Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

SIMULATION OPTIMIZATION FOR PROCESS SCHEDULING THROUGH SIMULATED ANNEALING

Alex Cave
Saeid Nahavandi
Abbas Kouzani

School of Engineering Technology
Deakin University
Geelong, Victoria 3216, AUSTRALIA

ABSTRACT

This paper presents a simulation optimization of a real scheduling problem in industry, simulated annealing is introduced for this purpose. Investigation is performed into the practicality of using simulated annealing to produce high quality schedules. Results on the solution quality and computational effort show the inherent properties of the simulated annealing. It is shown that when using this method, high quality schedules can be produced within reasonable time constraints.

1 INTRODUCTION

In manufacturing one of the purposes of an optimization algorithm is to generate practical schedules that may be implemented on the shopfloor as described by Hopp (1996). Huchison (1990) provides classification of scheduling methods and suggests that offline heuristic methods will in the future provide high quality scheduling solutions. The execution time is a constraint of the algorithm that is particularly important as the optimization process will be repeated many times for different production schedules. Investigation has been performed into the feasibility of using simulated annealing (SA) for this purpose.

Simulated annealing has the potential to generate high quality solutions for combinatorial optimization problems such as the job shop-scheduling problem (JSSP). Modern computation technique reviews for scheduling problems have been undertaken by ; Choi (2000), Jain and Meeran (1998), Jones and Rabelo (1998). These reviews show that for large optimization problems local search techniques such as SA are popular, matured optimization methods that provide high quality solutions.

Further work on scheduling and local search has been carried out but Aarts et al (1994) and Vassens (1995) these reviews provide comparison of different local search techniques for the JSSP. Studies show that simulated annealing requires more computation effort to achieve quality solutions then other methods with the advantage of being generally applicable.

Other researches have investigated customized flow-shop problems using SA these including; Parthasarathy and Rajendran (1998), Raine et al (1999), Ruiz-Torres et al (1997). These studies have practical implications for the manufacturing industry as simulation and computation becomes more affordable.

There is a practical limit to the amount of computational effort that can be applied to this problem. Investigation into the SA algorithm is constrained within the limit of 2×10^6 non-stochastic simulations of one week of shop floor production. This simulation limit and corresponding time limit was defined through consultation with production planners. SA is similar to many heuristic algorithms with regards to execution time, the longer the search is performed the higher the probability of obtaining high grade solutions thus a trade off must be applied. Some comparisons are applied between the SA and previous work using iterative improvement.

This paper is organized as follows. In Section 2, scheduling optimization problem is described. Then, representation of solution is discussed in Section 3. In Section 4, simulated annealing algorithm is briefly explained. Next, cost mapping function is described in Section 5. In Section 6, neighborhood selection function is presented. Then, experimental results are discussed in Section 7. Finally, the concluding remarks are provided in Section 8.

2 THE SCHEDULING OPTIMIZATION PROBLEM

The device to be scheduled is a plastic rotational molding machine that may be generalized into a flow shop or more specifically a power and free system. This instance of the power and free system has three jigs, processes and buffers in the arrangement shown in Figure 1. The maximum buffer length is one, and the jigs cannot overtake one another on the power and free tracks. For every different part type, there is a specific die that needs to be installed on the

