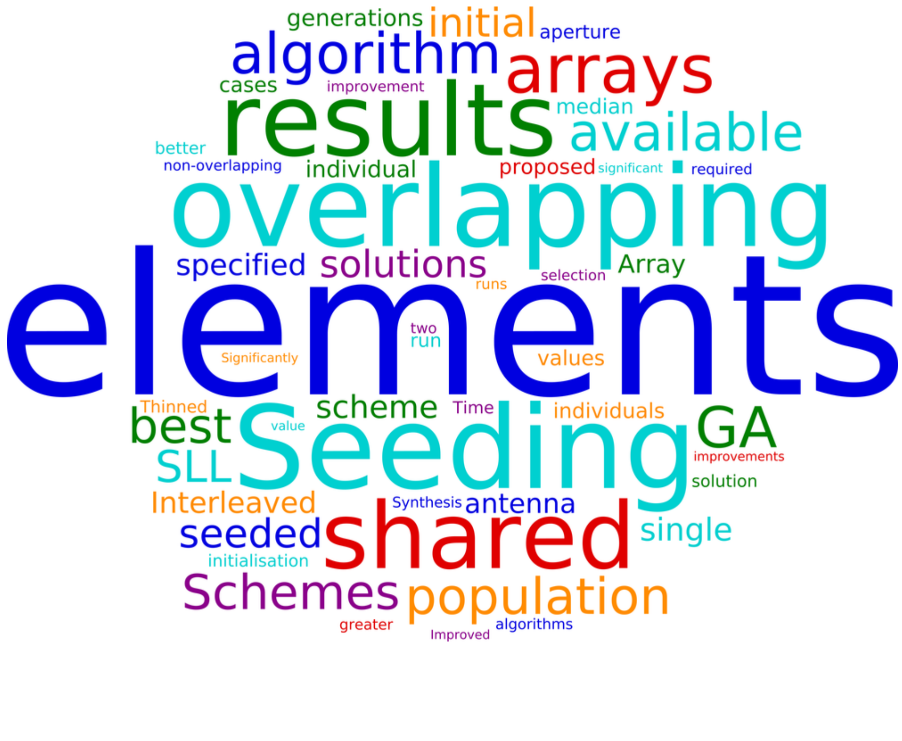
Submitted version of: W. P. du Plessis and A. bin Ghannam, "Improved Seeding Schemes for Interleaved Thinned Array Synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 11, pp. 5906-5910, November 2014.

ABBREVIATIONS

ADS	almost difference set
CDF	cumulative distribution function
CPU	central processing unit
CW	continuous wave
DS	difference set
GA	genetic algorithm
KACST	King Abdulaziz City for Science and Technology
NRF	National Research Foundation of South Africa
RAM	random-access memory
RF	radio frequency
SLL	sidelobe level



Improved Seeding Schemes for Interleaved Thinned Array Synthesis

W. P. du Plessis and A. bin Ghannam

***Abstract*—This paper considers the synthesis of interleaved antenna arrays with shared and inactive elements. A hybrid genetic algorithm (GA) where the initial population is seeded with good solutions is proposed. A number of seeding schemes are considered and the most effective of these are identified. The proposed algorithm is shown to reliably produce results with sidelobe level (SLL) values which are close to the optimum and to converge faster than the other algorithms considered. However, some of the seeding schemes mislead the GA and actually produce worse results than random initialisation.**

***Index Terms*—Interleaved arrays, sparse array antennas, thinned arrays, antenna arrays, initialisation, genetic algorithms.**

I. Introduction

Modern platforms are performing ever-increasing numbers of radio frequency (RF) functions, while simultaneously requiring a reduction in the number of antenna apertures [1]. Interleaved arrays are antenna arrays where different antenna elements within the array are allocated different functions, thereby achieving multiple functions within a single aperture [2]. Interleaved arrays have been developed to achieve wider operating bandwidths by interleaving differently-sized antenna elements [3], to achieve isolation between a transmitter and receiver in a continuous wave (CW) radar by interleaving transmit and receive elements [4], and to achieve multiple polarisations through the use of antenna elements with differing polarisations [5], [6].

Traditional interleaved arrays require each antenna element to be assigned to one of the sub-arrays as shown in Fig. 1(a). However, recent work suggests that an improved sidelobe level (SLL) can be achieved by also allowing antenna elements to be shared between sub-arrays or to be inactive as shown in Fig. 1(b) [7]. Examples of methods for sharing elements between a transmitter and receiver include switches (e.g. pulsed radar), circulators (e.g. CW radar) and duplexers (e.g. communications systems).

