DISTRIBUTED TASK SCHEDULING AND ALLOCATION USING GENETIC ALGORITHMS

David Todd (d.s.todd@ncl.ac.uk) and Pratyush Sen (pratyush.sen@ncl.ac.uk)

Engineering Design Centre and Department of Marine Technology
Armstrong Building, University of Newcastle,
Newcastle-upon-Tyne, NE1 7RU, United Kingdom.

ABSTRACT

As complexity and size of projects increase so do the problems associated with the scheduling and management of the design, manufacturing and assembly processes. In the context of large projects the ability to optimise the scheduling and allocation of these processes can also aid in tendering for contract as well as the management of the project itself. Many large projects will be constructed across distributed sites, each with their own capabilities and specific areas of expertise. Multiple sources may be needed to provide skilled personnel, raw materials, specialised components or facilities for the project, even whole sub-systems within a complex project may be contracted out for financial or time reasons. This paper demonstrates how a computational intelligence technique know as the Genetic Algorithm can be used to optimise design, manufacturing and construction schedules for multiple objectives such as minimising cost and time and maximising utilisation. The system generates a number of near-optimal project scenarios from which a single solution can be selected and implemented by the project manager.

KEYWORDS

Distributed, Genetic Algorithms, Manufacturing, Multiple Criteria, Planning, Scheduling, Systems, Task Allocation.

INTRODUCTION

Scheduling and coordination are vital elements in a well managed project. The problem tackled in this paper is most closely related to that of the Job-shop Scheduling Problem (JSP) (Davis 1985). The JSP uses a very simple model and is not really capable of modelling real manufacturing environments hence the Complex JSP (Todd and Sen 1998) was developed to handle the increased level of complexity. The Complex JSP formulation allows the user to define the manufacturing environment more completely. Instead of dealing with individual jobs and machines it takes a top level view of the engineering process, dealing with allocation of tasks (design, manufacture and assembly) to various distributed sites (internal, suppliers and sub-contractors). This technique is designed to be used in the early stages of the project or during the tendering stage. At which point a reasonable model of the basic processes and interrelationships involved exists and approximate costs and times of each of the tasks has been obtained from various suppliers and sub-contractors though a quote system.

DISTRIBUTED MANUFACTURING AND ASSEMBLY

Every project is made up of a set of tasks each of which has to be allocated to a given resource. Each of these tasks will have a duration and possibly a completion cost. Each task may also have a set of other tasks without which it cannot start. The tasks may be assigned to a number of different resources. They may also be carried out within the company at one of a number of distributed sites, each of which will have their own specialities and associated costs (Fig.1). There may also be several sub-contractors involved and this may result in having to make a choice, should a cheaper sub-contractor be used who has a longer lead time or is worth spending extra money to get the project done more quickly. Component suppliers may also be chosen on their completion times and costs. As the number of tasks, sites, contractors and suppliers increase, the management of the project and the ability to find good allocations and schedules becomes more difficult. If this problem can be defined in the correct form a Genetic Algorithm (GA) can be used to evolve good project schedules.

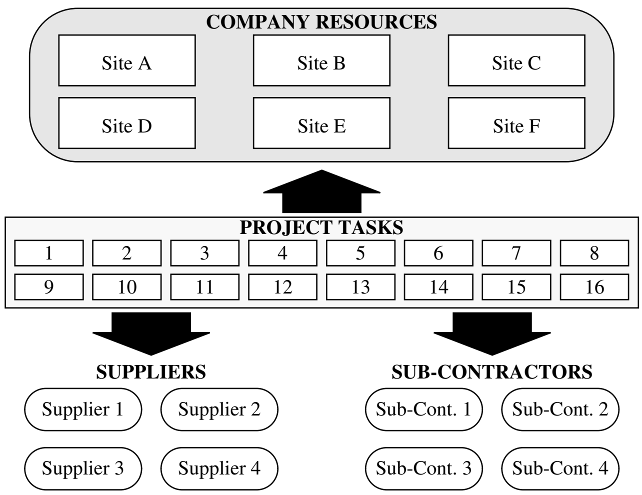

Fig.1 Distributed Manufacturing

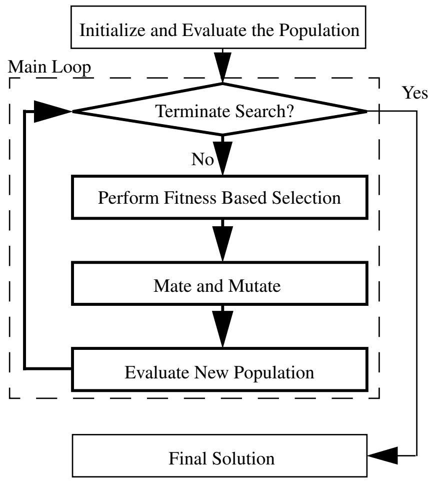

Fig.2 The Simple Genetic Algorithm

GENETIC ALGORITHMS

The Genetic Algorithm (GA) is a computational intelligence technique which evolves solutions from a population of individuals via a process of natural selection and random mating and mutation (Once round loop = 1 Generation). The technique has come in to prominence in the last 10 years due to the increases in available desktop computing power. Goldberg (1989) outlines the simple GA as shown in Fig.2. The genetic algorithm used in this paper is an expanded version of the simple GA, called the Multiple Criteria Genetic Algorithm (MCGA). Instead of finding just one solution the MCGA finds a set of solutions which define the trade-off surface between multiple search criteria. Then various Multiple Attribute Decision Making (MADM) methods (Huang 1981) can be used to select a single ‘best’ solution. A more detailed technical explanation of the mechanics of the MCGA is available elsewhere (Todd and Sen 1997).

