Monograph:

Shaw, K.J. and Fleming, P.J. (1996) Initial Study of Multi-Objective Genetic Algorithms for Scheduling the Production of Chilled Ready Meals. Research Report. ACSE Research Report 623 . Department of Automatic Control and Systems Engineering

Initial Study of Multi-Objective Genetic Algorithms for Scheduling the Production of Chilled Ready Meals

Katharine Jane Shaw and Peter. J. Fleming
Department of Automatic Control & Systems Engineering,
University of Sheffield,
Sheffield S1 4DU, UK.
Email: jane.shaw@sheffield.ac.uk Web: http://www.shef.ac.uk/~gaipp/

25 April 1996

Research Report 623

Abstract
Genetic algorithms are applied to a scheduling problem taken from the chilled ready meal industry. The application of a Pareto-optimal multi-objective genetic algorithm is demonstrated, together with its ability to produce a selection of near-optimal schedules according to the manufacturer's needs rather than a single solution.

Introduction

Using genetic algorithms (GAs) as an optimisation technique for solving production scheduling problems has gained popularity over the last ten years. Manufacturers demand techniques which allow them to timetable production of factory orders to varied constraints and objectives. In the ready meal industry, there is frequently no ideal method for this to the extent that some shop-floors may commonly timetable work by pencil and paper using rules of thumb. Alternative automated methods are too slow, inaccurate, difficult to implement or simply not reliable enough to provide schedules when needed in practice.

Following the lead of Davis, (1985), users of GA techniques have produced impressive theoretical methods for standard job-shop problems, researchers increasingly seek to apply them in real life; examples include brewery production (Starkweather, et al, 1993), and chemical plants (Bruns, 1993, and Cartwright and Tuson, 1994). The gap between theory and practical techniques that can be applied with confidence to the real-life factory floor still reveals some areas which need to be addressed to provide an effective scheduling system.

Our example is a ready meal manufacturer, who produces chilled ready meals to daily orders, where timing of each stage in the production is crucial. This is a novel application for GA scheduling and one that has highlighted several interesting areas in which use of GAs could be adapted to provide solutions to long-standing problems within this industry.

Applying GAs to scheduling problem

Previous work on GAs suggests that they have the potential to create a general - purpose, portable, accurate and fast scheduler for use in a real-life manufacturing environment. GAs are particularly suited to the problem because of their ability to search large, complicated and unpredictable search spaces. Scheduling problems often contain a mixture of functions, discrete or continuous, well or ill-defined. The parallel nature of GAs encourages further techniques such as multi-objective search, optimising several objectives in the same search and for dealing with the complex decision-making issues which frequently appear in scheduling problems. GAs are efficient for a problem that cannot be tackled in exponential time.

Description of a ready meal manufacturing problem

This work aims to differs from work on standard job-shop scheduling problems by studying the use of GAs on a difficult, real-life scheduling problem. Manufacturing data is provided by a local manufacturer of chilled ready meals for sale in supermarkets. The factory requirements indicate that some of the most difficult scheduling problems are to be found in the food and drink industry, (Shaw, 1996). The timescale is extremely short - orders come in each morning and have to be scheduled and manufactured ready for delivery by the end of the day. Cooking meals is extremely time-dependent and many variables involved are interdependent within the process.

In initial discussions with the factory, we identified the problem of the allocation of the orders to the production lines. This is one of the simpler problems, similar to a timetabling problem, as each of the 45 orders must be allocated between the 13 production lines according to a number of constraints such as machine limitations, packaging requirements, hygiene regulations, and due times, whilst allowing the final arrangement to limit several costs. Solving the allocation problem is not so difficult, but when working on a practical application, further complexities arise.

One particular requirement of this system is the need for an accurate rescheduling system, which must have two main functions - the ability to reschedule very quickly in the face of a change to the system, and to be able to change the priorities of the optimisation during manufacturing, to meet the altering demands of the customers throughout the day, allowing the flexibility to deal with commercial or political decisions rather than ones based on costs alone, where necessary. Using a multi-objective GA implementation may prove effective for working on this problem for both these requirements.

MOGAs for production scheduling

Any manufacturing problem involves a large number of defining variables. Constraints and objectives include ingredients or materials, machine capabilities, staffing requirements, product specifications, quality control, customer preferences, storage and delivery demands, many of which conflict or influence each other in indefinite ways. It would be a lengthy process to model all the constraints required in creating an automatic decision making process for scheduling and the process is still very reliant on human knowledge and experience. Although work in knowledge capture continues to improve this process, we have the task of building a scheduling system which uses enough constraints to work with some accuracy, whilst still allowing human influence when needed.

