PAPER NO.600

MINIMIZING FLOWTIME GENETIC ALGORITHMS IN MACHINE SCHEDULING

BY

Nasir Hayat and Andrew Wirth

MINIMIZING FLOWTIME GENETIC ALGORITHMS IN MACHINE SCHEDULING

Nasir Hayat[1] and Andrew Wirth[2]

ABSTRACT

Machine scheduling problems arise in diverse areas such as flexible manufacturing, production planning, computer design logistics, communications etc. A common feature of many of these problems is that no efficient algorithms are known that solve each instance optimally in a time bounded polynomially in the size of the problem. The only option left is to use heuristics but this is at the cost of quality loss of solutions. Thus efforts are directed towards finding heuristics which can help in finding near optimal solutions. This paper examines the application of a relatively new technique namely genetic algorithms to solve single machine scheduling problem with setup times when jobs belong to different types. The results show that genetic algorithms performs better than the available heuristics.
Keywords: Genetic Algorithms, Machine Scheduling, Setup time

INTRODUCTION

Practical scheduling problems often involve processing several types of related jobs on common facilities where a setup time is incurred whenever there is a switch of processing from a job in one type to a job of another type. If the setup times are sequence dependent, the problem is NP-complete. Usually sequence independent setup times problem is considered to be very easy simple [23] as the setup may be added to the processing time, but this may not be true when jobs are divided into mutually exclusive types with setup required when we change from one type of jobs to another type and a splitting can take place. Various scheduling problems in manufacturing and service organizations can be formulated as single facility problems with jobs divided into various type [17]. For example, the scheduling of jobs for a flexible manufacturing system which can produce several different types of products on the same machine but which requires a setup to rearrange or retool work stations when there is a switch in product type. Mason and Anderson [17] obtained necessary conditions for an optimum solution. They then use these to prune the search tree. William and Wirth [27] generated a heuristic which satisfied the Mason and Anderson conditions and appeared to perform well when compared with some standard heuristics.

The range of applications of genetic algorithms has grown rapidly in recent years and genetic algorithms are being used in industrial and financial fields. Recently a number of successful attempts ([1],[3],4],[5],[7],[8],[10],[11],[13],[14],[15],[19],[20],[22],[23],[24],[26]) have been made to demonstrate the applicability of genetic algorithms to machine scheduling problems.

[1] Mechanical Engineering Department, University of Engineering and Technology, Lahore, Pakistan.
[2] Department of Mechanical and Manufacturing Engineering, University of Melbourne, Australia.

SOME CHARACTERISTICS OF USED GENETIC ALGORITHMS

Evolver software package, an odd to the Excel spreadsheet has been used to implement the genetic algorithms to the chosen problem. Evolver uses a steady state approach. This means that only one organism rather than an entire population is replaced at a time. The number of generations can be found by dividing the trials by the size of the population. The following parameters must be set:

Population Size

The population size specifies how many organisms should be stored in memory at any given time.

Selection

In Evolver, parents are chosen using a rank-based mechanism. This procedure begins by rank ordering the population by fitness. Next an assignment function gives each individual a probability of inclusion. The assignment function can be linear or nonlinear. A roulette wheel is then built with the slots determined by the assignment function. The next generation of an unsized population is built by giving the wheel n spins. This procedure guides selection towards the better performing members of the population but does not force any particular individual into the next generation.

Crossover

We used the order solving option available in Evolver. It applies the order crossover operator [8]. Order Crossover (OX) creates children which preserve the order and the position of alleles in a subsequence of one parent while preserving the relative order of the remaining alleles from the other parent. It is implemented by selecting two random cross points which define the boundaries for a series of copying operations. Alleles from the first parent that fall between the two cross point are copied into the same position of the offspring. The remaining allele order is determined by the second parent. Non duplicative alleles are copied from the second parent to the offspring beginning at the position following the second cross point.Both the second parent and the offspring are traversed circularly from that point. A copy of the parent's next non duplicative allele is placed in the next available child position. An example for which the random cross points 3 and 6 is as follows:

Parent1: A B C | G H B | D E
Parent2: H D E | F G C | A D
Child: E F C | G H B | A D

Crossover rate can be varied from 0-1.