Unfortunately, the conclusions drawn from the results presented in [7] are limited by the performance of the algorithm used to generate the results. Improved synthesis algorithms are thus required to properly evaluate interleaved thinned arrays of the form shown in Fig. 1(b).

The use of genetic algorithms (GAs) in the synthesis of thinned antenna arrays (e.g. [8], [9]) and interleaved thinned arrays [2], [6], [7] is well established. Recent results have shown that seeding the initial population of a GA with good solutions (solutions whose SLL is better than that achieved by a random population) leads to hybrid algorithms which outperform their non-seeded counterparts for thinned-array synthesis (e.g. [10]–[12]). Based on observations about the form of the resulting solutions, a similar process was applied to the synthesis of interleaved thinned arrays and improvements to the synthesised results were obtained [7].

This paper presents an improved hybrid algorithm for the synthesis of interleaved thinned arrays through the use of a GA with a seeded initial population. A number of seeding schemes are evaluated, and the best of these identified. The hybrid algorithm is shown to lead to better results on a more consistently than the GA alone and than the other seeding schemes considered.

Manuscript received 27 October 2013; revised 5 July 2014. This work is based on the research supported in part by the King Abdulaziz City for Science and Technology (KACST) and by the National Research Foundation of South Africa (NRF) (Grant specific unique reference number (UID) 85845). The NRF Grantholder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are that of the author(s), and that the NRF accepts no liability whatsoever in this regard.

W. P. du Plessis is with the University of Pretoria, Pretoria, 0002, South Africa (e-mail: wduplessis@ieee.org).

A. bin Ghannam is with the King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia (e-mail: eng.ghannam1987@gmail.com).

Digital Object Identifier ??.????/TAP.2013.??????

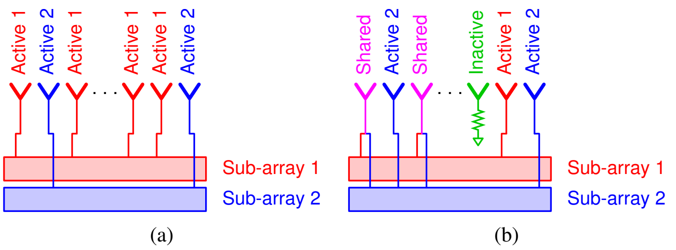


Fig. 1. Diagrammatic illustration of the difference between (a) traditional interleaved arrays and (b) interleaved arrays with shared and inactive elements.

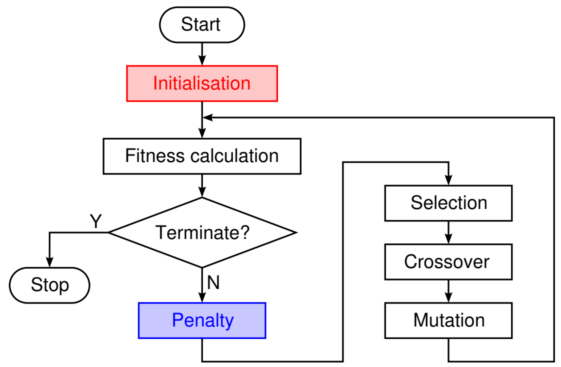


Fig. 2. The flowchart of the proposed algorithm with the modifications to a conventional GA highlighted.

II. Algorithm

The GA used here is based on the simple binary GA described by Goldberg [13], and a flowchart of the algorithm is provided in Fig. 2.

The initial population is randomly generated with some solutions set to good values as described below. A population size of 1000 is used, the GA is terminated after 150 generations, and selection is performed by exponential-ranking with a selection parameter (κ) of 0.005 [14]. The genetic operators used are uniform crossover with a probability of 0.9 and binary mutation with a probability of 0.01 [13]. Elitism is implemented by ensuring the best individual survives from each generation to the next [13].

The GA uses a representation composed of the concatenation of the two subarrays as shown in Fig. 3. The SLL of each subarray is independently computed using a half-wavelength spacing, omnidirectional antenna elements and equal weighting. The worse of the two subarray SLL values is used as the base fitness for that individual. The one drawback of this approach is that it does not allow control of the number of shared elements.

A penalty is thus added to each individual's base fitness to favour solutions with the specified number of shared elements (the overlap). The penalty in decibels is computed using

$$P = |O - O_s| \tag{1}$$

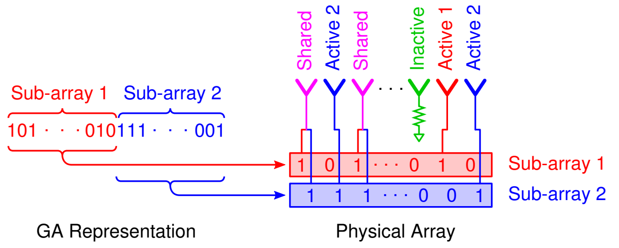


Fig. 3. The relationship between the representation used by the GA on the left and the physical array implied on the right.

