A GENETIC ALGORITHM APPROACH TO SYSTEM SCHEDULING

David S. Todd, Jaime A. Scott and Pratyush Sen

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

Abstract: The complexity of an engineered system is usually directly related to its size. As the size of the system increases, the number of components, the level of interaction, the difficulty of management, and the effort required to optimise, all increase. This paper outlines a method, based on a systems approach, which uses the evolutionary computing technique of Genetic Algorithms to schedule and co-ordinate processes (jobs or activities) and resources within such complex systems toward multiple objectives.

Keywords: Design, Scheduling Algorithms, Multiobjective Optimisations, Genetic Algorithms, Systems Engineering, Integration

1. INTRODUCTION

Based on the observations of Hall (1989), a system can be defined as a collection of four components; a hierarchy of sub-systems containing processes (activities or jobs), a set of tools and techniques, a set of people, and a set of inter-relationships. A process has a set of outputs and objectives. In a job-shop sub-system a process may be represented as the manufacture of a component on a machine. In this case, the output is the machined component, whilst the objective may be the requirement that the machined component meets certain quality standards. The outputs and objectives of each individual process are achieved through the use of resources. The resources relate to tools, techniques and people. Again, in the case of the job-shop, the tools include milling machines and lathes, the techniques include standard working practices for the operation of such machines, whilst the people resources are the machine operators. The processes along with their resources begin to define the system. Whilst each process has its individual outputs and objectives, the sub-system and the system as a whole also have their own outputs and objectives. For example, the job-shop sub-system outputs a batch of components with the objective of achieving a specific delivery deadline. Whilst the individual processes and their resources begin to define the system, the definition is completed by a clear and concise understanding of the inter-relationships that tie the sub-systems and processes together as a whole. In fact the cornerstone of a systems approach is based on such an understanding, without which local process efficiency gains will have little or no impact on the overall system efficiency.

Standard scheduling methods, such as CPM (critical path methods) and PERT (program evaluation and review technique), address the process inter-relationships in terms of activity precedence, with limited account taken of resource implications. Using a systems approach, the research summarised in this paper addresses the development of a genetic algorithm-based approach to large-scale system scheduling. Using not only an understanding of the process inter-relationships, but also an understanding of sub-system and resource inter-relationships. In this way, system schedules can be optimised according to time, cost and resource-based objectives, under the constraints imposed by inter-relationships across sub-system boundaries.

2. RESOURCE-CONSTRAINED SCHEDULING

The objective of CPM and PERT is to complete a project as soon as possible, with respect to the constraints of precedence between processes, assuming unlimited resource availability. However, in most cases the assumption of unlimited resources cannot be justified. In reality only a finite amount of resources will be available whilst the cost of acquiring additional resources could be prohibitive. As demonstrated by Just and Murphy (1994) the difference between schedules that are resource-constrained, and those that are not, can be significant. Unanticipated overtime, schedule delays, and the resulting cost overruns can all be symptomatic of scheduling undertaken without adequate consideration for resources and their limits.

Research has centred around optimising time-based objectives, such as the minimisation of total project duration (makespan), or the minimisation of tardiness. Sometimes, schedules optimised purely on a time basis are inefficient in terms of cost. Papers summarising research centred around cost-based objectives, such as the maximisation of a project's net present value (NPV), or the minimisation of total project cost, are less common. A third area of research addresses the optimisation of resource-based objectives, such as the maximisation of resource utilisation, or the minimisation of variation in resource levels. Resource-based objectives are usually considered as a combined objective with either, or both, the time or cost-based objectives.

Most reported methods deal with a single objective, whether it be time, cost, or resource-based. More recently, multiple objective approaches have begun to emerge (Speranza et al., 1993; Slowinski, 1981 & 1989). However, examples of such work are still scarce.

Solution approaches for solving the resource-constrained scheduling problem can be classified as either heuristic (approximate) or optimising (exact). Many priority rules have been developed for use with heuristic schedule methods. However their performance is problem dependant and they do not provide good solutions consistently (Boctor 1990).

Recent improvements in computer hardware capabilities have enabled the development of evolutionary computation search methods. The most popular methods include Simulated Annealing, Tabu Search (Lee et al., 1996) and Genetic Algorithms (Bruns, 1993; Nakano et al., 1991; Gen et al., 1994; Davis, L, 1993; Lee et al., 1996). Such techniques go some way towards overcoming the inefficiencies of previous optimisation techniques being able to tackle the problem by treating it as a 'black box'.