Mutation

The order solving method referred to above performs mutations by swapping the positions of some variables in the organism. The number of swaps performed is increased or

decreased proportionately to the increase and decrease in the mutation rate setting (from 0 to 1).

Replacement

The worst performing organism is replaced with the new organism that is created by selection, crossover and mutation.

SOME RESULTS

The problem under study was modeled in Excel spreadsheet. In the absence of any standard benchmark problems for this type of scheduling problems, it was decided to compare the performance of genetic algorithms with the heuristics developed by William and Wirth [27] and Dunstall [9]. Test problems such as the ones in the table below were generated with setup times and processing times (both integers) selected from uniform distribution [0,100] and [1,100] respectively. The number of jobs in each type was also randomly selected with the restriction that N jobs were required in total. The result of implementation for single machine and upto 80 jobs is given below.

Job-Classes-Machine	Lower Bound	Best Heuristic Solution	GA Solution	Best	Deviation of GA from Best Heuristic (%)
20-5-1	10833	12142	12142	-	0.000
30-5-1	18880	20988	20988	-	0.000
40-5-1	35547	42313	40773	GA	-3.639
60-5-1	67678	77183	76456	GA	-0.942
80-5-1	120769	145019	139368	GA	-3.897

CONCLUSIONS

In this paper the problem of scheduling single machine with sequence independent setup times, to minimize the total flow has been solved using Genetic Algorithms. The results shows that genetic algorithms outperforms that available best heuristics. Moreover for the sequence independent setup times, the better solution can be generally obtained by the splitting.

FUTURE RESEARCH

The results reported in this paper have been obtained as a result of preliminary experimentation. Since the successful implementation of the genetic algorithms depends upon the careful selection of the parameters values; optional parameters setting strategy can be found by devising a statistical test. Moreover the work can also be extended to the parallel machines.

REFERENCES

BAGHI, S., UCKUN, S., MIYABE, Y. and KAWAMURA, K., 1991, "Exploring Problem Specific Recombination Operators for Job Shop Scheduling", In Belew, R.K. and Booker, L. B. (editors), Proceedings of 4th International Conference of Genetic Algorithms, Morgan Kaufmann, pp. 10 - 17 [1]

BAKER, K.R., 1974, Introduction to Sequencing and Scheduling, John Wiley & Sons, New York [2]

BEAN, J. C., 1994, "Genetic Algorithms and Random Keys for Sequencing and Optimization", ORSA Journal on Computing, Vol. 6, No.2, pp 154-160. [3]

BIEGEL, J. E. and DAVERN, J.J., 1990, "Genetic Algorithms and Job Shop Scheduling" Computers and Industrial Engineering Vol. 19, Nos 1-4, pp.81-91. [4]

CHEN, C. L., VEMPATI, V.S. and ALJABER, N., 1995, "An Application of Genetic Algorithms for Flowshop Problem", European Journal of Operational Research, Vol. 80, pp.389-396. [5]

CONWAY, R. W., MAXWELL, W.L. and MILLER, L.W., 1967, Theory of scheduling, Addison-Wesley Publishing Co., Massachusetts. [6]

CROCE, F.D., TADEI, R. and VOLTA, G., 1995, "A Genetic Algorithm for the Job Shop Problem", Computers and Operations Reserach, Vol.22, No.1, pp.15-24. [7]

DAVIS, L., 1985, "Job Shop Scheduling with Genetic Algorithms" In Grefenstette, J.J. (editor), Proceedings of an International Conference on Genetic Algorithms and their Applications, Erlbaum, pp. 136-140. [8]

DUNSTALL, S., 1996, Personal Communication [9]