Scheduling with multiple objectives has often used one single function which aims to combine the costs in some meaningful way. This means predefining priorities and preferences, adjusting weights within the function to accurately reflect the decision making process, and even creating a search space which may not effectively represent the problem.

The multi-objective genetic algorithm presented by Fonseca and Fleming (1995) is attractive as it avoids this problem. The technique uses Pareto optimisation to search from a selection of possible solutions rather than working in a single search direction. This allows us to optimise schedules according to multiple criteria, and allows a (human) decision maker to pick out their preferred solution from an offered selection of solutions which are all optimal, and singly biased towards certain objectives. The ability to change preferences of the schedule as the system is running may be vital to reaching required targets.

For example, schedules could simultaneously be produced which are optimal to satisfying different customers' demands, and the production manager would be able to decide which schedule to use in order to give the most valued (or most demanding!) customer preferential treatment. Similarly, when optimising a schedule to several conflicting aims such as cost, time and quality, a manager could decide to use a schedule which impaired the time taken in order to give preference to quality in products, or vice versa. No single management policy need be taken before the scheduler starts working, but decision can be made dynamically as the need occurs on the shop floor.

Objectives for this problem

The problem is based on the description of the manufacturing process above. With thirteen lines, 45 products and many constraints on the assignments of jobs to lines, we aim to minimise three objectives simultaneously.

Cost 1 - Rejected Orders

Cost 1 measures the number of orders rejected by the shop floor as being infeasible for manufacturing within the schedule. This can be due to lack of compatibility with other jobs, time constraints, or possibly due to a decision made by the human scheduler, for example, in the case of a particular order being contaminated or not worth manufacturing within that day's run. However, this would be a rare occasion and we aim for a cost 1 value of zero.

Cost 2 - Lateness of batches within orders

The cost here is more complicated, and can be altered using weights to change the importance of each of the seven depots for which orders are destined. Every product has to be supplied to each of the seven depots, providing a function which evaluates the total cost of the lateness of any product being supplied to any depot. There is also a set of times at which each delivery to each depot leaves the factory, staged throughout the afternoon, and these must be taken into account within the schedule. For example, regional demands from Depot A may mean that it is more important to supply to this depot rather than Depot B, which might have a more efficient distribution system. Orders being delivered late to Depot A would incur a greater cost than those same orders being delivered late to Depot B.

Cost 3 - Variation in shifts and staffing balance

A further cost aims to try and ensure that shifts end at roughly the same time of day. This objective is for practical reasons, so that all staff work roughly the same hours, and so that the work load is balanced evenly across the various lines where possible. Certain products have a choice of lines and without this measure, it is possible that a large number of products might be made on the same line.

Implementation of the MOGA for the scheduling problem

The implementation of the MOGA scheme applied to the problem is based on Pareto-optimality, as suggested by Fonseca and Fleming, (1995). When presented with a population, it evaluates each cost separately for each individual, and picks out the 'non-dominated' solutions - solutions such that the set of costs found each cannot have any cost contained within them improved without adversely affecting another cost - as possible solutions for that population. This provides us with a set of possible solutions, rather than a single one as a genetic algorithm relying on a single objective given by weighted sum of the costs might do. The set of solutions can be presented to a decision maker to choose which one is given preference, and the schedule that this solution provides may be used as a practical solution. We can also set automated priorities within the implementation so that one cost is favoured above others, although here they are given equal priority, and allowing to select the final choice of solution.

Initial runs of the MOGA were compared against a standard single-objective, weighted sum genetic algorithm, in which all three costs combine into a linear combination, giving a single value for minimisation. Both methods were implemented using combinatorial chromosomes to represent the permutations of the incoming orders to the shop floor allocation process, and a separate translation stage provided the information from the shop floor to create a potential working schedule and evaluate each of the three costs for the individual. This representation was preferred rather than a knowledge-inclusive representation so as to allow flexible use of the problem data. The MOGA was further tested with and without mating restrictions and sharing, (referred to as MOGA+ and MOGA- respectively). The inclusion of mating restrictions and sharing aims to prevent the multi-objective search becoming too concentrated in one area of the objective trade-off surface, further discussed in Fonseca and Fleming, (1995). Each method was run 6 times as a generational GA, using 50 individuals for 40 generations. Preliminary tests had shown that the edge recombination operator and splice mutation operator were effective for this problem, and these were used. The initial results of the runs are presented below.