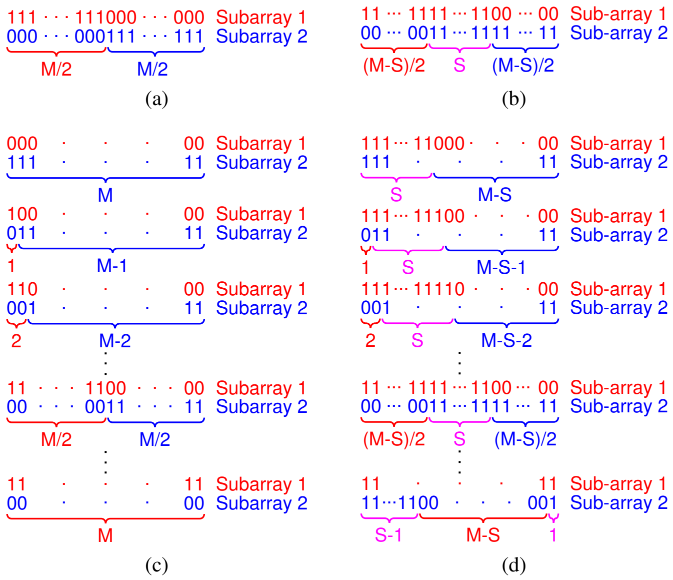


Fig. 4. The (a) single non-overlapping seeded individual [7], (b) single overlapping seeded individual, (c) multiple non-overlapping seeded individual and (d) multiple overlapping seeded individual initialisation schemes where M is the available number of elements and S is the number of shared elements.

where O and O_s are the number of shared elements of the solution being evaluated and the specified number of shared elements respectively. In this way, the correct overlap is ensured by the penalty-function formulation.

The objective function which was minimised by the algorithm is thus given by

$$F = \max\left(SLL_1, SLL_2\right) + |O - O_s| \tag{2}$$

where F is the fitness of a solution, and SLL_1 and SLL_2 are the SLL values of the two arrays in decibels.

The initial population is seeded with good individuals [13] using a number of seeding schemes which are summarised in Fig. 4.

The first initialisation scheme tested is that proposed by [15], where difference sets (DSs) and almost difference sets (ADSs) can be used to seed the initial population. The nature of this approach is that the configuration shown in Fig. 1(a) is used. The DS or ADS from [16], [17] was used to seed a single solution in the initial population.

The first new seeding scheme investigated is that described in [7], and is based on the observation that the two subarrays tend to have the majority of their elements on opposite sides of the array aperture. This seeding scheme initialises the two subarrays to each have half the available elements active, but on opposite sides of the aperture as shown in Fig. 4(a). While this seeding scheme dramatically improves the results for small numbers of shared elements, the results are less impressive for larger numbers of shared elements [7].

