metric results are considered, DE_{minL} appears to perform poorly in relation to DE. Certainly this is the case on the large grids of 9×9 and 10×10, where it is possible that it retards the search by eliminating solutions that are necessary parents for good

---

[5]A single outlier point was removed from four of the 42 attainment surfaces examined to ensure the upper bound of 2,250 was a good fit to the majority of points.
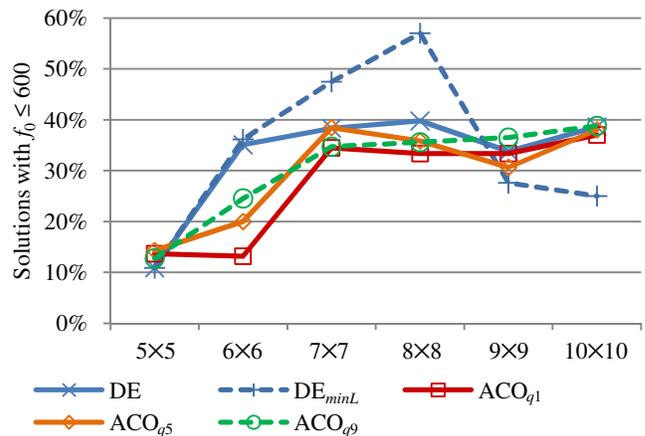

Figure 5: **Proportion of solutions with resonance frequency ≤ 600, by grid size and algorithm. Values are only valid at the markers; lines have been added as a visual aid.**

solutions. Further, in the relatively small search spaces of grids 5×5 and 6×6 DE and DE_{minL} perform almost equivalently, sharing 83% and 47% of final solutions, respectively. However, on grids 7×7 and 8×8, use of the minimum antenna length constraint has a positive impact on the distribution of solutions, achieving its aim of producing more solutions with a relatively low f_0. Figure 5 presents the proportion of solutions in attainment surfaces produced by DE and ACO where $f_0 \leq 600$ and illustrates its efficacy on these grid sizes. If only points where $f_0 \leq 600$ are considered, $C(DE_{minL}, DE) = 0.83$ for the 7×7 grid and $C(DE_{minL}, DE) = 0.71$ for the 8×8 grid, the latter being substantially higher than when the entire front is considered. In future work, the use of the minimum length constraint at a later stage in the search will be investigated.

Visual inspection of the attainment surfaces produced by DE (and DE_{minL}) on the largest grid size suggests that the

algorithm will continue to make improvements if afforded more function evaluations. DE and DE_{minL} were rerun on the 10×10 grid and allowed to produce 20,000 solutions. Results from these double-length runs show that they do continue to improve, although their final results are still dominated by those of ACO. Since the ACO shows improved performance when its backbite-based local search procedure is employed, the use and efficacy of this technique with the DE algorithm will be a subject of future investigation.

5. CONCLUSIONS

ACO was the first metaheuristic to be applied to the problem of designing generalised meander-line RFID antennas for high efficiency and low resonant frequency. It appeared an obvious choice, due to the natural fit between the constructive nature of the algorithm and the necessary task of constructing the path for an antenna element as part of the optimisation problem. The continuous solver DE is not a natural fit to this problem and thus required development of a suitable solution encoding and antenna construction scheme. A novel encoding and construction mechanism have been described that not only map from continuous to discrete space, but also deal with the severe move limitations that characterise this problem. This clearly demonstrates that it is possible to implement a non-constructive metaheuristic for a problem that is essentially constructive in nature. Moreover, the described DE produces results that are competitive with ACO.

Without another algorithm for comparison, until now it has been difficult to confirm that the previous ACO algorithms were producing attainment surfaces that approached the true Pareto front for this problem. This is the first study to present such a comparison. Results here confirm that *both* methods can create good antenna designs and lend support to the claim that ACO's results are approaching the true Pareto front.

Considering the limitations of the current DE (outlined in Section 4.2.1), its comparative performance is impressive. At present the quality of results delivered by ACO are better than those of DE, particularly on larger grids at lower resonant frequencies. However, it may be noted that, in the experiments reported in this work, DE makes use of a population of 100 trial solutions, while ACO only uses 10 ants in each iteration. This implies that, given sufficient parallel resources, DE can deliver results of comparable quality significantly faster than ACO due to its greater concurrency. While it is reasonable to assume that increasing the size of the ant population may accelerate evolution of solutions, the extent to which this effect might be observed is a subject for further investigation.