Comparison of the schemes

Using Pareto optimality, we consider the sets of the best non-dominated solutions found by each method over all the runs. In particular, we examine the trade-off surface provided by trying to optimise cost 2 and cost 3 simultaneously. All solutions found by all methods had minimised cost 1 to one of two values (0 or 1), and so this simplified the analysis of the results to studying the Pareto optimal solutions found for cost 2 and cost 3, whilst referring to their values for cost 1. Histograms of the distributions of all three costs found by each solution are presented, together with plots of the non-dominated points found on the solution trade-off axes..

Initial Results of the Application of MOGAs to the scheduling problem

Results of Comparisons of Schemes

We examine all non-dominated (ND) solutions found by each run of each method. The number of actual solutions found varies between methods depending on their performance within the multi-objective search space.

Method	Number of ND solutions found	Number of zero cost 1 solutions found
MOGA +	20	8
MOGA -	18	9
WS	6	0

Despite each method running for the same amount of evaluations, the weighted sum presents only one non-dominated solution for each of its runs - that is, one 'absolute' solution rather than a selection, suggesting that it is searching in a more linear manner than the other two. As the single objective function of the weighted sum consists of a linear combination, we may expect this.

It is also interesting to note that although all methods managed to minimise cost 1 to only one of two values (0 or 1), the weighted sum method failed to find any solutions in which cost 1 was zero. In practice this means that it was finding schedules in which an order had been rejected in order to minimise the other costs more effectively. This may indicate that the weighted sum method may over-emphasise the trade-off between the costs.

Qualitative comparison of non-dominated solutions for each MOGA method

Figures 1 and 2 show the non-dominated points found by each method for all the runs. Concentrating on the area closest to the origin, we are looking for points which are not dominated by others as they approach the optimum for both costs. It can be seen that the optimal solutions regarding costs 2 and costs 3, found by one WS solution ('*')and one MOGA- ('x'), followed by several MOGA+ ('+') solutions, are in fact non-zero for cost 1. As explained above, and seen clearly in the 3-D plot, none of the weighted sum solutions ('*') found a zero cost 1. The best zero cost 1 solution, and indeed subsequent ones, are found by MOGA- and MOGA+.

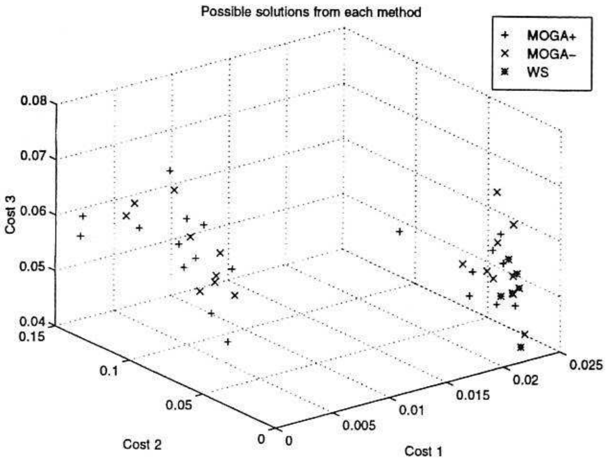

Figure 1 - plot of all non-dominated solution found by all runs of each method

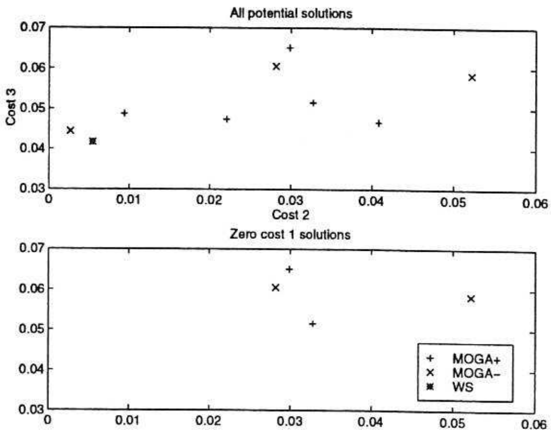

Figure 2 - solutions divided between zero and non-zero values of cost 1

The division between these costs is illustrated in the three dimensional plot of Figure 1; the group of solutions closest to the origin are those where cost 1 is zero and it can perhaps be seen more clearly that both MOGA schemes provide the best solutions in this case.