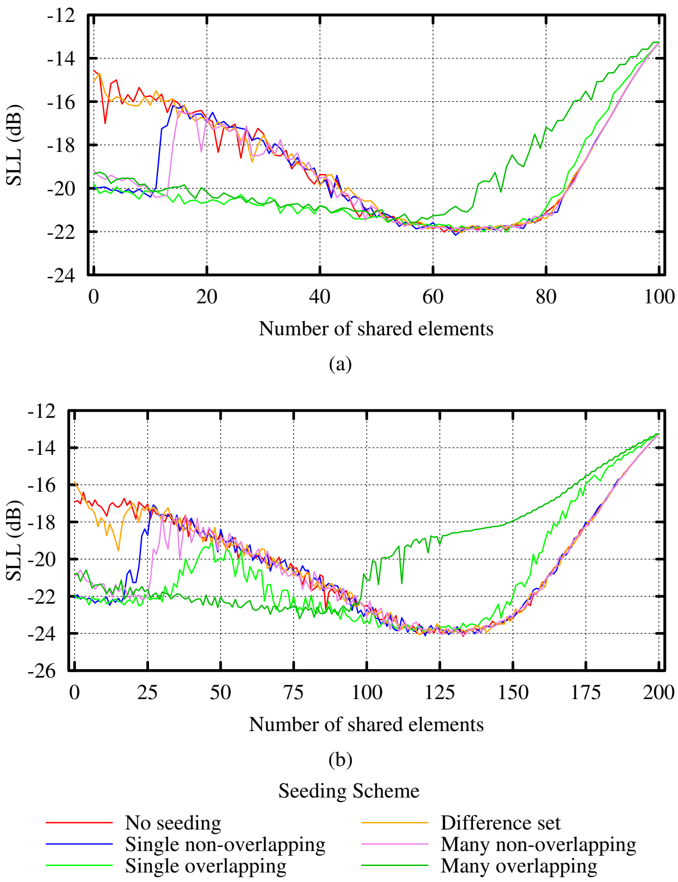


Fig. 5. The best results obtained for arrays with (a) 100 available elements and (b) 200 available elements.

In an effort to improve the results with larger numbers of shared elements, the seeding scheme in [7] was modified to achieve the specified number of shared elements at the centre of the array as shown in Fig. 4(b). In this way, the number of active elements in each subarray increases as the specified number of shared elements increases, while the shared elements are concentrated at the centre of the array as noted in previous results [7].

The one drawback of the above approaches is that only a single individual in the population is seeded, so the effect of the seeding is limited. While [7] has shown the effectiveness of this approach, seeding additional individuals is expected to produce improved results. The seeding schemes in Figs 4(c) and 4(d) are realised by shifting the transition from one subarray to the other in Figs 4(a) and 4(b) across the antenna aperture.

III. Results

The best and median results obtained from 51 runs of the GA for arrays of 100 and 200 available elements are shown in Figs 5 and 6 respectively. The general shape of the curves mirrors the results presented previously [7], with the notable difference that the results using the new seeding schemes are shown to generally have lower SLL in Fig. 5 and to be far more consistent in Fig. 6.

The improvement associated with the use of seeding is less noticeable with 100 available elements because the problem is simpler than in the 200 available-element case. This conclusion arises from the fact that there are 2^N possible solutions for an array with N elements, so increasing N leads to a rapid increase in problem complexity.

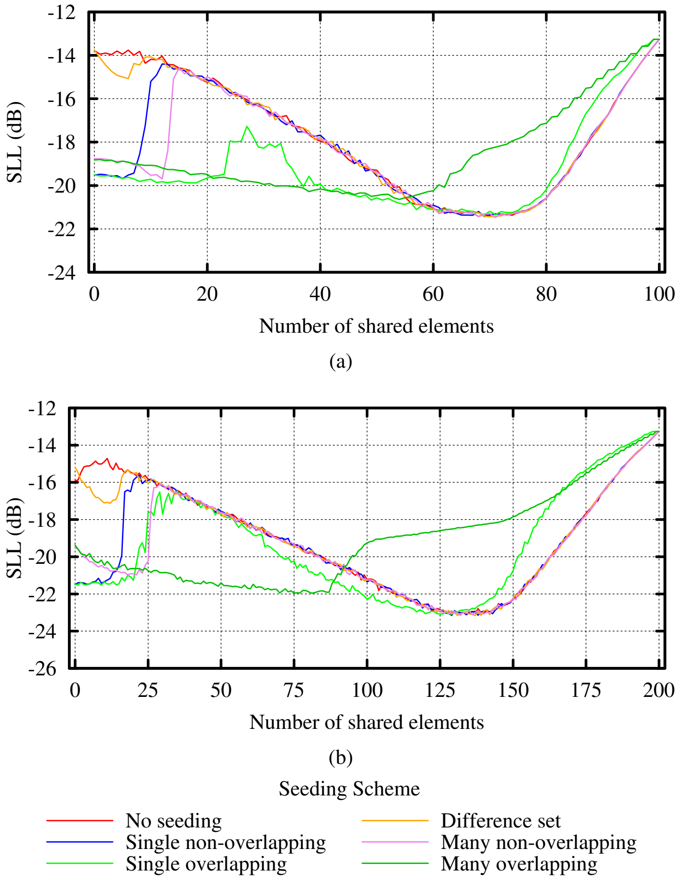


Fig. 6. The median results obtained for arrays with (a) 100 available elements and (b) 200 available elements.