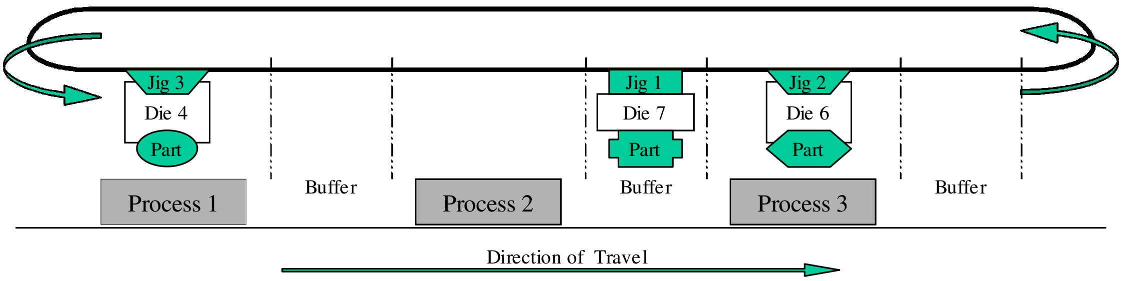

Figure 1: Power and Free Schematic

jig prior to processing. In this instance, there is only one die available of each part type. There are three jigs in the system named J1, J2 and J3. J1 is of a different type to J2 and J3. This places further constrains on the system as only one type of die fits on J1, and a different type of die fits on J2 and J3. In order to produce a part, the part must go through three processing stations P1, P2 and P3. Each processing station requires a different quantity of processing time depending on the type of part being produced. The cycle time of the system is defined as the amount of time required for the jig 1 to perform a lap of the system. This time is measured from the start of first process. Installing a die onto a jig requires a fixed amount of setup time that is dependent on the type of the die. The die setup time may be considered as an extension of the P1 process time for the first part produced using the die.

Production demand is placed on the device in the form of a job list. A job consists of the number of parts of certain type that need to be produced. The cycle time is interdependent between jigs and dies. J1 completes a cycle together with J2 and J3. This interaction cause the make-span of the job list to be altered depending upon the production sequence of the job list. One objective of the algorithm is to search for the sequence with the smallest make-span. However this is not the only objective of the algorithm as other objectives must be taken into account in order to make any solution practical to implement. Each die setup not only requires time but also has a cost in terms of labor. Production planners also require a means to increase the priority of producing a job as production deadlines must be met. These are incorporated into the algorithm through the cost mapping function.

One general measure of the quality of a schedule is device efficiency. This is a measure for the quality of production sequencing. The device efficiency is calculated using a job list that is generated for the schedule to be evaluated. The jobs list is used to calculate the ideal quantity of machine time required to produce the job. The ideal production time is equivalent to the amount of system time it would take to produce the job, if it was the only job running. The actual production time is the amount of time the die was actually in use on the system for a particular schedule, this is calculated from simulation. Using this data the overall efficiency of the device is calculated using the formula given in Equation 1.

$$\begin{aligned} &\text{Eff} = \frac{100}{\text{N}} \times \sum_{N}^{n=1} \frac{\text{Ipt(n)}}{\text{Apt(n)}} = \text{Device Efficiency} \\ &\text{N} = \text{The number of different jobs created.} \\ &\text{Ipt(n)} = \text{The ideal production time of job n} \\ &\text{Apt(n)} = \text{Actual production time for job n} \end{aligned} \tag{1}$$

In these experiments, more than one week of activities is scheduled and the efficiency is measured for the first week. Other common performance measures are make-span and mean tardiness. These measures are restricted to particular jobs or job lists where device efficiency may be applied to incomplete jobs and job lists. Device efficiency may be calculated from any schedule for the device, it allows performance comparison between schedules with dissimilar job lists, manually generated schedules, and historical device data. Distinction needs to be made between the fitness or cost mapping and the device efficiency. The fitness is generated from the cost mapping function and is a continuous variable is optimized by the algorithm. Fitness is a comparative measure of the quality of solutions of the same scheduling problem and takes many factors into consideration. Device efficiency allows comparison between different problems and scheduling methods and is less ambiguous than fitness.

3 REPRESENTATION OF THE SOLUTION

Several investigators have determined empirically, that optimal schedules do not necessarily produce all parts of a particular job in one batch. Also there are times when it is beneficial to not produce any parts at all on one or more machines for a number of process opportunities. To facilitate the skipping of process opportunities, the concept of a null part is introduced. A null part is when an arm is run with an empty die; when this occurs the processing time for a null part is zero. The average cycle time of the system is 127.7 minutes and the production period that is available for scheduling is approximately one week. When this information is taken into account sequencing the machine, one part at a time seems to be a practical proposition. In order to generalize the system for other applications, it may be useful to consider a single part as a batch of arbitrarily set size.