'Typical' schedules for each method

Schedules found by each method are presented as an example of the output of these techniques. The runs used were picked simply as the first ones of each batch for each method. Schedules are presented as Gantt charts indicating the allocation of orders (blocks numbered 1-45) to the production lines (y-axis, 1-13), with times of day from 06.00 to 06.00 the following day on the x-axis. The first deadline of 14.00 is marked by a vertical line.

As explained above, the MOGA techniques present a selection of possible solutions for the scheduler to make the final decision. Figure 3 shows possible solutions offered by each method on the trade-off axes for cost 2 and cost 3, with the lower half of the graph showing only those same solutions with a zero cost 1.

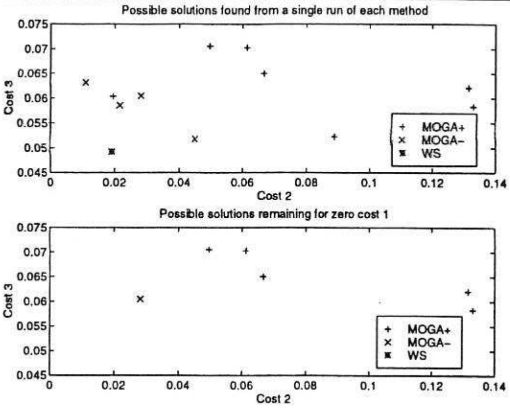

Figure 3 - Graph showing 'possible' solutions offered by a single run of each method.

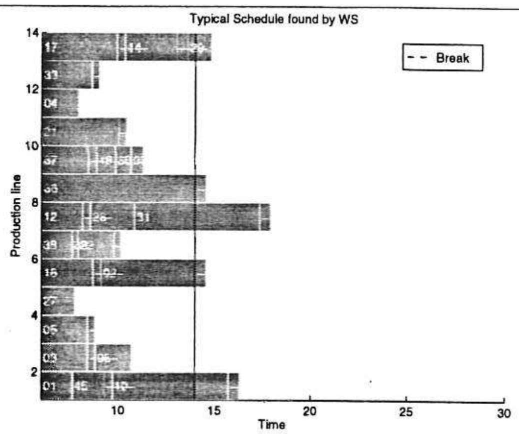

Figure 4 - Optimal schedule found by weighted sum from one run

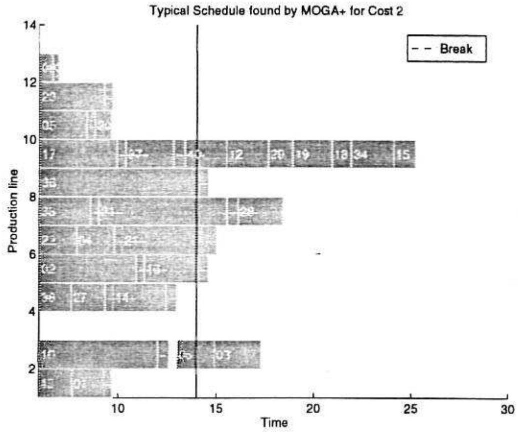

Figure 5 - One of the possible optimal schedules found by one run of MOGA+ , favouring cost 2

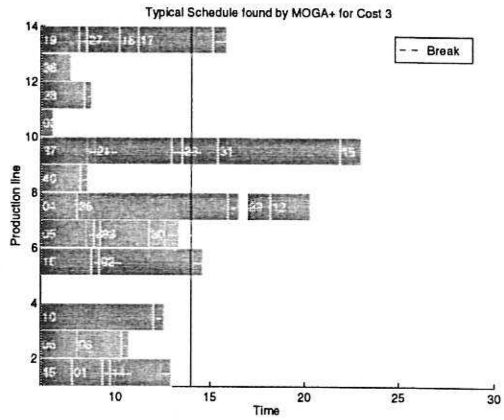

Figure 6 - One of the possible optimal schedules found by one run of MOGA+, favouring cost 3

Figure 4 shows the one schedule provided as a solution by the weighted sum. We must remember that although the late orders (indicated by blocks crossing the line) are infrequent and lines are reasonably balanced, this has been done at the expense of completely rejecting some of the orders from the scheduling process. However, the costs 2 and 3 have been reasonably satisfied by this compromise, although this may be unsatisfactory to the manufacturer.