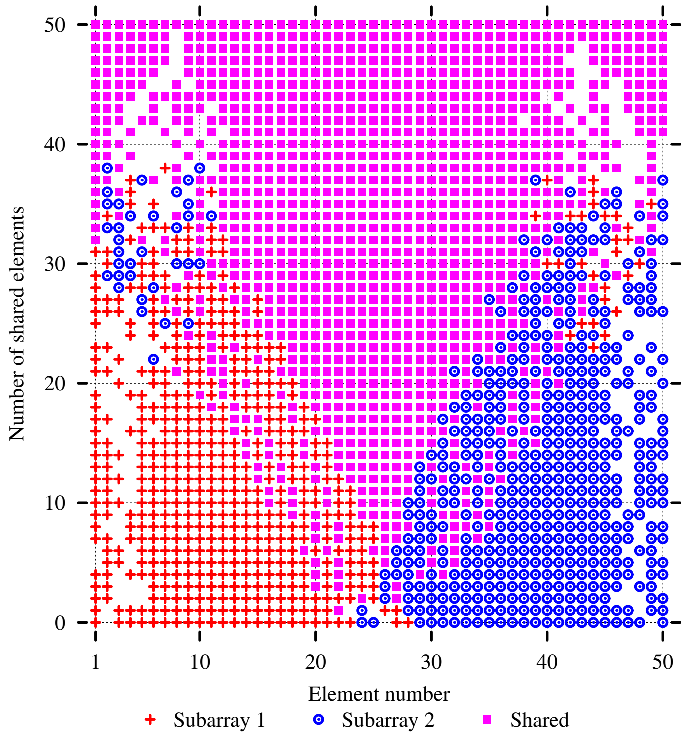


Fig. 7. The best excitations for an array with 50 available elements.

The variation in the best results in Fig. 5 is far greater than for the median case in Fig. 6. This outcome is anticipated because the best solutions from 51 independent runs of the GA are used, and even a single outlier will affect the best-case results. The median results are thus better indicators of the performance of the underlying algorithms because they are not as strongly affected by outliers.

Seeding with single individuals as shown in Figs 4(a) and 4(b) is seen to improve the results, despite the fact that there is only a small change to the initial population (only one of 1000 individuals is seeded). The implementation of elitism means that the single seeded individual survives to future generations until it is improved upon, so its effect on the algorithm can be greater than its effect on the initial population suggests.

Significantly, the results obtained when seeding with even the simplest of the proposed schemes (Fig. 4(a)) produces results which are significantly better than those obtained using DSs and ADSs for small numbers of shared elements and comparable results otherwise.

Still better results are obtained when a number of individuals are seeded using the schemes shown in Figs 4(c) and 4(d). The improvement is a result of the fact that a greater proportion of the initial population is seeded, thereby increasing the effect of the seeding on subsequent generations. The fact that the position of the shared elements is shifted across the array also contributes to the improvement achieved because the shared elements are not limited to the centre of the array in the best solutions [7].

The differences between the various cases are small when the number of shared elements is greater than about 50% of the available elements. In this case, the performance of the GA itself ensures that good results are achieved largely irrespective of the seeding. The exceptions to this observation are the cases where individuals are initialised with the specified number of shared elements (Figs. 4(b) and 4(d)).

Fig. 7 shows the best positions of the elements within an aperture of 50 available elements from 51 runs of the algorithm with a population size of 5000. When the number of shared elements is low, the shared elements tend to cluster near the centre of the aperture as observed previously [7]. However, as the number of shared elements increases, the shared elements split into two groups, one near the centre of the aperture and another near the edges of the aperture. The seeding schemes in Figs 4(b) and 4(d) are based on the assumption that the shared elements are clustered together at the centre of the array. The initialisation thus causes the GA to focus the search in the wrong areas of the problem space, leading to suboptimum solutions as seen in Figs 5 and 6.

The solution to this difficulty is simply not to seed the initial GA population when the number of shared elements is more than 50% of the number of available elements. As outlined previously, all the initialisation schemes except those in Figs 4(b) and 4(d) give approximately equal results in this case, so seeding is not required.

Further information about the performance of each seeding scheme can be gleaned from Fig. 8, which shows the cumulative distribution functions (CDFs) of the results for each seeding scheme when a quarter of the available elements are shared.

The non-overlapping seeding with both single and multiple solutions (Figs 4(a) and 4(c)) show only small improvements to the unseeded case. This is because these seeding schemes do not consider shared elements.

