Biosystems Engineering, Volume 88, Issue 4, August 2004, Pages 401-410.

Optimising Fungicide Applications on Winter Wheat using Genetic Algorithms

D.J. Parsons[1] & D. Te Beest[2]

[1] Silsoe Research Institute, Wrest Park, Silsoe, Bedford, MK45 4HS, United Kingdom;

email of corresponding author: david.parsons@bbsrc.ac.uk

[2] Wageningen University, Postbus 16, 6700AA, Wageningen, The Netherlands; e-mail:

dennis.tebeest@bbsrc.ac.uk

* NOTICE: this is the author's version of a work that was accepted for publication in Biosystems Engineering. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Biosystems Engineering, 88 (4) 2004, doi:10.1016/j.biosystemseng.2004.04.012

Abstract

A genetic algorithm is used in a decision support system to select the combinations of chemicals and the timing of successive treatments for the optimal control of fungal diseases in winter wheat crops, using a simulation model to predict the performance of different treatments. The search space is large and discrete, making the use of conventional optimisation methods impractical. Furthermore, the user requirements specify that the method must supply lists of near-optimal solutions, which fits with the use of populations of solutions in the genetic algorithm. Substantial improvements in the performance of the algorithm were obtained by tuning the fitness, selection, reproduction and replacement methods for the optimisation of short-term and long-term decisions. These also ensured rapid convergence in the former and prevented premature convergence in the latter.

The algorithm has proved to be effective at finding optimal and near optimal solutions within an acceptable time. When compared with exhaustive searches for cases where this is possible (short-term planning with restricted choices), it typically finds 5-8 of the top 10 plans and a similar number of the next 10. The results of the system in field and user trials have been good.

1. Introduction

In the UK, approximately two million hectares of land is used to grow wheat, and each crop receives on average 2.5 fungicide sprays (Thomas *et al.*, 1997). Hence, appropriate fungicide application is an important part of cereal husbandry. However there are around 20 appropriate chemical actives and at least one new chemical is introduced every year, replacing less effective old ones. These are formulated in combinations into hundreds of products. Each chemical has different levels of control against different diseases. Timing and dose both affect this degree of control (Paveley *et al.*, 2000). Chemicals can be combined together to improve results. Disease progress is very complex and variable from year to year, site to site and variety to variety. Surveys (Thomas *et al.*, 1997) have shown that, as a result, the use of fungicides by farmers is little affected by likely disease pressure or variety. Thus, some farmers are using inappropriate fungicide application. It would benefit both the environment and the farmer's profit if more appropriate timings and doses could be used.

Several European countries have made use of information technology to assist growers with disease control in wheat and other crops. One of the first systems was EPIPRE in the Netherlands (Zadoks, 1981; Rijsdijk, 1983). The name is derived from "epidemic prediction and prevention." The EPIPRE system used empirical models to relate observed disease levels to likely losses. Use of the system has now declined. Germany has a web-based advisory system called proPlant Expert (www.proplantexpert.com). It functions as an expert advisory system for a range of crops, pests and diseases. PC-Plant Protection, developed in Denmark (Murali *et al*. 1999) covers control of weeds, pests and diseases in wheat and other crops, including barley, peas and oilseed rape with an emphasis on reducing chemical use. Models are used to assess the need for control based on observations, then it gives recommendations of product choice and dose. Approximately 2500 licences have been sold.

Within the UK, Decision Support Systems for Arable Crops (DESSAC) was set up as a LINK project, that is, one with joint government and industry funding. The DESSAC system is designed as a personal computer-based platform to support the development of decision support systems, providing access to farm, field, pesticide and weather databases (Brooks, 1998). Wheat Disease Manager (WDM) is the first module and was developed in conjunction with the DESSAC platform. It is designed for use by farmers and their advisors to support decisions relating to the management of fungal disease on winter wheat. Its approach is somewhat different from the systems mentioned above, in that it uses semi-mechanistic crop and disease development models, and produces recommendations by optimising the control variables - chemicals, dose and timing - in the model.

There are four aspects to improving the application of fungicides to wheat crops: predicting the effect on the diseases; predicting the impact of reduced disease on increased yield; transferring this knowledge to the farmer or adviser; determining the optimum application of fungicides. This paper focuses on the fourth aspect within WDM. The models on which it is based are described elsewhere and cover simulation of the wheat canopy (Milne *et al*., 2003), yield formation, foliar disease and disease of the stem base and ear.

The WDM module contains four components: the user interface, the process model, the fungicide module and the decision model. The user interface allows the user to enter

observations of crop state and disease level into the farm database, and to set the requirements from the models. The process model is a collective term for the models outlined above, and receives information from the weather database, farm database (variety, sowing date & observations) and varieties database (disease resistance of wheat variety). The fungicide module handles the interactions with the database of information about chemicals, and validates spray programmes for compliance with constraints on spray timings and mixtures. It allows the user to choose the list of products to be considered by the decision model: typically a short list of 10-20 products preferred by the user. The decision model, which is the subject of this paper, suggests courses of action to the user by making use of all this information.

