A Genetic Approach for Dynamic Job-Shop Scheduling Problems

Ana Madureira[*] *Carlos Ramos*[*] *Sílvio do Carmo Silva*[†]

[*] Instituto Superior de Engenharia do Porto, Dept. de Engenharia Informática
Rua de São Tomé 4200, Porto-Portugal,
Email: {anamadur, csr}@dei.isep.ipp.pt

[†] Universidade do Minho, Dept. de Produção e Sistemas
4710-057, Braga-Portugal
Email: scarmo@dps.uminho.pt

1 Introduction

Since Davis [4] proposed the first Genetic Algorithm(GA) to address scheduling problems in 1985, GAs have been widely used in manufacturing scheduling applications. However, most of the works deal with optimisation of the scheduling problem in static environments, whereas many real world problems are dynamic, frequently subject to several sorts of random occurrences and perturbations, such as random job releases, machine breakdowns, jobs cancellation and due date and time processing changes.

Due to their dynamic nature, real scheduling problems have an additional complexity in relation to static ones. In many situations these problems, even for apparently simple situations, are hard to solve, i.e. the time required to compute an optimal solution increases exponentially with the size of the problem [1]. GAs have been extensively used in the context of Job-Shop Scheduling Problems (JSSP). If all jobs are known before processing starts the JSSP is called static, while if job release times are not fixed at a single point in time, i.e. jobs arrive to the system at different times, the problem is called dynamic. Scheduling problems can also be classified as deterministic, when processing times and all other parameters are known and fixed, and stochastic, when some or all parameters are uncertain [7].

The proposed approach deals with these two cases of dynamic scheduling: deterministic and stochastic. For such class of problems, the goal is no longer to find a single optimum, but rather to continuously adapt the solution to the changing environment. The purpose of this paper is to describe an approach based on GA for solving dynamic scheduling problems, where the products (jobs) to be processed have due dates. This paper starts by presenting a scheduling system, based on Genetic Algorithms for the resolution of the dynamic version of Single Machine Scheduling Problem (SMSP). The approach used adapts the resolution of the static problem to the dynamic one in which changes may occur continually. This takes into account dynamic occurrences in a system and adapts the current population to a new regenerated population. Then, it is proposed an approach for the resolution of the Job-Shop Scheduling Problem (JSSP) in dynamic environments.

The paper is structured as follows: section 2 provides a description of the considered scheduling problem. Section 3 summarises an approach for the resolution of the Dynamic Single Machine Scheduling Problem. The proposed approach for dynamic scheduling is presented in section 4. Finally, the paper concludes with a summary and some ideas for future work.

2 Problem Definition

Almost all practical scheduling problems can be described in terms of the Job-Shop Scheduling Problem. Usually as restricted or relaxed versions of this classic combinatorial optimisation problem. The general Job-Shop Scheduling Problem (JSSP) can generally be described as the allocation of a set of resources over time to perform a set of tasks. In manufacturing systems, scheduling typically concerns allocating a set of machines to perform a set of jobs within a certain time period. Each job has a specified processing order through the machines, i.e. a job is composed of an ordered list of operations, which are characterised by the machine required, and the processing time on it. Several constraints on jobs and machines can be defined: machines are always available and never break down; there are no precedence constraints among operations of the different jobs; the operations processing can not be interrupted and each machine can process only one job at a time; each job can be processed only on one machine at a time; setup times are independent of the schedules and are included in processing times, due dates and technological constraints are deterministic and known in advance. A schedule is an assignment of jobs over time onto the respective machines. An additional constraint will be considered in our approach for the JSSP, the existence of different job release dates. The objective is to find a schedule, which optimizes some performance measure. For literature on this subject see for example Blazewicz [1] and Brucker [2].

It is well known that this problem is hard to solve, i.e. the time required to compute an optimal solution increases exponentially with the size of the problem [1]. A vast amount of literature about GA applications to the scheduling problem has been published in the last few decades. One of the earliest published works on the application of GA to scheduling is that by Davis [4]. For an overview see for example [3]. In reality many scheduling problems are not so well defined. The environment may be dynamic with new jobs arriving at unpredictable intervals, machine breakdowns, jobs cancellation and due date and time processing changes. GA approaches to dynamic JSSP are scarce [3].

3 Dynamic Single Machine Scheduling Problem

The study of Single Machine Scheduling Problem (SMSP) is important by itself and also, among other reasons, because it can provide help and insight into the resolution, understanding, managing and modelling more complex multi-processor problems. In fact, quite often, the single-machine problem appears as a component in larger scheduling problems [1]. Sometimes SMSP are independently solved and results incorporated into larger and more complex problems. The static SMSP refers to the situation in which all jobs are simultaneously available for processing. The complexity increases when the dynamic problem is to be solved. Considering different job release times, the completion time C_j of a job j in a sequence can be given by:

$$C_j = Max\{C_{j-1}, r_j\} + p_j \tag{1}$$

where r_j is the release time of j.