By comparison, the seeding schemes with the specified number of shared elements (Figs 4(b) and 4(d)) produce significantly better results than the unseeded GA. This improvement is due to the exploitation of additional information about the problem (the specified number of shared elements). Also noticeable is the significantly smaller variation in the achieved SLL when the population is seeded with a number of solutions with the specified number of shared

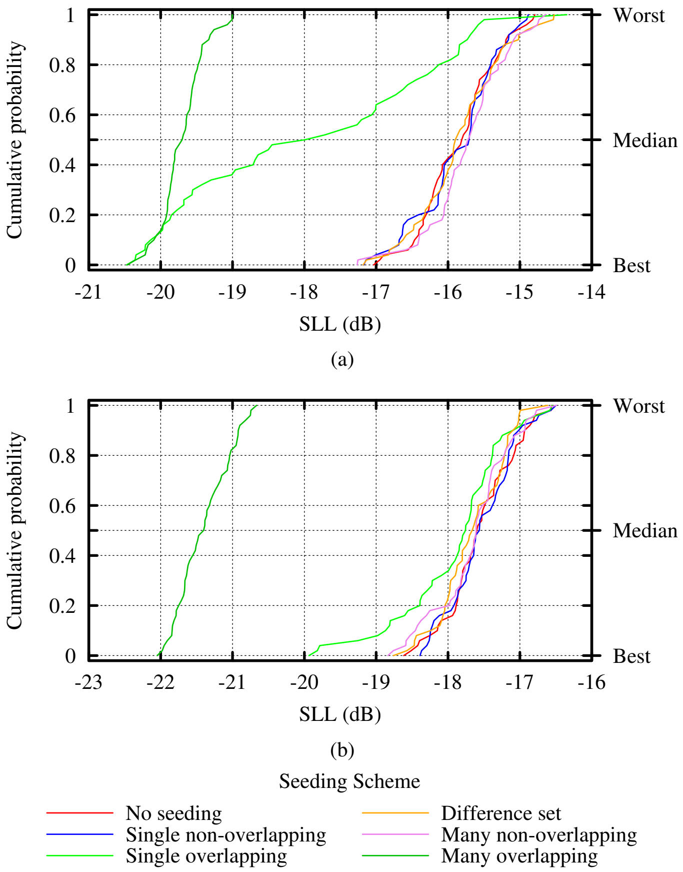


Fig. 8. The cumulative distributions when a quarter of the available elements are shared for arrays with (a) 100 available elements and (b) 200 available elements.

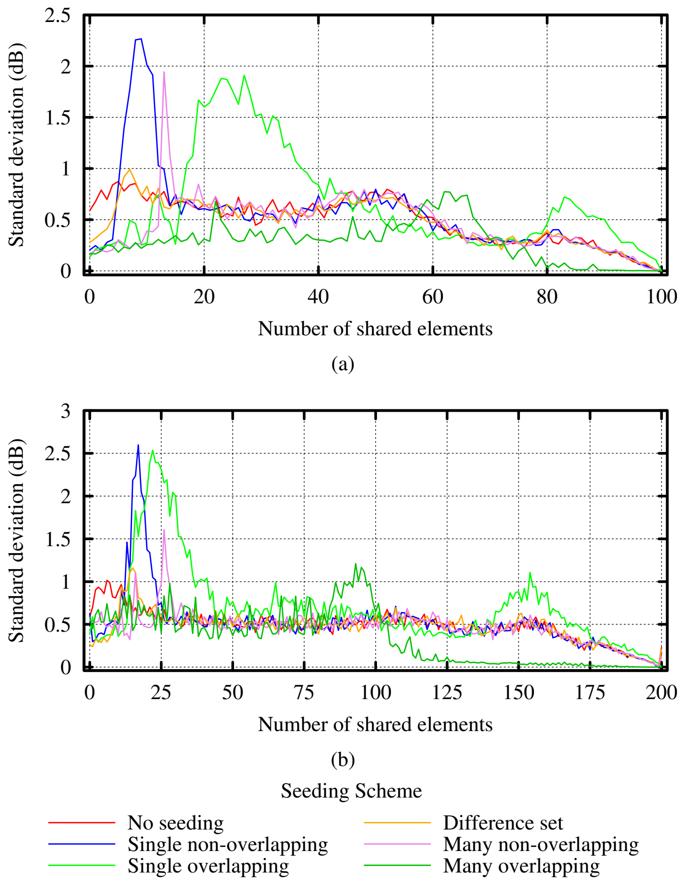


Fig. 9. The standard deviation of the results obtained for arrays with (a) 100 elements and (b) 200 elements.

elements (Fig. 4(d)).

The variation of the results achieved by the algorithms is explored in Fig. 9, where the standard deviation obtained for each algorithm over 51 runs is shown. In general, seeding the initial population leads to more consistent results.

However, some of the seeding schemes are seen to have a significant increase in their standard deviation for some numbers of shared elements. This is a result of the population being seeded with misleading solutions which cause the GA to focus its search on the incorrect portions of the solution space.