Winter wheat is attacked throughout its life by a number of fungal diseases of the stem, leaves and ear. The four major spring foliar diseases in the UK are *Septoria tritici*, yellow rust (*Puccinia striiformis*), brown rust (*Puccinia recondita*) and powdery mildew (*Erysiphe gramminis*). All of these diseases can cause substantial losses of grain yield if not properly controlled. Stem base diseases such as eyespot can also cause yield losses and ear diseases including *fusarium* and sooty moulds can reduce the value of the grain. The dispersal and development of these diseases are strongly influenced by weather variables, including temperature, humidity, wind speed and rainfall. The variables of importance differ from disease to disease.

The number of possible choices available to the farmer, even within a short time period, is very large: there are on the market over 200 fungicide products, containing between one and three active ingredients, with different levels of activity against each disease. Some of the products may also be mixed safely for application. Furthermore, there is growing interest in moving away from the standard doses towards adjusting the dose according to the disease risk. Thus, even if the choices are restricted to, for example, 20 well-known products, the number of possible two-spray programmes allowing one or two products per application over a four week period is almost 1,000,000 (20 single products and 380 mixtures on each date, giving 400^2 combinations, and 6 possible timing combinations with a minimum of 2 weeks between applications, as described in Section 2.2.2).

2. Decision model

2.1. *Requirements*

The objective of the decision model is to maximise the margin over fungicide costs, that is the value of the grain produced less the cost of the chemicals and their application. This will be referred to simply as the *margin*. This was a requirement from the users, who wanted the optimisation to be in terms of a financial objective, not yield maximisation or disease minimisation.

The decision model interacts with the process model through a collection of control variables. These are fairly complex, consisting of time of application, in discrete steps, dose, which could be continuous, and chemicals, which may include mixtures. There is also a requirement from the users that the decision model should not simply provide a single optimum solution, but should give a ranked list of near-optimal spray programmes.

Taken together, these considerations motivated the choice of a genetic algorithm. Firstly, it is capable of being used with a black box simulation model. Secondly, it is intrinsically suited to discrete variables, whether numerical or combinatorial. Thirdly, the existence of a population can be used to generate a list of near-optimal solutions in a very natural way. A previous application to a silage harvesting problem with somewhat similar structure had shown that genetic algorithms were suited to problems of operation scheduling (Parsons, 1998). The details of the operation of genetic algorithms can be found in reference works such as Holland (1992), Goldberg (1989) and Davis (1991).

The decision model is required for strategic or long-term planning prior to the fungicide treatment period (the ‘spraying season’), which is taken to be 15 March to 15 July, and for tactical or short-term decision making during the season. The different requirements for these two uses result in differences in implementation. The intention is that the user should create and save one or more long-term plans, then use the short-term mode to revise one as the season progresses. Each combination of variety and sowing date within a farm will require a different plan.

For long-term planning, the time scale covers the whole of the spraying season (15 March-15 July: 122 days). The user can choose to apply up to 4 sprays within this period, each of which can contain up to three products, subject to restrictions on permitted mixtures, timing and doses. The algorithm must search the space of possible spray plans, which includes all permitted mixtures, doses and timings, thoroughly; for long-term planning, speed of operation is a secondary consideration. The use of the long-term mode is not restricted to pre-season planning: if the user feels that it is necessary, it can be used at any point in the season to create a new plan, in which only the sprays after that date will be optimised.

For short-term decision making, the emphasis is on revisions to an existing plan, which may be required because the crop and disease have not developed as forecast at the start of the season. The system will have been updated with weather data and observations of the state of the crop and the amount of infection to allow more accurate predictions to be made. Applications of sprays prior to the decision point will also have been recorded. It is important for the system to operate quickly when used in this way. This is achieved primarily by restricting the planning horizon to 4 weeks. When combined with other constraints, this limits the number of possible spray applications to one or two within that period. As described below, this allows a more efficient encoding of the problem for use by the genetic algorithm. The other decision variables are the same as for the long-term optimisation.

In addition, there are several constraints on the spray plans. Firstly, there are restrictions imposed by pesticide approvals, such as the maximum total dose of a product applied during a season or the minimum interval between spraying and harvest. These have the force of law, and the program is not permitted to break them. Secondly, there are published approved mixtures of products. Although these are not part of the formal approvals, the program will not recommend mixtures that are not on the approved list, although the user can use the program to experiment with them. Finally, a decision was taken to impose a minimum interval between applications of 2 weeks, because this is approximately the interval before the emergence of the next unprotected leaf layer.

The algorithm also allows additional constraints to be imposed, such as fixing the date of one of the sprays, or fixing the contents of a spray. This could be used to answer questions such as

"What is the best product to spray today?" or "When would be the best time to spray product X?"

2.2. *Implementation*

The key features of the use of a genetic algorithm for any problem are the method of encoding the problem as a chromosome, the fitness function (related to the optimisation objective function), the method of selection of individuals for reproduction, the reproduction operators, and the method of replacement of old individuals by new ones.