This work expands the range of available techniques for generating antenna designs. It is conceivable that the same approach can be used with other continuous metaheuristics, such as PSO. Future work will investigate the relative performance of both algorithms, addressing limitations of the current implementation of DE by allowing antennas to be built from a range of starting points and investigating the use a backbite-based local search operator, and assessing the effect of increased population sizes in ACO.

6. ACKNOWLEDGEMENTS

The authors wish to thank Andrew Lewis and Marcus

Table 1: C-metric comparisons for DE and ACO. Cell (row, col) corresponds to $C(A, B)$, which is the proportion of solutions produced by B that are dominated by solutions produced by A

5×5	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	0	0.03	0.2	0.17
DE_{minL}	0.04	—	0.01	0.16	0.15
ACO_{q1}	0.22	0.17	—	0.29	0.34
ACO_{q5}	0.04	0.04	0.08	—	0.15
ACO_{q9}	0.04	0.04	0	0.04	—
6×6	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	0.2	0.33	0.39	0.51
DE_{minL}	0.16	—	0.29	0.39	0.47
ACO_{q1}	0.14	0.24	—	0.3	0.43
ACO_{q5}	0.17	0.24	0.14	—	0.43
ACO_{q9}	0.1	0.13	0.08	0.13	—
7×7	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	0.17	0.33	0.18	0.18
DE_{minL}	0.73	—	0.31	0.23	0.27
ACO_{q1}	0.56	0.39	—	0.28	0.24
ACO_{q5}	0.63	0.47	0.24	—	0.22
ACO_{q9}	0.53	0.39	0.21	0.12	—
8×8	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	0.47	0.16	0.26	0.34
DE_{minL}	0.34	—	0.02	0.08	0.25
ACO_{q1}	0.78	0.77	—	0.4	0.59
ACO_{q5}	0.69	0.65	0.25	—	0.51
ACO_{q9}	0.66	0.58	0.09	0.19	—
9×9	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	0.88	0.17	0.25	0.3
DE_{minL}	0.09	—	0.01	0.04	0.15
ACO_{q1}	0.82	0.98	—	0.57	0.61
ACO_{q5}	0.73	0.97	0.17	—	0.54
ACO_{q9}	0.66	0.84	0.1	0.16	—
10×10	DE	DE_{minL}	ACO_{q1}	ACO_{q5}	ACO_{q9}
DE	—	1	0.11	0.2	0.39
DE_{minL}	0	—	0	0	0.01
ACO_{q1}	0.89	1	—	0.41	0.7
ACO_{q5}	0.81	1	0.4	—	0.66
ACO_{q9}	0.65	0.95	0.12	0.1	—

7. REFERENCES

[1] D. Ashlock, K. Bryden, S. Corns, and J. Schonfeld. An updated taxonomy of evolutionary computation problems using graph-based evolutionary algorithms. In *IEEE Congress on Evolutionary Computation, CEC 2006*, pages 403–410, 2006.

[2] G. Burke, A. Poggio, J. Logan, and J. Rockway. NEC — Numerical electromagnetics code for antennas and scattering. In *Antennas and Propagation Society International Symposium, 1979*, pages 147–150, June 1979.

[3] C. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.

[4] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2002.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

[6] A. Galehdar, D. Thiel, and S. O'Keefe. Antenna efficiency calculations for electrically small, RFID antennas. *IEEE Antenna and Wireless Propagation Letters*, 6:156–159, 2007.

[7] Hong Kong International Airport. HKIA boosts baggage handling efficiency with RFID technology. http://www.hongkongairport.com/eng/media/press-releases/pr_914.html [Accessed 19/1/2011], 15 January 2008.

[8] A. W. Iorio and X. Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence*, volume 3339 of *Lecture Notes in Artificial Intelligence*, pages 861–872. Springer-Verlag, 2004.