3. GENETIC ALGORITHMS

The Genetic Algorithm (GA) is an evolutionary problem solving technique. The technique is named as such because it evolves solutions from a population of individuals via a random-based process of natural selection, mating and mutation. The technique was first developed by Holland (1972), however, only recently has the technique become more widely used. Goldberg (1989) outlines the simple GA as shown in Figure 1.

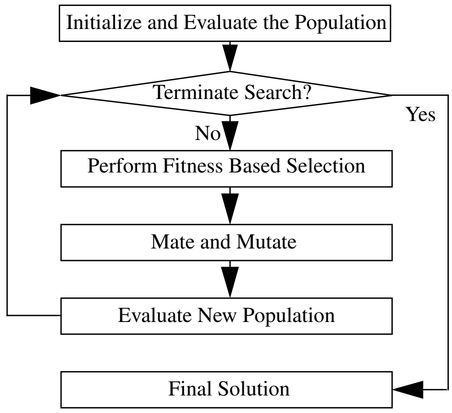

Figure 1. The simple genetic algorithm

The genetic algorithm used for the applications detailed in Section 4 is an expanded version of the simple GA, called the Multiple Criteria Genetic Algorithm (MCGA). The MCGA randomly generates a population of individuals. Each individual represents one possible solution to the problem under investigation. The individuals can be assessed on several criteria which can be specified based on the problem objectives. These criteria are maintained separately at all times, allowing simultaneous maximisation and minimisation of separate criteria during the search. However, instead of finding a single near-optimal solution, the case for a normal GA, the MCGA finds a set of solutions which define the trade-off surface between these search criteria. These solutions form a Pareto Set or Efficient Boundary. The solution which most closely matches the decision makers preferences can then be selected from this set and either analysed further or implemented accordingly. A further more detailed explanation of the MCGA is available in Todd et al. (1997a).

4. THE MCGA SCHEDULER

Research into the application of genetic algorithms to resource-constrained scheduling is gaining popularity. However, most, if not all, documented research addresses the limited case of single-mode resource allocation. In such cases, an initial schedule is built from a list of processes that is ordered according to precedence constraints. The processes are considered according to the ordering of the precedence list, and are scheduled to begin at the earliest point in time when sufficient resources become available. The GA develops new sequences through crossover and mutation. However, because of precedence constraints, standard operators require modification in order to avoid the creation of infeasible schedules. Developed sequences are evaluated and chosen for inclusion into the next generation's mating pool using selection.

The MCGA Scheduler combines the previously described MCGA Search Mechanism with a Schedule Simulator. It differs most from existing GA-based schedulers through the consideration of multiple criteria. Instead of minimising a single objective such as makespan, multiple objectives can be handled in order to develop schedules that are optimal with respect to a wider range of combined criteria. In addition, the MCGA Scheduler is capable of handling multiple-mode resource allocation, as well as developing schedules that can take into account resource unavailability, set-up times between operational modes, and constraints on individual process start and finish times. In order to improve flexibility of application, the MCGA Scheduler can also handle, not only precedence constraints between process stages, but also precedence constraints between processes. The MCGA Scheduler formulates the resource-constrained scheduling problem using a number of generalized components(Todd et al., 1998).

Precedence Matrix: The precedence matrix defines the processes and their precedence inter-relationships

Resource Efficiency Matrix: The resource efficiency matrix lists all resources and details their relative efficiency against a list of operation-types. For a job-shop problem, the resources relate to the individual machines (e.g. Machine 01, Machine 02, etc.), whilst the operation-types relate to job descriptors (e.g. milling, turning, drilling etc). For design organisation, the resources are the individual design engineers, whilst the operation-types are activity descriptors (e.g. stress analysis, engineering drawing, etc). The matrix elements represent the relative efficiencies of each resource, for each of the pre-defined operation-types. The resource efficiency matrix is defined at the organisational level, and can be updated over-time. Resources can be added and deleted, whilst the efficiencies are updated as a result of maintenance (Job Shop) and training and experience (Design Project Organisation)

Process-Operation Matrix: The process-operation matrix lists all processes detailing the operation-type that describes the process. For added flexibility, each process can be broken down into a sequence of stages to allow representation of variable resource allocation over the duration of a process.

Constraint data: The MCGA Scheduler can incorporate periods of time, during which activities cannot be scheduled, as well as constrained process start and finish times.

Using all input data, the Schedule Simulator builds a valid sequence, by trying, at each decision point in time, to put in process all unscheduled processes that satisfy precedence constraints, resource availability, down-time considerations and any pre-defined process start/finish constraints. In this manner, the algorithm builds a complete schedule by successively adding unscheduled processes to a partial schedule.