The encoding was necessarily somewhat different for the short- and long-term optimisations, but initially they used similar fitness, selection, reproduction and replacement methods. These were chosen to give good performance in the short-term optimisation, but were found to be unsuitable for the long-term one. For the short-term it was necessary to introduce strong selection pressures to ensure that the optimum plan was found quickly, but to balance this with methods to preserve population diversity. However, these choices caused premature convergence to inferior solutions in the much larger long-term search space. A systematic investigation resulted in changes to most of the options. Substantial improvements were obtained, which are illustrated in the results section. This illustrates the importance of tailoring genetic algorithms to the problem being considered; they cannot simply be applied with the default settings.

2.2.1. *Population size and run length*

As the complexity of the problem varies dramatically, it is not possible to use a fixed population size and number of generations. After experimenting to find values that gave acceptable speed, reliability and diversity of the final population with different sizes of problems, heuristics were derived to adjust these according to the length of the chromosome, which is used as a measure of the complexity of the problem. The number of generations is proportional to the chromosome length, while the population size is proportional to its square:

$$\text{Generations} = 2\ \text{Length},$$

$$\text{Population} = \text{Length}^2/10.$$

2.2.2. *Encoding*

The way in which the problem variables are encoded as a chromosome can have a profound effect on the efficiency of a genetic algorithm. In general, the chromosome should be as short as possible and avoid redundancy. For the genetic operators to work well, short *schemata* within the chromosome should carry significant information, as it is these that tend to be passed from generation to generation (Holland, 1992, chapter 4). As far as possible constraints should be encoded into the structure of the chromosome, rather than imposed after it is decoded.

A binary alphabet is used for the chromosome, that is it consists of a string of 0s and 1s. It is convenient to consider this as divided into several genes, each determining a feature of the spray plan:

(1) date(s) of spray(s) as relative week numbers;
(2) number of chemicals in each spray;
(3) choice of each chemical; and
(4) dose of each chemical in steps of 1/4 dose.

All except the date genes are identical in both the short-term and long-term optimisations. These are explained in more detail below, but a simple example will help to illustrate the construction of the chromosome. Consider the case where a short-term, one-spray programme with up to two products being mixed from a list of 16 possibilities is being optimised. A programme containing a mixture of a half dose of product number 9 and a full dose of product number 2 applied 2 weeks from the current date would encode as

01,11,1001,0010,1,010,

where the commas indicate the divisions between the genes. Taking the genes in that order, 01 means a dose of (1+1)/4, 11 means a dose of (3+1)/4, 1001 is product 9, 0010 is product 2, 1 means 1+1 products and 010 means week 2. Note that the doses and the number of products cannot be 0, so 1 is added to the value of the gene interpreted as a binary number.

A temporal resolution of 1 week is used for the dates, that is, the date of a spray is simply the number of weeks from the start of the decision period. Given the inherent uncertainties and the fact that it may be impossible to spray on the optimum date due to weather or logistics, a finer resolution is inappropriate. It also helps to reduce the size of the search space, so reducing the time required to find a solution.

In short-term mode, this means that there are five possible one-spray timings and six possible two-spray timings within the 4 week horizon, given the minimum interval of 2 weeks between sprays. There is no simple, efficient representation for both one-spray and two-spray plans, so they are considered separately and the results are combined after optimisation. In each case the date or dates can be represented by a gene of three bits. For one spray this is treated as an integer representation of the week number relative to 'today', as in the example above. For two sprays it is mapped to a pair of week numbers using the representation shown in Table 1. Clearly, in both cases, there are a few unused bit combinations. These are treated as infeasible values.

In the long-term optimisation it is necessary to cover a much greater span of dates, and the user can specify consideration of up to four sprays, so enumerating the combinations as above would be much more complex. If only one spray is being optimised, the encoding is essentially the same as for the short-term case: a binary integer representing the week as an offset from 'today' or the start of the spraying season, whichever is later. The gene is increased to 4 bits to give it a range of 16 weeks (112 days). This is slightly shorter than the spraying season (122 days), but a spray programme containing a single spray in the last 10 days is highly unlikely to be effective.

When considering more sprays, it would be possible to give each a gene representing its date relative to the starting point, but this would require multiple constraints to enforce the correct sequence and spraying intervals. Parsons (1998) found that an efficient encoding of dates for operation sequences was to use genes that represent the *intervals* between the operations. That is, the date gene for the second spray encodes the number of weeks after the first spray that it should be applied, and so on. Again, the genes are treated as integers, but an offset of 2 weeks is added to each to enforce the constraint on the minimum interval between sprays. When there is more than one spray, each date gene is 3 bits, so the range spanned by two sprays is again16 weeks (7 + 2 + 7). If 4 bits were used, as for the single spray, most of the search space would be infeasible, or extend well outside the spraying season. It is not possible to reduce this further when there are three or four sprays without excluding some reasonable solutions. Inevitably, chromosomes arise that place the later sprays beyond the end of the spraying season. Rather than rejecting these, the late sprays are simply ignored, so that they become plans with fewer, widely-

spaced sprays. If they result in high margins, these will survive in the population, so the user's setting is effectively the maximum number of sprays to apply.