We can see the static SMSP as a relaxation of the dynamic SMPS by considering release times equal to zero, i.e. r_j=0, for all j. In a previous work [6] the adequacy and efficiency of the Genetic Algorithms (GA), for the static Single Machine Scheduling Problem (SMSP) was studied. This work starts by studying the performance of two interrelated GA for the minimisation of the static Weighted Tardiness. One is a single start GA, the other, called MetaGA, is a multi-start version of the GA. The performance was evaluated, on the basis of the quality of scheduling solutions obtained for a limit on computation time. The obtained results show that these GA perform well for the cases studied, being possible to find good solutions in a short time, i.e. a few minutes of CPU time. Substantial performance improvements with the MetaGA were obtained in relation to single start GA.

In a dynamic environment frequent rescheduling of work is necessary due to variations on working conditions and requirements over time. This is due to many random events or disturbances as previously

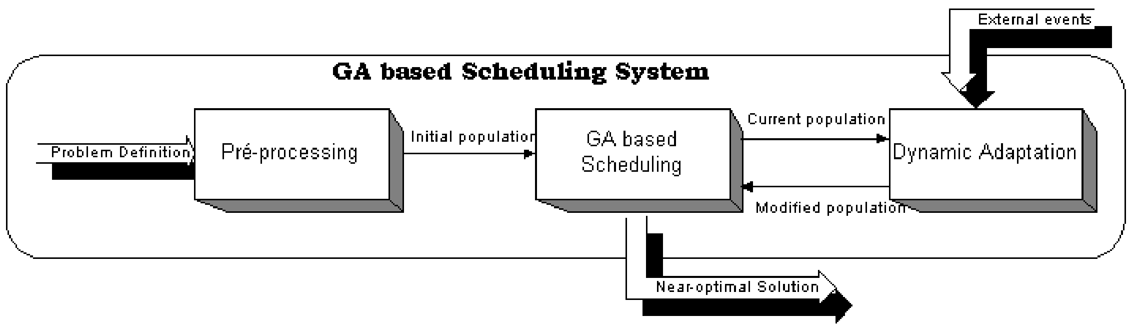

Figure 1: Dynamic Scheduling System based on GA

referred. A scheduling system based on GA for the resolution of the dynamic version of the SMSP is proposed in [6]. The used approach adapts the resolution of the static problem to the dynamic one in which changes may occur continually. This takes into account dynamic occurrences in a system and adapts the current population to a new regenerated population.

This dynamic scheduling system can be structured in three modules, namely pre-processing, scheduling and dynamic adaptation modules, Figure 1.

Pre-processing module

The pre-processing module deals with processing input information, namely problem definition and instantiation of algorithm components and parameters. With the input information the pre-processing module generates the initial individual and the initial population for the $1\,|r_j|\,L_{max}$ problem, following the mechanisms described in [5] and [6].

GA based Scheduling module

The scheduling module is concerned with the application of the genetic algorithm for finding a near-optimal solution to a deterministic problem, where all job attributes are known in advance. The algorithm described in [6] and its MetaGA version, can be applied after some adaptations to take into account different job release times. One of these has to do with the fitness function. Thus, whenever new events occur, deterministic problems are generated by the dynamic adaptation module and then solved by the scheduling module.

Dynamic Adaptation module

Rescheduling due to new events can be based on two approaches. The first discards the current population and generates a new population from the beginning whenever a new event, changing the problem conditions, occurs. This is achieved by restarting the scheduling module with the new problem. In real scheduling problems, continually subject to several changes over time, on occasions this approach may not perform well, because it implies many successive restarts, lost of inherit genetic characteristics from current population of individuals and is likely to be time inefficient. With the other approach, a new population is created through modification of the existing one to take into account changes that have occurred on individuals. This modification of the current population is carried out between successive random events. This means modifying the current individuals and population size taking into account problem changes due to external events, allowing the schedule module to continue the search process based on the new modified or regenerated population. This procedure, leads to inheritance of good characteristics of schedules and is likely to be more efficient. When the random event is a new job arrival then it must be inserted into the sequence of existing jobs. To carry out this performance two procedures can be implemented: randomly select one position to insert the new gene into the chromosome, i.e. randomly schedule a new job among the existing jobs; or use some intelligent mechanism to insert the new job into the sequence. When a job is cancelled, the correspondent gene has to be deleted from every chromosome. After the insertion or deletion of genes is carried out, population updating is done by updating the size of the population and ensuring a structure identical to the existing one. Thus, either some further chromosomes have to be generated through some procedure, i.e. REED rule [5], or some existing chromosomes have to be deleted. In this case either the choice could be random or fall on the less adapted chromosomes. After this is done, the scheduling module can apply the search process with the new updated population.