The variation in the derived schedules results from the variable choice of resource allocation. The specific allocation of resource to a process represents an operational mode. Because each process has at least one operational mode, and since the operational mode determines the duration of a process, there are a large number of possible combinations of such modes, each representing a different schedule. Each schedule is evaluated against a number of criteria, including lead-time, utilisation of most efficient resources and project cost. A fitness value for each of the criteria is derived. The resulting combinations of operational modes and their corresponding fitness values are returned to the MCGA in order to derive fitter combinations of operational modes, and the corresponding schedules.

The Schedule Simulator has been designed in such a way that it will tackle a broad range of scheduling problems. Two such examples are briefly documented in the following sections.

4.1 Flexible Manufacturing Scheduling

The aim of the scheduler in this case is to determine the job sequences on each machine in order to minimise the total makespan of the overall system. Whilst the MCGA Scheduler has been used on much larger scale problems (Todd et al., 1997b), a simple 4 job, 3 machine example is used to demonstrate the limitations of most existing single-mode resource allocation job-shop scheduling algorithms. From basic data a preliminary schedule can be generated (Figure 2). Obviously the schedule in Figure 2 is far from optimal and could be modified, with limited thought and intuition, to obtain the optimal schedule (Figure 3).

The job-shop sub-system belongs to a higher level system, that of the whole manufacturing facility. With this systems approach in mind, the job-shop should not be considered in isolation. With the application of

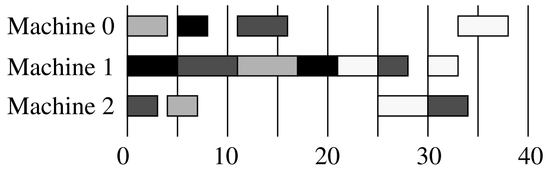

Figure 2. Preliminary Schedule

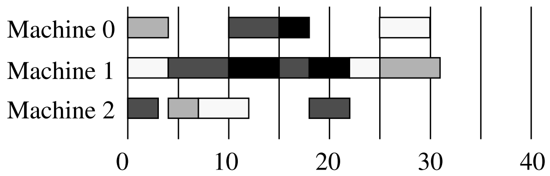

Figure 3. Single Mode Optimised Schedule

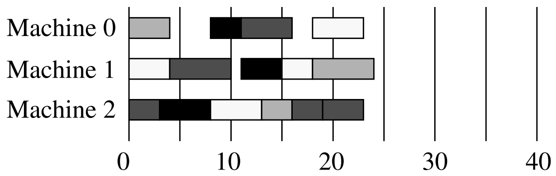

Figure 4: Upgraded Machine Schedule

concurrent engineering, there are even greater requirements for the integration of the design, manufacturing and assembly stages of the total manufacturing system. The relationships between the job-shop and its up-stream and down-stream sub-systems must be considered by the scheduler. Job-shop start-times may be dictated by the completion of upstream sub-systems, whilst finish-times may be constrained by strict deadlines imposed by downstream requirements. There are also cases when certain manufacturing jobs must wait for the completion of other jobs. Whilst most classical job-shop methods consider the precedence constraints between job operations, few consider the concept of precedence constraints existing between jobs, both internal and external to the job-shop.

The application of flexible manufacturing practices, precipitated by the development of more flexible machinery, means that jobs are no longer tied to a single machine (single mode). Instead, jobs can be processed on one or more of a choice of machines (multiple mode). Such machines are capable of performing several manufacturing operations with varying time-based operational efficiencies. Because such machines require tool change and reset between different operational modes, set-up times become an integral part of any derived schedule. In addition, in order for schedules to be more realistic, schedulers must be capable of incorporating down-time periods, during which machines are taken off-line for routine maintenance.

Based on these factors, which all affect the development of an effective production schedule, the objectives of a job-shop scheduler become more complex. Makespan is no longer the only objective, we may also want to minimise penalty costs for late deliveries, reduce the amount of jobs in progress because of limited space requirements, maximise machine utilisation or reduce queues. The MCGA Scheduler is able to handle this complexity and develop near optimal solutions and allows experimentation with different machine configurations and multiple-mode resource allocation. For example, if a machine was identified as a bottleneck, a reconfiguration could be implemented in the Schedule Simulator, by adding a second machine or by upgrading the efficiency of the existing machine through repair and maintenance. The benefits of this change can be assessed by comparing the new trade-off surface with the surface derived previously, comparing the gains derived against the cost of such an implementation. Figure 4, based on the previously specified problem, shows that gains can be made through the reduction of makespan if Machine 2 were to be modified to allow it to complete the jobs previously allocated to Machine 1.