The remainder of the chromosome consists of sets of genes representing the number and selection of chemicals in each spray and the corresponding doses. The user may choose to mix up to three chemicals in each spray, so up to 2 bits are required to encode this as a binary integer. If the number of chemicals is fixed at 1, this gene is omitted. Each spray is then represented by up to 3 gene pairs for chemical and dose. The chemical selection is represented by a binary integer, which simply indexes the chemical within the list of those available: either the full database or a subset chosen by the user. The length of the gene depends on the length of the list, and will contain unused values if this is not a power of 2. The dose of each chemical is represented by 2 bits, which are mapped to ¼, ½, ¾ or 1 full dose. If the user wishes to impose constraints, such as fixing the date of application or the chemical applied, this is easily accommodated by removing the corresponding gene.

As noted above, some chromosomes will not represent valid spray programmes. In each case it would be possible to convert these to valid spray plans, but this would introduce a bias, so instead they are treated as infeasible, that is, not in the permitted solution space of the optimisation. Similarly, plans may be infeasible because they violate practical constraints, such as dose-time approvals or approved mixtures, which are built into the pesticides database in the system. All infeasible spray plans are given a value of 80% of the margin for an unsprayed crop. This ensures that they will not survive in the population, but does not bias the distribution of objective values to the extent that using an extreme value, such as 0, would. This is significant with some types of fitness function.

2.2.3. *Fitness normalisation*

The objective function - the margin, as defined in Section 2.1- has to be converted to a fitness value that will directly control the probability of selection of individuals for reproduction. Various types of linear and nonlinear mapping are possible, and the most appropriate for a given problem depends on the distribution of values of the objective function both in the early stages and when close to convergence.

For the short-term optimisation, rank-based fitness scaling was found to work well. This applies a linear transformation to the rank of each individual within the population, so that the best has fitness 1.0 and the worst 0.5. This imposes a high selection pressure even in the late stages of optimisation, because it means that individuals in the middle of the ranking have a fitness around 0.75 even if their objective function value is close to the maximum. This can lead to premature convergence, but, by selection of suitable population sizes and number of generations, produced quick and reliable performance.

The long-term optimisation uses a linear scaling of the objective function values so that the best has fitness 1.0 and the worst 0.5. Although this appears very similar to the above, it differs significantly in a population where there is a cluster of members close to the maximum with others scattered at lower values, because all those close to the maximum will have fitness values close to 1.0. In this case, rank-based selection is biassed towards the better members of the cluster, which can cause premature convergence, whereas linear scaling gives them almost equal probabilities of selection.

2.2.4. *Selection*

In each generation, a proportion of the population is selected for reproduction using a randomised procedure ('roulette wheel selection') in which the probability of an individual being selected is proportional to its fitness. The proportion used is 0.15 for the short-term and 0.8 for the long term. Partial replacement was found to give better convergence and eliminate infeasible plans more rapidly than replacement of the whole population at each generation. However, the improvement over replacement of the whole population in each generation was marginal in the long-term case, and further reducing the proportion replaced did not improve the results.

2.2.5. *Reproduction*

Reproduction consists of two operations: crossover and mutation. Crossover takes two individuals as parents and produces pairs of offspring by selecting one or more points in the chromosome at random and exchanging the strings for the two parents around these points. The short-term algorithm uses one-point crossover, whereas the long-term uses three crossover points.

Mutation randomly changes the value of one or more bits in the string, with an expectation of 1 bit per chromosome in short-term and 2 in long-term. Mutation encourages searching subspaces of the problem space that are not spanned by the original population.

2.2.6. *Replacement*

After reproduction new individuals replace old ones. Initially this used a rank-based replacement method similar to that employed in GENITOR (Whitley, 1989), in which new individuals are allowed into the population if they have higher fitness than the worst existing member. This gave very rapid convergence and sometimes resulted in a final population of the short-term optimisation containing only one or two distinct plans, thus failing to meet the users' requirement for a set of options. In order to maintain diversity, alternative replacement methods in which new individuals only have to compete with sub-populations were used. This is known as *tournament replacement* and is applied in both versions, with slight differences.

In the long-term optimisation, *parental replacement* is used: the tournament consists only of the new individual and one of the parent chromosomes that generated it. The parent is replaced if the new individual has higher fitness. However, the algorithm is constructed with a small probability of allowing the inferior of the two individuals to survive. This probability decreases as the algorithm proceeds in a process known as *annealing*. In the short-term version the tournament contains nine other individuals selected to have the most similar chromosomes to the new one. The member of the tournament with the lowest fitness is removed. This type of tournament is known as *crowding replacement*, because it tries to prevent the formation of crowds of individuals with very similar genomes.