[9] M. Keskilammi, L. Sydänheimo, and M. Kivikoski. Radio Frequency Technology for Automated Manufacturing and Logistics Control. Part 1: Passive RFID Systems and the Effects of Antenna Parameters on Operational Distance. *The International Journal of Advanced Manufacturing Technology*, 21(10):769–774, 2003.

[10] A. Lewis, M. Randall, A. Galehdar, S. Thiel, and G. Weis. Using ant colony optimisation to construct meander-line RFID antennas. In A. Lewis, S. Mostaghim, and M. Randall, editors, *Biologically-Inspired Optimisation Methods: Parallel Algorithms, Systems and Applications*, volume 210 of *Studies in Computational Intelligence*, pages 189–217. Springer-Verlag, 2009.

[11] A. Lewis, G. Weis, M. Randall, A. Galehdar, and D. Thiel. Optimising efficiency and gain of small meander line RFID antennas using ant colony system. In *Proceedings of the Congress on Evolutionary Computation*, pages 1486–1492, 2009.

[12] N. K. Madavan. Multiobjective optimization using a pareto differential evolution approach. In *IEEE CEC 2002*, volume 2, pages 1145–1150. IEEE Press, 2002.

[13] E. Mezura-Montes, M. Reyes-Sierra, and C. Coello Coello. Multi-objective optimization using differential evolution: A survey of the state-of-the-art. In U. Chakraborty, editor, *Advances in Differential Evolution*, volume 143 of *Studies in Computational Intelligence*, pages 173–196. Springer, 2008.

[14] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello. A comparative study of differential evolution variants for global optimization. In *Genetic and Evolutionary Computation Conference (GECCO'06)*, pages 485–492, Seattle, Washington, USA, 2006.

[15] J. Montgomery. Crossover and the different faces of differential evolution searches. In *IEEE CEC 2010*, Barcelona, Spain, 2010. IEEE.

[16] R. Oberdorf, A. Ferguson, J. Jacobsen, and J. Kondev. Secondary structures in long compact polymers. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(5):051801, 2006.

[17] G. C. Onwubolu and D. Davendra, editors. *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, volume 175 of *Studies in Computational Intelligence*. Springer, 2009.

[18] K. Price, R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, 2005.

[19] M. Randall, A. Lewis, A. Galehdar, and D. Thiel. Using ant colony optimisation to improve the efficiency of small meander line RFID antennas. In *Proceedings of the 3$^{\mathrm{rd}}$ IEEE International e-Science and Grid Computing Conference*, 2007.

[20] G. Reinelt. *The traveling salesman: Computational solutions for TSP applications*. Springer-Verlag, Berlin, 1994.

[21] K. Seshagiri Rao, P. Nikitin, and S. Lam. Antenna design for UHF RFID tags: A review and a practical application. *IEEE Transactions on Antenna Propagation*, 53:3870–3876, 2005.

[22] A. Sokal. Monte carlo methods for the selfavoiding walk. *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*, pages 47–124, March 1994.

[23] H. Stockman. Communication by means of reflected power. In *Proceedings of the Institute of Radio Engineers*, pages 1196–1204, 1948.

[24] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11:341–359, 1997.

[25] G. Weis, A. Lewis, M. Randall, A. Galehdar, and D. Thiel. Local search for ant colony system to improve the efficiency of small meander line RFID antennas. In *IEEE Congress on Evolutionary Computation, 2008*, pages 1708–1713, 2008.

[26] F. Xue, A. C. Sanderson, and R. J. Graves. Pareto-based multi-objective differential evolution. In *IEEE Congress on Evolutionary Computation, 2003 (CEC 2003)*, volume 2, pages 862–869, Canberra, Australia, 2003. IEEE Press.

[27] D. Zaharie. Critical values for the control parameters of differential evolution algorithms. In R. Matoušek and P. Ošmera, editors, *MENDEL 2002, 8th International Conference on Soft Computing*, pages 62–67, Brno, Czech Republic, 2002.

[28] D. Zaharie. Influence of crossover on the behavior of differential evolution algorithms. *Applied Soft Computing*, 9(3):1126–1138, 2009.

[29] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms — A comparative case study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H. P. Schwefel, editors, *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, pages 292–301, Berlin, Germany, June 1998.