FANG, H.L., ROSS, P. and CORNE, D., 1993, " A Promising Genetic Algorithm Approach to Job Shop Scheduling, Rescheduling and Open Shop Scheduling Problems" In Schaffer, J.D. (editor), Proceedings of 3rd International Conference on Genetic Algorithms,pp.375-382 [10]

FALKENAUER, E., and BOUFFOUIX, S., 1991 "A genetic algorithms for job shop scheduling", Proceedings of 1991 IEEE International Conference on Robotics and Automation, pp.824-829 [11]

GOLDBERG, G. E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, Mass. [12]

GUPTA, M.C., GUPTA, Y. P. and KUMAR, A., 1993, "Minimizing Flow Time Variance in a Single Machine Using Genetic Algorithms", European Journal of Operational Research Vol. 70, pp.289-303. [13]

KANET, J. J. and SRIDHARN, V., 1991, "PROGENITOR: A Genetic Algorithm for Production Scheduling", Wirtschaftsinformatik, pp. 332-336. [14]

LEE, C. Y. and CHOI, J.Y., 1995, "A Genetic Algorithms for Job Sequencing Problems with Distinct Due Dates and General Early-Tardy Penalty Weights", Computers and Operational Research, Vol. 22, No.8 pp.857-869. [15]

LEE, C.Y., 1995, "Genetic Algorithms for Single Machine Job Scheduling with Common Due Date and Symmetric Penalties", Journal of Operations Research Society of Japan, Vol, 37, No.2 pp.83-95. [16]

MASON, A. J. and ANDERSON, E.J., 1991, "Minimizing Flow Time on a Single Machine with Job Classes and Setup Times", Naval Research Logistics, Vol. 38, No. 7, pp.333-350.

MORKAWA, K., FURUHASHI, T. and UCHIKAWA, Y., 1993, "Single Populated Genetic Algorithm and its Application to Job Shop Scheduling", Transactions of the SICE, Vol. 27, No. 5, pp.1014-1019. [18]

MULKENS, H., 1994, "Revisiting the Johnson Algorithm for FlowShop Scheduling with Genetic Algorithms", Journals of Evolutionary Computation, pp. 69-80. [19]

NAKANO, R. and YAMADA, T., 1991, "Conventional Genetic Algorithm for Job Shop Problem" In Belew, R.K and Booker, L.B. (editors), Proceedings of 4th International Conference on Genetic Algorithms, Morgan Kaufmann, pp. 474-479. [20]

POON, P.W. and CARTER, J.N., 1995, "Genetic Algorithms Crossover Operators for Ordering Applications", Computers and Operations Research, Vol.22, No.1, pp. 135-147. [21]

REEVES, C.R., 1995, "A Genetic Algorithm for Flowshop Sequencing", Computers and Operations Research, Vol.22, No.1, pp.5-13. [22]

RUBIN, P.A. and RAGATZ, G.L., 1995, "Scheduling in a Sequence Dependent Setup Environment with Genetic Search", Computers and Operations Research, Vol.22, No.1, pp. 85-99. [23]

SRIDHAR, J. and RAJENDRAN, C., 1994, "A Genetic Algorithm for Family and Job Scheduling in a Flowline based Manufacturing Cell", Computers and Industrial Engineering, Vol. 27, Nos. 1-4, pp. 469-472 [24]

STARKWEATHER, T., WHITLEY, D., MATHIAS, K., and McDANIEL, S., 1992, "Sequence Scheduling with Genetic Algorithms", In Fandel, G. and Jones, J. (editors), New Directions for Operations Research in Manufacturing, Springer-Verlag, pp.129-148. [25]

WHITLEY, D., STARKWEATHER, T. and FUQUAY, D., 1989, "Scheduling Problems and Travelling Salesman: The Genetic Edge Recombination Operator" In Davis, L. (editor), International Conference on Genetic Algorithms, pp.133-140. [26]

WILLIAMS, D.N. and WIRTH, A., 1996, "A New Heuristic for a Single Machine Scheduling Problem with Setup Times", Journal of Operational Research Society, Vol, 47, pp. 175-180. [27]