In both cases, the best member of the old generation is always allowed to survive. This is known as *elitism* and ensures that the maximum fitness never decreases.

2.3. *Post-processing*

The two versions of the genetic algorithm described above were successful at producing diverse populations of plans, but the upper ranks of the population were often still dominated by

plans containing only a few cost-effective products with different timings, doses and mixtures. The users preferred to see a wider range of products used, so a post processing step was added to reduce the repetition of products in the population. The following rule is applied to the list displayed to the user, working from the top down:

> A plan is displayed provided that it contains at least one product that has not appeared in two previous plans.

This succeeds in producing a more acceptable list, because there is sufficient diversity in the population to ensure that plans with more than a few products have survived. Of course, some products may still appear frequently, if they do so in combination with a variety of others. Throughout the following section, which is concerned with the performance of the algorithm, the results considered are the full populations produced before application of the post-processing step.

3. Results

3.1. *Short-term optimisation*

The initial trials were conducted with the sprays restricted to single components. This allowed comparisons to be made with the results of exhaustive searches, which became impractical when complexity of the problem was increased. Several scenarios were investigated, including early spring and late spring decisions. In the early spring scenario, a fixed spray in the late spring was sometimes included. Similarly, in the late spring scenario an early fixed spray was sometimes included.

Agronomically, the three main spray timings are often referred to as T1, T2 and T3. The earliest, T1, is around growth stage 31-33 (first node detectable to third node detectable, or approximately leaf 3 emergence), and is significant in the control of stem-base diseases, such as eyespot, as well as foliar diseases. The next, T2, is around growth stage 39 (flag leaf emergence) and is very important because the flag leaf is normally the largest contributor to yield. Finally, T3, comes around growth stage 59 (inflorescence emergence to anthesis) and may contribute to preserving the photosynthetic area and control of ear diseases. The T2 spray is rarely omitted; a

two-spray programme will normally contain T1 and T2 or T2 and T3 depending on circumstances.

These tests gave consistently good results with the true optimum (found by exhaustive search) being achieved in almost every case, together with a collection of near-optimal solutions. Even where the optimum was not found, the difference in margin was much smaller than that arising from the natural variability in the system. *Figure 1* shows a typical result for an early season scenario (around T1). The margins for the 24 best one-spray and two-spray plans are ranked. In this case, both sets show good diversity and the two-spray plans, corresponding to T1 and T2, are clearly superior. In this scenario the levels of disease arising from the combination of variety and weather were too high to be controlled effectively by a single spray.

Figure 2 compares the results for one spray with an exhaustive search. It can be seen that the genetic algorithm has found 6 of the top 10 solutions, which is typical of its performance. Below this there is a diverse population covering the range of the best 100 plans from the exhaustive search. The algorithm thus meets both the requirements placed on it: optimisation and diversity.

3.2. *Long-term optimisation*

Owing to the size of the search space used in the long-term optimisation, it was not practical to test the solutions by exhaustive evaluation, so a different procedure was used to test the performance of the algorithm. The stochastic nature of genetic algorithms means that the final result may be sensitive to the sequence of numbers generated by the pseudo-random number generator (RNG), which can be changed by the value chosen to seed it. If the genetic algorithm is robust, there should be little variation in the best member of the final population, but a poor algorithm will result in wide variation. The reduction in variability as a measure of performance was used as the basis for testing.

Each configuration of the genetic algorithm was run with five randomly selected seed values and the results were recorded. The configurations could then be compared in terms of the maximum margin obtained for all five runs and the range of maxima. A good configuration would have a large maximum and a small range. Given the uncertainties present in the field, a range of a few pounds is not significant, but over £10 is a cause for concern. The results that

follow represent a scenario early in the spraying season, about T1, at which point the short-term optimisation can also be run for comparison. The tests were conducted with 5 different wheat varieties with different levels of resistance to the main foliar diseases, particularly yellow rust and *Septoria tritici*. Varieties with high resistance are likely to have less serious infections and so be less responsive to spraying. The varieties used were Brigadier and Riband (low resistance), Consort and Rialto (moderate resistance) and Claire (high resistance).

For the initial testing, to allow a closer comparison with the short-term optimisation, the length of the date gene was set to 2 bits for each spray in a multiple-spray programme and to 3 bits in a one-spray programme. The main settings that controlled the genetic algorithm, such as the selection, crossover and replacement methods were also the same as the ones that were known to work well in the short-term version. This is referred to in the tables as the 'original configuration'. It was found that the short-term optimisation (Table 2) was actually producing higher mean values than the long-term (Table 3). This undesirable result was contrary to expectations, because the search space for the short-term version is a subspace of that for long-term, so the latter should be able to find every solution found by the former. The long-term algorithm also showed a very large range, especially on the low-resistance varieties. Both of these were indicative of defects in the configuration of the algorithm.