SCHEDULING USING GENETIC ALGORITHMS

Genetic Algorithms require only a suitable representation and an evaluation mechanism. As such it is a very general technique which can be applied to a wide range of problems. In this case each task is given a number and the action of the GA is to modify the ordering and allocation. The ordering reflects the priority of the tasks at each site and hence good orderings will give improvement in time. These priority lists are put into a schedule builder which acts as the evaluation mechanism. The schedule builder takes the priority lists uses them to create a schedule and collect performance data for that schedule. This data is passed back to the MCGA for analysis. The code will also generate plans showing each site, what tasks are being performed, the availability of site and also statistics from the run.

TEST PROBLEM

In order to test the system a 20 site 50 task problem was used. Each of the tasks could be completed by up to 5 different sites each offering different time and cost quotes. For each of the tasks a list of dependencies is given. Some tasks are generated with no upstream or downstream dependencies. These tasks can be scheduled freely but as they use a resource they still effect total processing time. It is also possible to perform multi-project scheduling by having several separate dependency diagrams and combining them in one problem. It is assumed that each site can only perform one assigned task but large sites can be simulated by several sites grouped together. In this example the schedules are evaluated on two criteria, total time for project completion and total project cost, both of which are to be minimised. After every generation the best strings found so far are stored within a Pareto Population which is used to give a picture of the developing trade-off surface. Figure 3 shows how this surface develops over the 100 generations of the run. The initial randomly generated population contains solutions which are clearly sub-optimal. However, from these solutions the MCGA is able to synthesise better and better solutions. Generation 10 shows the

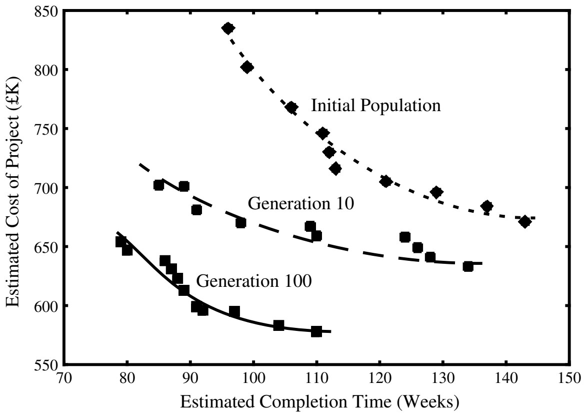

Fig.3 MCGA Generated Trade-off Curve of Cost Against Completion Time

early progress of search. Initial generations tend to undergo the biggest improvements as useful and significant configuration changes are easily found. The final curve shows the trade-off after 100 generations when search is terminated. At this point the user can obtain a printout of any of the schedules and their associated cost and utilisation breakdowns. The system allows the user to experiment with different goals and site configurations. It can also be used to investigate how the effects of cost increases within the internal workings of the company effect the distribution of tasks and competitiveness. Additional criteria can be added such as maximise internal money allocation in order to maximise profits or maximise the utilisation of site x if site x is known to be a efficient or cost effective site. Sites can be easily added and removed as can tasks and dependencies. The schedules can be re-optimised and results can be plotted as sets of curves from which direct comparisons of the various configurations can be carried out. This ability to add and remove criteria and edit the site and task information make the system useful and highly flexible.

CONCLUSION

This paper was concerned with the application of Genetic Algorithms to a task allocation and scheduling problem. The various techniques involved were explained and demonstrated using a 50 task - 20 site example. From the results the software can be seen to be improving the quality of schedules over the run in terms of project time and costs. The software does not require and excessive amount of computing time in order to gain reasonable estimates of the trade-offs surfaces involved. However, for larger scale problems and those with more criteria the execution time does increase.

The software provides plans of the output schedules and relevant statistical data from which various graphs can be plotted. These plans can be used as a guide for the project planner to aid him throughout the course of the project. Due to the short processing time the software can also be used in a pseudo real-time manner. For example if a particular task in the project is found to be over-running a new estimate of its completion time can be given and all remaining tasks can be re-scheduled to give a new set of schedules. From that point a new schedule could be implemented. This may involve moving jobs from one site to another in order to meet a specified deadline.

The problem model presently used is a fairly simple one involving only times and costs however it is reasonable for the problem tackled here. Additional complexity can be added by including transportation costs between sites and the possibility of the introduction of stochastic methods which increase and decrease costs and times about a given mean value. This would mean that the software could generate best, mean and worst case schedules for comparison. This would take the schedule into the area of robust scheduling, where minimum time and cost is considered against susceptibility to task delays. Supply chain management is also an area currently receiving much research interest and a model which looks further down the supply chain would also provide an interesting area for further work.

It is hoped that this paper has demonstrated the usefulness of Computational Intelligence techniques within the Industrial Scheduling and Allocation field. Increased awareness in the operation and possible application of such techniques can only lead to future benefits within the field of engineering and within Industry worldwide.

REFERENCES

Davis, L. (1985). Job shop scheduling with genetic algorithms. *Proceedings of The First International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates, Hillsdale, NJ. 136-140.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimisation and machine learning*. Addison-Wesley Publishing, ISBN 0-201-15767-5.

Huang, C.L. and Yoon, K. (1981). *Multiple Attribute Decision Making Methods and Applications: a State of the Art Survey*. Springer Verlag, New York.

Todd, D.S. and Sen, P. (1997). A multiple criteria genetic algorithm for containership loading. *Proceedings of the Seventh International Conference on Genetic Algorithms*. East Lansing, MI, Morgan Kaufmann Publishers. 674-681.

Todd, D.S. and Sen, P. (1998). Tackling complex scheduling problems using operations based scheduling. *Adaptive Computing in Design and Manufacture*: *ACDM '98*. Plymouth, UK, Springer-Verlag, London. 45-58.