The improvement to the SLL as the number of generations increases when a quarter of the elements are shared is shown in Fig. 10 for the best solution obtained with each seeding scheme. Only 80 generations are shown as the only case where the solutions shown improve after 80 generations is the case where no initialisation is used. This case reaches its final value of -17.05 dB after 127 generations. The decision to run the GA for 150 generations was motivated by the fact that convergence takes place long before 150 generations, so this value allows the performance of the underlying algorithm to be demonstrated.

The most important observation from Fig. 10 is that the seeding schemes with the specified number of shared elements (Figs 4(b) and 4(d)) improve significantly faster than the other cases for 100 available elements. For 200 available elements, the seeding scheme in Fig. 4(b) improves at a rate comparable to that of the other cases, but continues to improve after the other cases have converged. As with 100 available elements, the seeding scheme in Fig. 4(d) improves substantially quicker than the other cases for 200 available elements.

TABLE I
TIME REQUIRED TO RUN THE PROPOSED ALGORITHM.

Seeding Scheme	100 elements	200 elements
No seeding	41.25 s	49.35 s
Difference set [15]	40.98 s	49.05 s
Single non-overlapping (Fig 4(a))	41.48 s	50.42 s
Single overlapping (Fig 4(b))	40.98 s	49.97 s
Many non-overlapping (Fig 4(c))	41.18 s	48.42 s
Many overlapping (Fig 4(d))	41.44 s	48.76 s

These convergence-rate improvements are achieved despite the initial solutions (generation 0) having comparable SLL values for all the cases considered. The improvement to the final results is thus not due to better starting SLL values, but rather due to the elements in the aperture having initial distributions which are closer to the optimum distributions.

The time required to run the algorithm is summarised in Table I, where the averages of thirty independent runs of the algorithm are provided. The algorithm was coded in MATLAB R2014a and was run on a computer with an Intel Xeon X5355 central processing unit (CPU) (quad-core running at 2.66 GHz), and 8 GB of random-access memory (RAM).

Table I shows that no significant additional run time is required by the proposed seeding schemes. This is anticipated as seeding is only performed once at the start of the algorithm and all the seeding schemes considered have simple implementations.

IV. Conclusion

A GA with suitable seeding of the initial population has been shown to produce excellent results far more reliably than the GA

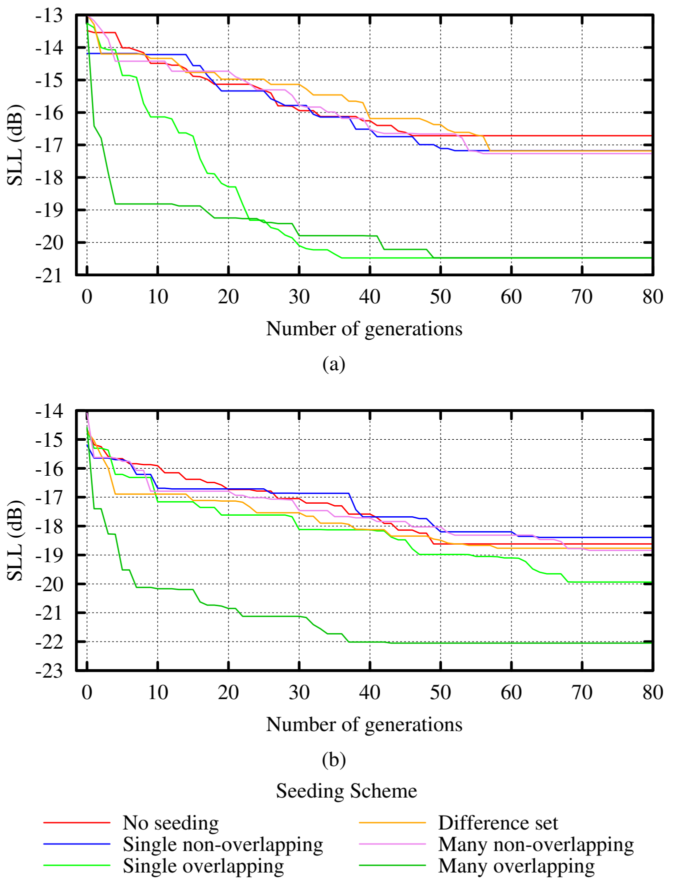


Fig. 10. The evolution of the SLL as the population ages for the best arrays when a quarter of the available elements are shared with (a) 100 elements and (b) 200 elements. The 100-element "no seeding" case is the only case which improves after 80 generations and reaches its final value of -17.05 dB after 127 generations.

alone and previously-proposed approaches to seeding the initial population, including those based on DSs and ADSs. Furthermore, the use of the proposed seeding schemes does not lead to any significant increase in the algorithm run time.