Systematic testing of options within the genetic algorithm resulted in the configuration described in Section 2.2. This produced substantial improvements: the maximum and average margins now always exceeded those found by the short-term algorithm, as they should. The range of maxima between runs had also been reduced dramatically, showing that the performance was much more consistent (Table 4). When the date genes were restored to the length described in Section 2.2.2 (4 bits for single sprays and 3 bits per spray for multiple sprays), the margins improved substantially, as shown in Table 5, reflecting the fact that the search space included the full spraying season. The range of variation between runs remained small, as required. An inevitable consequence of these changes is that the long-term optimisation is slower than the short-term, by about a factor of 10. This is acceptable, given that it will be used occasionally for planning, whereas short-term optimisation is intended for tactical management decisions within the season.

Over the full range of 30 scenarios considered in this way, the changes described above improved the maximum margin by over 10% in 11 of them, and over 20% in five. These results illustrate the importance for effective operation of a genetic algorithm of tuning the settings carefully and choosing the right encoding. In this case, a partially redundant encoding (which generates some sprays beyond the end of the season) is preferable to a more compact one that fails to span the problem space.

3. Discussion

As the results show, the genetic algorithm used is successful in giving solutions to this practical problem. For short term optimisation with single-component sprays, for which an objective comparison is available, the performance is very good and the diversity of choices in the final population meets the user requirements.

A genetic algorithm is well-suited to this problem, because of its ability to handle different types of variables and produce a set of solutions. For use in real-time, it also has the desirable 'anytime' property: the user can observe the objective function value and interrupt when the result is felt to be good enough for practical purposes.

The plant-disease system is subject to large uncertainties: those intrinsic in biological systems and the effects of future weather. These mean that absolute precision in locating the optimum is less important than locating alternatives that can be compared for their robustness across a several scenarios for future events. The DESSAC system provides multiple weather sets for each site, allowing such assessments to be made. At present this has to be done manually, by selecting each set in turn, but this will be automated in future.

3.1. *Field trials*

During development the system underwent two years of replicated plot trials on eight sites across the UK. In the first year, it was found to give a satisfactory selection of chemicals on most sites, but tended to spray slightly late on many of them, which was not sufficiently risk-averse where disease pressures were high. This was due to the performance of the underlying process model in its first year of trials. Subsequent revisions have improved

performance in more recent trials. In the 2002 harvest season, trials intended to represent farm practice were conducted at six sites in the UK and Ireland, mainly on split fields, including two at one site (Paveley, N. personal communication). In comparison with the reference fields managed according to good agronomic practice, WDM achieved a higher yield on all fields (mean difference 0.5 t/ha) and a higher margin on five of the seven (mean difference £7/ha). Simultaneous trials in Denmark also gave good results, but are omitted from the means above because the protocols were different.

3.2. *User trials*

Potential users of the system have been involved throughout its design and development to ensure that it meets their needs. Early designs were based on the results of extensive user interviews, and these were then altered in response to user comment and tested again. This was repeated throughout the project. In parallel with the field trials there have been two years of usability trials of prototypes, involving over 60 farmers and consultants in the second season. In total, over 200 individuals have taken part in the evaluation of the software.

Users have sometimes been surprised by the suggestions provided by the optimisation, or perceived a preference for certain chemicals. However, the suggestions were rarely bettered by manual adjustments to the spray programme. This confirmed that the optimisation was accurately reflecting the process model and showed a difference between intuition and the results of the model. The one feature of the genetic algorithm itself that caused comment was the fact that the list of solutions could differ on successive identical runs due to its stochastic nature. To prevent this, the random number generator is now reseeded with a fixed value at the start of each optimisation.

Conclusions

The use of a genetic algorithm in a decision support system for the selection of fungicide spray plans in both the long-term strategic planning and short-term tactical revision modes of operation meets the requirement to produce a list of optimal and near-optimal programmes from which the user can select their preferred option. The performance of the algorithm has been demonstrated in systematic testing and the system has performed successfully in field trials.

In order to test the algorithm for optimisation problems that were too large for exhaustive search methods to provide a benchmark, a method based on using the variation between randomly seeded runs as a measure of robustness was developed.

It was a found that substantial changes in the configuration of the genetic algorithm were required to achieve good, robust performance in the long-term and short-term modes. This illustrates the need to tailor the algorithm to the problem and to test the options carefully.

Acknowledgements

The DESSAC project was supported by the Ministry of Agriculture Fisheries and Food (now Defra) through a number of programmes including LINK 'Technologies for Sustainable Systems' with joint funding from the Home-Grown Cereals Authority, the Biotechnology and Biological Sciences Research Council and Farmplan Computer Systems.

The genetic algorithm software used in this project is the SUGAL package written by Dr. Andrew Hunter at the University of Sunderland, England.