Figure 5 shows a schedule picked from the selection of possible MOGA+ solutions to give a low value for cost 2 (late orders). We must remember that although orders do appear beyond the 'time limit' line, this may be necessary due to line restrictions and the weights put on the different depots - it may be that it is cheaper to let these products run late than to make them sooner at the expense of other orders.

Figure 6 shows a schedule picked from the selection of possible MOGA+ solutions to give a low value for cost 3. Orders appear to be slightly later but the schedule has tried to balance this out more than previously. We can see that a production manager could pick one of these or the other three found by MOGA+ according to suit the extra constraints which may only be evaluated by his or her own expertise.

Clock Time

These methods were all run on a SPARC station, using MATLAB version 4.2. All take just under an hour to run, meeting the manufacturer's target of having a method which takes less than one hour to provide a schedule.

Discussion

Whilst the comparison methods are simplistic at this stage, it is hoped that the multi-objective genetic algorithm techniques are demonstrated on a real-life scheduling problem and give adequately flexible and fast solutions to satisfy even the severe demands of a manufacturing environment such as a chilled ready meals factory. Scheduling can take place within the hour and even at the end of that hour, the scheduler is able to choose from a set of possible near-optimal solutions rather than be presented with one single answer, which may not ultimately suit the user. The MOGA has been shown to find good solutions when minimising three costs simultaneously compared with the weighted sum.

Whilst the weighted sum can still provide good solutions for costs 2 and 3, it seems to be at the expense of minimising cost 1. It could be that the weights for these costs are not suitable adjusted. The linear nature of the objective function may also prevents it effectively searching the extreme parts of the trade-off surface as the multi-objective methods can. Finally it is disadvantaged in its rigid presentation of only one solution at the end of its run.

The MOGA offers an effective method in comparison to the weighted sum regarding flexibility and solution quality. It is hoped that extensive statistical comparison such as those suggested by Fonseca and Fleming, (1996), may provide quantitative evaluations of these methods in the future.

Conclusion

Multi-objective GAs in production scheduling will allow a more accurate optimisation process which better reflects the state of the factory without the need to calculate weights for single objectives. Exploration of parameter values may improve further the already promising performance of these techniques. An important product of the use of MOGAs for scheduling is having a method which provides a set of possible solutions rather than a single answer can allow human influence in the final decision, giving the chance to include immediate subtle decision factors in the choice.

Combining the search power of genetic algorithms with the flexibility of multiple solutions presents exciting possibilities for a powerful and realistic scheduling technique to use in ready meal manufacturing.

Acknowledgements

Thanks to Pennine Foods for collaboration in this project & Dr. Carlos Fonseca for his useful advice on this work. Also thanks to EPSRC for funding the Ph.D. research studentship of the first author.

References

Bruns, R., (1993). Direct Chromosome Representation and Advanced Genetic Operators for Production Scheduling, *Proceedings of Fifth International Conference on Genetic Algorithms,* Morgan Kaufmann Publishers.

Cartwright, H.M., and Tuson, A., (1994). Genetic Algorithms and Flow shop scheduling: towards the development of a real-time process control system, *Proceedings of the AISB Workshop on Evolutionary Computing, Leeds University, 1994, available by anonymous ftp from ftp muriel.pcl.ox.ac.uk*

Davis, L., (1985). Job Shop Scheduling with Genetic Algorithms, *Proceedings of International Conference on Genetic Algorithms and their Applications.*

Fonseca, C. M. & Fleming, P.J., (1995). Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction, *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 44 - 52, The Institution of Electrical Engineers.

Fonseca, C. M., and Fleming, P. J., (1996). On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers, *to be submitted to Parallel Problem Solving From Nature 4, Berlin 1996.*

Shaw, R, (1996). Modified Atmosphere Packaging of Ready Meals, *to appear in Proceedings of the Symposium of 'Minimal Processing and Ready Made Foods'*, SIK Goeteborg, Sweden, April 1996.

Starkweather, T., Whitley, D., and Cookson, B. (1993). A Genetic Algorithm for Scheduling and Resource Consumption, Operations Research in Production Planning and Control, Lecture Notes in Economics and Mathematical Systems, pp. 567 - 583, Springer - Verlag 1993.