The GA used was conventional apart from the fact that it includes a penalty function. The penalty function was required to ensure that the specified number of shared elements is achieved.

A number of seeding schemes were evaluated, and the improvement in the final SLL results due to suitable seeding was clearly demonstrated. Even seeding just one individual of a large initial population can lead to significant improvements to the final results, though greater improvements were noted when a greater number of individuals were seeded. Seeding a number of individuals with the specified number of shared elements was shown to produce the best results. This seeding scheme was also shown to converge significantly faster than the other cases considered.

However, some of the seeding schemes lead to worse results than random initialisation under certain circumstances. This is a result of the seeding misleading the GA and causing the incorrect portions of the solution space to be explored.

Despite this limitation, the value of seeding is clearly demonstrated, especially when the number of shared elements is small (less than approximately 50% of the number of available elements). Remarkably, the difference between the best and worst results for 51 runs of the proposed algorithm differ by on the order of 1.5 dB and have a standard deviation of approximately 0.5 dB, so even a single run of the algorithm will produce a result which is close to the optimum.

Acknowledgement

The authors would like to thank the anonymous reviewers for their comments and suggestions which greatly enhanced this paper.

References

[1] B. Manz, "The coming age of multifunction systems," *J. of Electron. Defence*, vol. 36, no. 4, pp. 28–36, 2013.

[2] R. L. Haupt, "Interleaved thinned linear arrays," *IEEE Trans. Antennas Propag.*, vol. 53, no. 9, pp. 2858–2864, September 2005.

[3] C. I. Coman, I. E. Lager, and L. P. Ligthart, "The design of shared-aperture antennas consisting of differently sized elements," *IEEE Trans. Antennas Propag.*, vol. 54, no. 2, pp. 379–383, February 2006.

[4] I. E. Lager, C. Trampuz, M. Simeoni, and L. P. Ligthart, "Interleaved array antennas for FMCW radar applications," *IEEE Trans. Antennas Propag.*, vol. 57, no. 8, pp. 2486–2490, Aug. 2009.

[5] M. Simeoni, I. E. Lager, C. I. Coman, and A. G. Roederer, "Implementation of polarization agility in planar phased-array antennas by means of interleaved subarrays," *Radio Sci.*, vol. 44, 2009.

[6] R. Guinvarc'h and R. L. Haupt, "Dual polarization interleaved spiral antenna phased array with an octave bandwidth," *IEEE Trans. Antennas Propag.*, vol. 58, no. 2, pp. 397–403, Feb. 2010.

[7] W. P. du Plessis, C. Kitching, and A. bin Ghannam, "Element sharing in interleaved antenna arrays," in *Euro. Conf. on Antennas Propag. (EUCAP)*, March 2012, pp. 2230–2234.

[8] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propag.*, vol. 42, no. 7, pp. 993–999, July 1994.

[9] D. S. Weile and E. Michielssen, "Integer coded Pareto genetic algorithm design of constrained antenna arrays," *Electron. Lett.*, vol. 32, no. 19, pp. 1744–1745, 12 September 1996.

[10] S. Caorsi, A. Lommi, A. Massa, and M. Pastorino, "Peak sidelobe level reduction with a hybrid approach based on GAs and difference sets," *IEEE Trans. Antennas Propag.*, vol. 52, no. 4, pp. 1116–1121, April 2004.

[11] G. Oliveri and A. Massa, "Genetic algorithm (GA)-enhanced almost difference set (ADS)-based approach for array thinning," *IET Microw. Antennas Propag.*, vol. 5, no. 3, pp. 305–315, 21 February 2011.

[12] W. P. du Plessis, "Efficient synthesis of large-scale thinned arrays using a density-taper initialized genetic algorithm," in *Int. Conf. Electromagn. Adv. Appl. (ICEAA)*, September 2011, pp. 363–366.

[13] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[14] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evol. Comput.*, vol. 4, no. 4, pp. 361–394, 1996.

[15] G. Oliveri and A. Massa, "Fully interleaved linear arrays with predictable sidelobes based on almost difference sets," *IET Radar, Sonar Nav.*, vol. 4, no. 5, pp. 649–661, October 2010.

[16] D. Gordon. (2003, September) La Jolla cyclic difference set repository. Accessed 25 October 2012. [Online]. Available: http://www.ccrwest.org/diffsets/diff_sets/index.html

[17] (2011, May) ELEDIA cyclic set repository. Accessed 25 October 2012. [Online]. Available: http://www.ing.unitn.it/~eledia