References

Brooks D H (1998). Decision support system for arable crops (DESSAC): an integrated approach to decision support. In: Pests and Disease, Proceedings of an International Conference, Brighton, UK, 16-19 November, 1998. BCPC, Bracknell, Volume 1, 239-246

Davis L (editor) (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York

Goldberg D E (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts

Holland J H (1992). Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, Massachusetts

Milne A E; Paveley N D; Audsley E; Livermore P (2003). A wheat canopy model for use in disease management decision support systems. Annals of Applied Biology, **143**, 265-274

Murali N S; Secher B J M; Rydahl P; Andreasen F M (1999). Application of information technology in plant protection in Denmark: from vision to reality. Computers and Electronics In Agriculture, **22**, 109-115

Parsons D J (1998). Optimising silage harvesting plans in a grass and grazing simulation using the revised simplex method and a genetic algorithm. Agricultural Systems, **56** (1), 29-44

Paveley N D; Lockley D; Vaughan TB; Thomas J; Schmidt K (2000). Predicting effective fungicide doses through observation of leaf emergence. Plant Pathology, **49** (6), 748-766

Rijsdijk F H (1983). The EPIPRE system. In: Decision Making in the Practice of Crop Protection. (Austin R B ed). Monograph 25, BCPC, Bracknell, 65-76

Thomas M R; Garthwaite D G; Banham A R (1997). Pesticide Usage Survey Report 141: Arable crops in Great Britain 1996. Defra Publications, London

Whitley L D (1989); The GENITOR algorithm and selective pressure: why rank-based allocation of reproductive trials is best. In: Proceedings of the Third International Conference on Genetic Algorithms (ICGA). (Schaffer J D ed). Morgan Kauffman, San Francisco, 116-121

Zadoks J C (1981). EPIPRE: a disease and pest management system for winter wheat developed in the Netherlands. EPPO Bulletin 11, 365-369

Table 1. Representation of date pairs as binary genes in short-term optimisation

Date pair, week numbers		*Representation*
0	2	0
	3	1
1	3	10
1	4	11
0	4	100
2	4	101

Table 2. Results from 5 runs of short-term optimisation with 2 sprays and mixtures up to 3 products

	Margin, £			
Variety	*Mean*	*Min*	*Max*	*Range*
Brigadier	421	418	424	6
Riband	425	424	426	2
Consort	431	430	432	2
Rialto	436	435	438	3
Claire	468	465	470	5

Table 3. Results from 5 runs of long-term optimisation with 2 sprays and mixtures up to 3 products using original settings and 2 bit date encoding

	Margin, £				*Difference from*
Variety	*Mean*	*Min*	*Max*	*Range*	*short-term mean*
Brigadier	411	399	418	19	-10
Riband	408	398	417	19	-7
Consort	420	415	422	7	-11
Rialto	429	423	437	14	-7
Claire	462	459	467	8	-6

Table 4. Results from 5 runs of long-term optimisation with 2 sprays and mixtures up to 3 products using revised settings and 2 bit date encoding

	Margin, £				*Difference from*	*Difference from*
Variety	*Mean*	*Min*	*Max*	*Range*	*short-term mean*	*mean in Table 3*
Brigadier	424	423	424	1	3	13
Riband	427	423	429	6	2	19
Consort	433	430	435	5	2	13
Rialto	440	438	441	3	4	11
Claire	468	68	468	0	0	6

Table 5. Results from 5 runs of long-term optimisation with 2 sprays and mixtures up to 3 products using revised settings and 3 bit date encoding

	Margin, £				*Difference from*
Variety	*Mean*	*Min*	*Max*	*Range*	*mean in Table 4*
Brigadier	514	512	516	4	90
Riband	509	508	511	3	82
Consort	518	516	519	3	85
Rialto	522	519	524	5	82
Claire	536	536	537	1	68