If optimisation is carried out on a more complex problem involving all of the elements described above a trade off curve is generated. If the manufacturing environment is modified and re-optimised a set of curves can be generated such as the ones shown in Figure 5. These allow analysis of the relative benefits of modifying or expanding the manufacturing system in different ways.

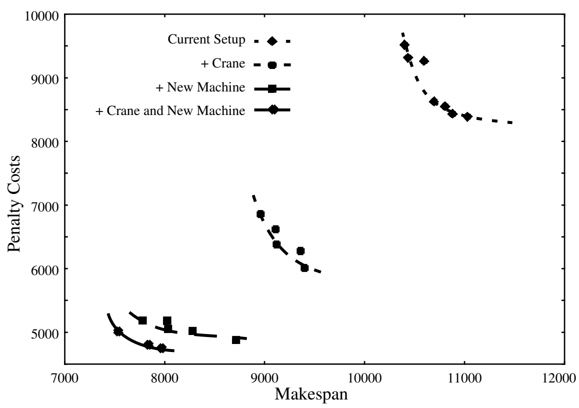

Figure 5: Relative Effect of System Modifications

4.2 Resource-constrained Design Project Scheduling

The specific application addressed by ongoing research is that of resource-constrained scheduling applied to design projects. Existing network planning techniques, such as Critical Path Method (CPM), have been developed specifically for the planning and monitoring of fabrication projects. Ideally, fabrication projects do not involve iteration or rework. In fact, one of the responsibilities of engineering design is to resolve all iteration and feedback prior to the fabrication phase of any project. As a result, existing planning and scheduling techniques do not have the flexibility to adequately represent data flow-based projects. In order to effectively plan and schedule design projects, a modelling technique is required that can deal specifically with the intricacies of data flow. A representation of the design process, called the design structure matrix (DSM) system (Steward, 1981), can be used as the basis for analysis prior to scheduling. The system begins with a precedence matrix detailing the design activities and the data flow dependencies between the activities. By partitioning the precedence matrix (to reduce iteration) and then resolving the remaining iterative blocks (on the basis of heuristics and full user interaction) the starting matrix can be unravelled to obtain a sequence of design activities, linked only by downstream data flow (Scott 1998a,b). Figure 6 illustrates a partitioned matrix after Iterative Block Resolution. The matrix can then undergo resource-constrained scheduling.

It is necessary to carry out preliminary analysis prior to scheduling for data-flow projects, and in particular, for design projects. It should be highlighted that resource-constrained scheduling offers maximum benefit therefore, in terms of lead-time and resource utilisation, for projects that use scarce, distinct resources. In order to gain full benefits, it is necessary to quantify individual resources (e.g. design engineers) and their relative skill-set. The majority of large-scale fabrication projects use large numbers of resources that require no further definition beyond the size and type of each trade-group (e.g. number of welders). Resource-constrained scheduling may not

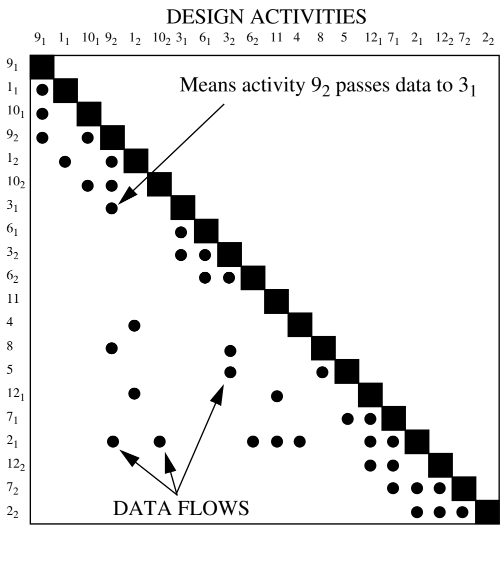

Figure 6: Unravelled -Partitioned Precedence Matrix

be necessary for such projects, however fabrication projects that use specialised, and highly skilled, resources can reap benefits through the use of such scheduling.

The following example illustrates the application of the MCGA Scheduler to a resource-constrained project representing the activities, durations and precedences illustrated in Figure 6. Each of the twelve distinctive design activities are defined by an individual operation type. There are three design engineers available, with capabilities and efficiencies.

