Genetic algorithm to solve a multi-objective scheduling problem

Gilles Mouzon*, Mehmet B. Yildirim

Department of Industrial and Manufacturing Engineering, College of Engineering

Abstract. Energy is expensive and a potential way to reduce energy consumption may be through using intelligent scheduling techniques. In this paper, we propose a genetic algorithm to solve a single machine scheduling problem where the objectives are minimizing total completion time and energy consumption of manufacturing equipment when all of the processing time and release dates of the job are known. This problem has several applications including scheduling in manufacturing industry, energy minimization in computers, cell phones, sensors, etc. Different fitness functions and parameters depending on the problem's characteristics were tested using a design of experiment approach. The proposed methodology generates several pareto optimal solutions. A decision maker can select one of these solutions using Analytical Hierarchical Process.

1. Introduction

The development of the industrial world has emphasized the need for the use of algorithms capable of scheduling as well as reducing energy consumption. Mass consumption has increased energy usage by 1500% in the last century. Regarding this issue and the limitation of petroleum on earth, engineers started to be more concerned about energy and some research has been going on trying to reduce its consumption.

In industry, the completion time of a product is important because as soon as the customer orders it, he wants it in his hand. Completion time is the point in time when a machine has finished processing a part. The shortest processing time rule solves the problem when the release dates of the jobs are zero [1]. Otherwise, if all of the jobs are not available at the beginning, the total completion time problem cannot be solved in polynomial time, i.e., is NP-hard [2,3]. But only minimizing completion time can lead to a lot of energy usage especially if we keep the machine on all the time. Thus, it is also important to have energy concerns and the problem is to make a good compromise between completion time and energy usage. Minimization of energy consumption has been an area of interest especially in computer and embedded electronic systems to expand battery life [4,5,6,7], but the objective has not been combined with scheduling objectives.

This problem is a complication of the single objective problem of minimizing total completion time with release dates which is NP-hard so heuristics are the only way to find a good solution in a reasonable amount of time. Moreover, in multi-objectives optimization, there is not only one solution to the problem but a set of solution called pareto optimal set. The decision maker can base its decision on this set and combine our proposed solution with AHP techniques to get the right solution. This paper solves a single machine scheduling problem of minimizing total completion time and energy usage by the use of a genetic algorithm or more precisely a multi-objective genetic algorithm (MOGA).

The genetic algorithm has been developed using the exact job release dates and the processing time. We consider setup that takes time and energy. The set of non-dominated solutions found by the GA can help the controller or the decision maker to make the final decision. The solution of the problem will give the state of the machine during the shift (on or off) as well as the time when the job should start to be processed.

2. Experiment, Results, Discussion, and Significance

The first step in multi-objectives optimization is to write the optimization problem with its objectives and constraints (Figure 1).

The first set of constraints (3) simply means that a job cannot be processed before it is released. The second set (4) of constraints is related to energy. If a job j precedes a job k and if we perform a setup then $y_{jk}$ is equal to the corresponding idle energy. The last set of constraints (5) means that two jobs cannot be processed at the same time. The first objective (1) is to minimize the sum of the energy consumption minus the processing energy which is a constant for all solutions of the problem. The second objective (2) is to minimize the total completion time.
The second step is to develop the MOGA. An initial population of size k is randomly generated (or generated using an heuristic). Using this generation and operation of crossovers and mutations, we generate a new generation of size k. Only the best individuals according to a fitness function from those two generations will survive and compose a population of size k. The process is performed until we reach a stopping criteria (number of generations).

Each chromosome may have more than one solution depending on the completion time of each job. We use a series of LP to find the set of pareto optimal solutions corresponding to a chromosome. Then a new generation of n individuals is composed using crossovers and mutations. Only the n best individuals from the two populations will survive and continue the evolution process to reach a set of near-optimal or optimal solutions. The algorithm keeps going until it reaches the stopping criteria. Also, the non-dominated solutions are kept in a vector as we move from generation to generation. When the algorithm reaches its stopping criteria, the solutions contained in the non-dominated set form the near-optimal pareto front.

In the 1970's, Saaty developed a technique to choose the best alternative among a set of non-dominated solutions according to different criteria and sub-criteria called Analytic Hierarchy Process (AHP). The decision maker has to structure the problem in a hierarchy of criteria and sub-criteria. He also has to provide pairwise comparison of the criteria to determine the value of the weight that will permit him to evaluate each alternative. Also pairwise comparisons between the alternatives are needed in order to evaluate each of them. The pairwise comparisons have to be done using a ratio scale (Saaty provided a ratio scale).

In the case of our problem, the obvious two first criteria are the total completion time and the energy consumption. To those criteria we can add the wear out of the machine which can be measured by the number of setups and the maximum completion time.

In order to evaluate the MOGA, we need to use a measure of performance. In this article, the measure used is the area covered by the non-dominated set of solutions.

In figure 2, we can see the pareto front we got with our algorithm on a 50 jobs problem with the optimized parameters.

3. Conclusions

The multi-objectives optimization problem is a NP-hard problem. Finding the exact solution takes a lot of time. As a result, we developed a MOGA to solve this problem in a reasonable amount of time. The description of the MOGA includes the definition of all the steps such as mutations, crossovers, fitness function, etc. We have illustrated how to get a set of non-dominated solutions using our heuristic combine with the LP and how our algorithm is different from the one usually present in multi-objective optimization. This makeup of the MOGA presented the first to solve the energy-total completion time problem and the first algorithm with chromosomes which can represent more than one solution.

4. Acknowledgements

The authors acknowledge the National Science Foundation for support of the research reported in the article: DMI-NSF 0537839