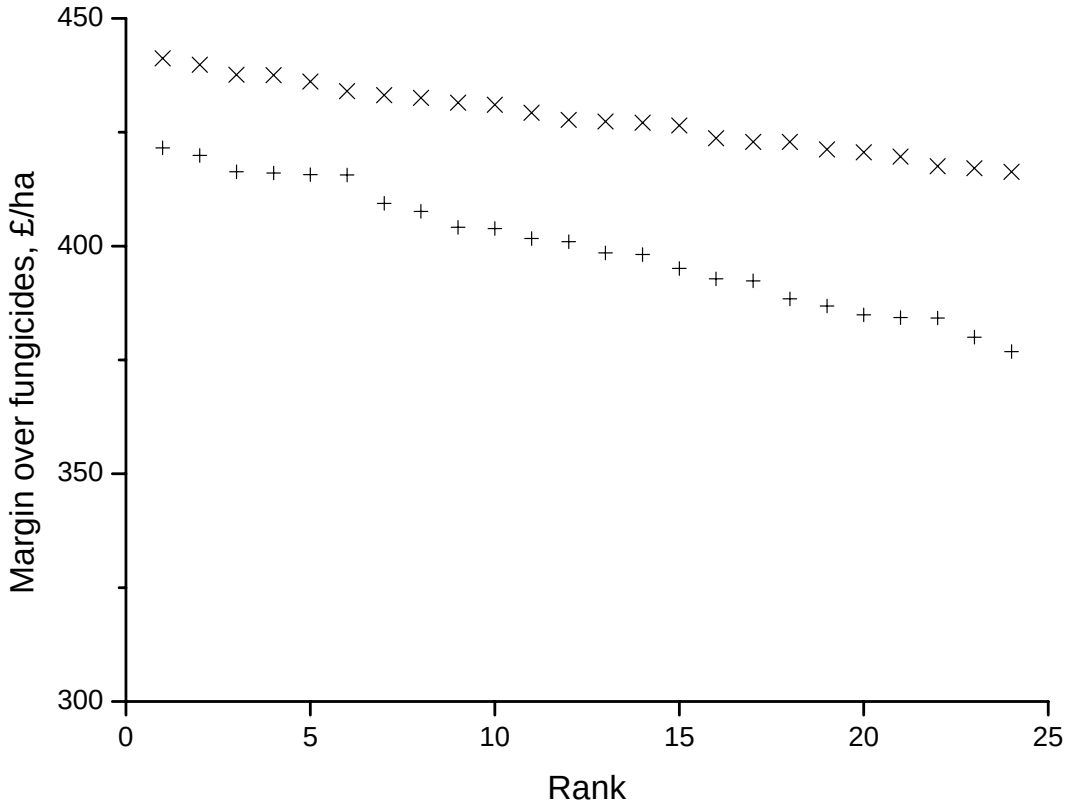

Fig 1. Comparison of best one-spray (+) *and two-spray* (x) *programs in the results of the genetic algorithm for an early season, short-term optimisation*

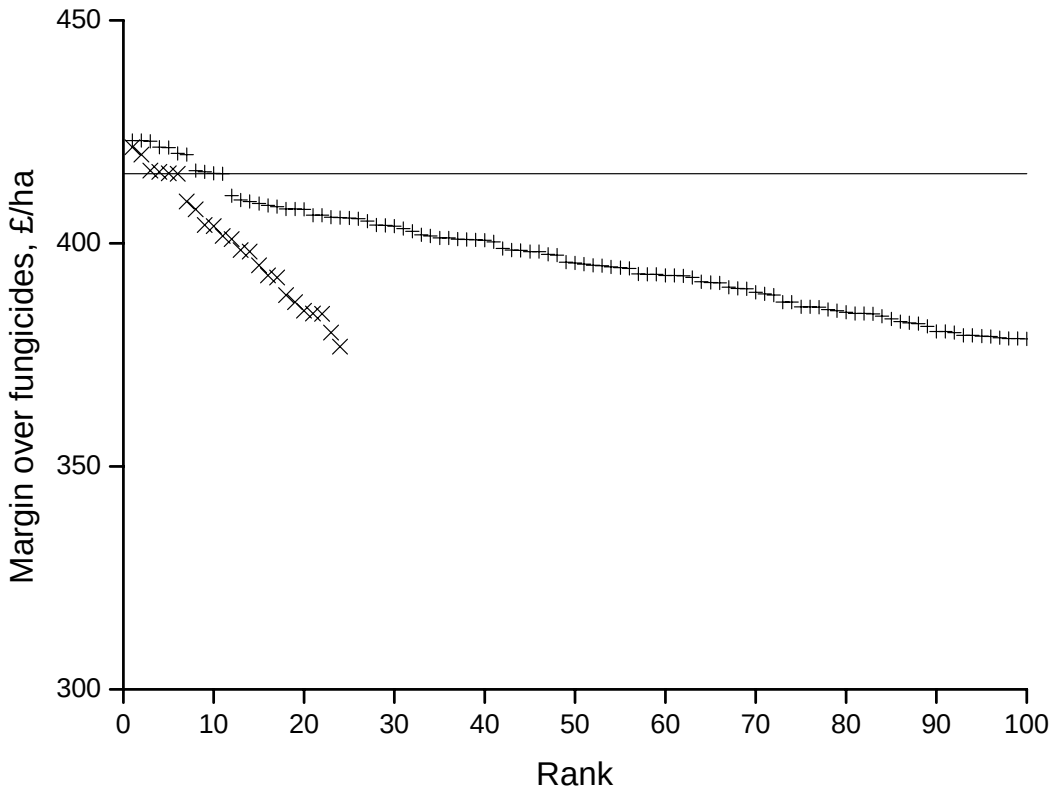

Fig 2. Comparison of best one-spray programs from exhaustive search (+) and genetic algorithm (x) for an early season, short-term optimisation. The horizontal line passes through the 11th best plan from the exhaustive search; six of the genetic algorithm solutions lie above this line.

https://dspace.org/

Optimising fungicide applications on Winter wheat using genetic algorithms

Parsons, David J.; Te, Beest D.

2004-08-01T00:00:00Z
Elsevier
http://dx.doi.org/doi:10.1016/j.biosystemseng.2004.04.012
Parsons DJ, Te Beest D. (2004) Optimising fungicide applications on Winter wheat using genetic algorithms. Biosystems Engineering, Volume 88, Issue 4, August 2004, pp. 401-410

Downloaded from a DSpace repository. https://dspace.org/