Critical path analysis, without resource considerations, yields the earliest possible project completion time equal to 82 time units. The MCGA Scheduler has been used to develop a resource-constrained schedule with three objectives; minimisation of project completion time, maximisation of all resource utilisation, and maximisation of the utilisation of Resource 1. The resulting schedule is illustrated in Figure 7.

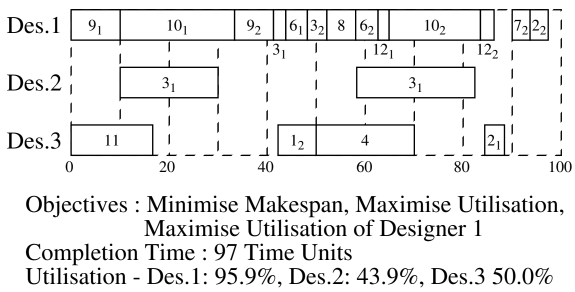

Figure 7: Multiple criteria optimised schedule.

5. CONCLUSIONS

This paper explained how, in any scheduling problem, it is important to understand, not only the inter-relationships between processes within a sub-system, but also those between the sub-systems themselves in order to gain maximum benefits. By developing a general purpose scheduler that considers delivery and due dates as constraints, scheduling of processes can be considered within the context of the whole system. For large-scale systems, with many sub-systems and processes, an understanding of inter-relationships across sub-system boundaries is imperative. Two scheduling problems were used to demonstrate the action of the MCGA Scheduler, which has been developed to deal simultaneously with the range of criteria that often arise in such problems. The scheduler is under continuous development and future reports on more complex, real-life applications are anticipated.

REFERENCES

Boctor, F.F. (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, Vol 49, No 6, pp 3-13.

Bruns, R. (1993). Direct chromosome representation and advanced genetic operators for production scheduling. *Proceedings of the Fifth International Conference on Genetic Algorithms*. pp 352-359, Morgan Kaufmann Publishers.

Davis, L. (1993). Job shop scheduling with genetic algorithms. *Proceedings of the Fifth International Conference on Genetic Algorithms*. pp136-140, Morgan Kaufman Publishers.

Gen, M., Y. Tsujimura and E. Kubota (1994). Solving job shop scheduling problems by genetic algorithm. *IEE International Conference on Systems, Man and Cybernetics.* Vol 2, pp 1577-1582.

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

Hall, A.D. (1989). *Metasystems methodology: a new synthesis and unification*. Pergamon Press.

Holland, J.H. (1962). Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery* (ACM). Vol 3, pp 297-314.

Just, M.R. and J.P. Murphy (1994). The effect of resource constraints on project schedules. *Transactions of the American Association of Cost Engineers*. pp E1.1-1.6.

Lee, J. and Y. Kim (1996). Search heuristics for resource constrained project scheduling. *Journal of the Operational Research Society*. Vol 47, No 6, pp 678-689.

Nakano, R. and T. Yamada (1991). Conventional genetic algorithm for job shop problems. *Proceedings of the Fourth International Conference on Genetic Algorithms.* pp 474-479, Morgan Kaufmann Publishers.

Scott, J.A. and P. Sen (1998a). Focusing design re use initiatives using the design structure matrix system. *Engineering Design Conference '98.* June 98, United Kingdom.

Scott, J.A. and P. Sen (1998b). Reducing product development times using the design structure matrix system. *2nd International Conference on Engineering Design and Automation*. August 98, Hawaii, USA. To be published.

Slowinski, R. (1981). Multiobjective network scheduling with efficient use of renewable and non-renewable resources. *European Journal of Operational Research*. Vol 7, pp 265-273.

Slowinski, R. (1989). Multiobjective scheduling under multiple category resource constraints. In *Advances in Project Scheduling*. Slowinski, R. and Werglarz, J. (eds.), Elsevier, pp 151-167.

Speranza, M.G. and C. Vercellis (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*. Vol 64, pp 312-325.

Steward, D.V (1981). *Systems analysis & management*. Petrocelli Books.

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

Todd, D.S. and P. Sen (1997b). Multiple criteria scheduling using genetic algorithms in a shipyard environment. *Proceedings of ICCAS 97, the International Conference on Computer Applications in Shipbuilding*. Yokohama, Japan.

Todd, D.S. and P. Sen (1998). Tackling Complex Job-Shop Problems Using Operations Based Scheduling. *Proceedings of ACDM 98, the International Conference on Adaptive Computing in Design and Manufacture*, Plymouth, UK, Springer Publishers.