4 Dynamic Job-Shop Scheduling Problem

The static scheduling problem refers to the situation in which all jobs are simultaneously available for processing. The complexity increases for dynamic problems. Whenever an unexpected event happens in a manufacturing environment, a scheduling decision must be made in real time about the possible reordering of jobs. This process is known as "rescheduling". The proposed approach to solve the JSSP consists on generate a predictive schedule [8] in advance using the available information. When disruptions occur in the system during the execution, the current schedule is modified or revised in order to consider the recent modifications. But an additional problem appears to be solved: "When should rescheduling be performed ?". The approaches considered in the literature can be grouped into three categories: continuous, periodic and hybrid rescheduling. In continuous rescheduling the system reschedule whenever an event modifying the state of the system occurs. In periodic rescheduling the current schedule is modified at regular time intervals. Finally, for the hybrid rescheduling the current schedule is modified at regular time intervals if some disruption occurred. The proposed approach can be applied to static and dynamic production environments, where new jobs arrive continually in the manufacturing system, jobs can be cancelled, and due date and time processing changes can occur. The problem $J\,|r_j|\,L_{max}$ is decomposed into a serie of deterministic SMSP $1\,|r_j|\,L_{max}$, which will be solved consecutively, one by one. Each machine is considered as $1\,|r_j|\,L_{max}$ problem with release dates and due dates and solved to optimality by a Genetic Algorithm (the algorithm parameterization is defined on [6]), the obtained solutions are then incorporated into the main problem, using the following procedure:

Step 1: Determine the estimates of due dates for all operations of each job;

Step 2: Determine the interval of estimates of release times for all operations of each job;

Step 3: Define all SMSP $1\,|r_j|\,L_{max}$ based on information defined in Step1 and Step 2;

Step 4: Solve all SMSP $1\,|r_j|\,L_{max}$ with those release and due dates, using the GA described in [5];

Step 5: Integrate all the optimal or near-optimal solutions into the main problem;

Step 6: Verify if they constitute a feasible solution and terminate with a local optimum; otherwise it is necessary to apply a repair mechanism.

4.1 Dynamic Scheduling

In a dynamic environment frequent rescheduling of work is necessary due to variations in working conditions and requirements over time. This is due to many random events or disturbances as previously referred. A scheduling system based on a GA [6] is designed to react to such random events. These could be classified in two categories: **Partial events**, which imply changes in job (or operations) attributes, such as processing times, due dates and release times; **Total events**, which imply changes in population structure, such as new job arrivals or jobs cancellation. While partial events only require a modification procedure to redefine job attributes and a re-evaluation of the fitness function of solutions, total events require a modification on chromosome structure and size, by inserting or deleting genes, as well as the re-evaluation the fitness function. Therefore, under a total event the modification of the solutions population is imperative. In this work, this is carried out by the mechanisms described in [5]. Rescheduling from the beginning is normally to be avoided, considering the processing times involved and the frequency of the rescheduling. However if work has not yet started and time is available, then an obvious and simple approach to rescheduling would be to restart the scheduling from scratch with the new modified population resulting from a disturbance, i. e. a new job arrival. When there is not enough time to reschedule from scratch or job processing has already started, a re-use approach must be considered which builds on the current schedule.

5 Conclusions

In this paper it was proposed an approach based on GA for solving the dynamic scheduling problem, where the products (jobs) to be processed have due dates.

The paper starts by presenting a scheduling system, based on Genetic Algorithms to solve the dynamic version of the SMSP. The approach adapts the resolution of the static problem to the dynamic one in which changes may occur continually. A population regenerating mechanism is put forward. This adapts the population to a new population, which increases or decreases according to new job arrivals or cancellations.

In studying the dynamic problem we are particularly concerned with evaluating the adequacy of the GA approach to real world scheduling problems. For that, it was proposed an approach for the resolution of the Job-Shop Scheduling Problem (JSSP) in dynamic environments. These developments will permit to construct multi-criteria analysis procedures that may be used in Decision Support Systems for scheduling.

References

[1] J. Blazewicz, K.H. Ecker, E. Pesch, G. Smith, and J. Weglarz. *Scheduling Computer and Manufacturing Processes*. Springer, New York, 2nd edition, 2001.

[2] P. Brucker. *Scheduling Algorithms*. Springer, New York, 3rd edition, 2001.

[3] C.Dimopoulos and Ali M. S. Zalzala. Recent developments in evolutionary computation for manufacturing optimization: Problems, solutions and comparisons. *IEEE Trans. on Evol. Computation*, 4(2):93–113, 2000.

[4] Lawrence Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[5] Ana Madureira. Meta-heuristics for the Single Machine Scheduling Total Weighted Tardiness Problem. In *Proc. ISATP'99 (IEEE Int. Symposium on Assembly and Task Planning)*, pages 405–410. IEEE, 1999.

[6] Ana Madureira, Carlos Ramos, and Sílvio do Carmo Silva. A Genetic Algorithm for the Dynamic Single Machine Scheduling Problem. In *Advances in Networked Enterprises - BASYS'2000* , pages 315–323. Kluwer Academic Publishers, 2000.

[7] S.French. *Sequencing and Scheduling - An Introduction to the Mathematics of Job-Shop*. John Wiley & Sons, 1982.

[8] S.F. Smith. OPIS:A Methodology and Architecture for Reactive Scheduling, journal = Zweben and Fox (Eds)Intelligent Scheduling. pages 29–66, 1994.