An example scheduling sequence is shown in Figure 2. The algorithm uses the same representation for simulation of the process times. The elements within the collections A1, A2 and A3 represent production of a single part of a part-type that is represented by the numerical value of the element. The permutation of the collections represents the sequence of production for the jigs. The elements of value 0 represents null parts, also shown is the two important concepts of systems cycle and transitional edge.

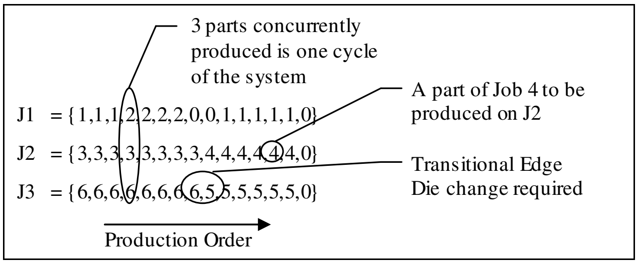

Figure 2: The Part Sequencing Representation

It has been calculated that for a typical week scheduling problem of one week of production, the size of the solution domain is approximately 7.77×10^{110}. The size of the solution domain grows exponentially as a function of the job list. As this growth is faster then any polynomial function and the cost mapping of the algorithm to be optimize is a continuous variable. The problem may be categorized as NP-Hard. This calculation also shows that the solution domain is too large to use exhaustive enumeration with practical time limits and justifies the use of heuristic search algorithms for this problem.

4 SIMULATED ANNEALING

Simulated annealing is a local search process and is analogous to simulation of annealing of solids, van Laarhoven (1987) provides a thorough description of this method. It is based on the idea that making small changes may improve a given solution; if the small change degenerates the solution then it will be accepted with a probability that is related to the size of the degeneration.

For any possible solution, there is a cost mapping $C : F \rightarrow \Re$ which is the function to be optimized. The neighborhood function is the mapping $N : F \rightarrow F'$, which defines for every solution $i \in F$, a neighborhood exists $N(i) \subseteq F$. Each solution in $N(i)$ is called a neighbor of i. The execution of a local search algorithm defines a walk in F such that each solution visited is a neighbor of the previous one. A solution i is called a local optima with respect to the neighborhood function N if $C(i) \leq C(j), \forall j \in N(i)$ or there is no neighbor of i with a cost mapping that is less then $C(i)$.

A cooling schedule proposed by Kirkpatrick et al (1982) is used. The initial value c_0 is set so that 80% of random sampled degenerative transitions are accepted according to Equation 2. The value $k = 0,1,2,....,$ relates to the temperature of the algorithm k is incremented according to the criterion that there is a fixed number *(g)* of failed subsequent transition attempts. The control parameter c_k for $k > 0$ is calculated by $c_{k+1} = \alpha \bullet c_k$ where $\alpha = 0.95$.

By choosing a random transition *j* from the neighborhood of *i* there is an associated difference in the cost mapping $\Delta C_{ij} = C(i) - C(j)$. The probability ($P$) for configuration *j* to be the next accepted configuration in the sequence is 1, if $\Delta Cij < 0$ and is given by the Metropolis criterion proposed by Metropolis et al (1953) is shown in Equation 2 if $\Delta Cij \geq 0$. For a non-homogenous SA algorithm the value c_k is incrementally decreases in magnitude as the algorithm progresses according to the cooling schedule. This decreasing of c_k causes degenerative transition to be accepted with a lower probability as the algorithm progresses and corresponds to a lowering of temperature.

$$P = \exp\left(\frac{-\Delta C_{ij}}{c_k}\right) \tag{2}$$

Limiting the maximum value for k sets the stopping criterion. The maximum values for k and g was set through initial experimentation. High k relates to a very low probability acceptance and is set to a level that is sufficient for the progress of the algorithm to cease. The parameter g influences the length of the Markov chain for each lowering to the algorithm temperature. The length of this chain decreases for each subsequent lowering of temperature. The target maximum of simulations for each run of the algorithm is 2×10^6 and g is set to a threshold so this target is achieved prior to the control parameter k terminating the algorithm.

5 COST MAPPING

It is the objective of the algorithm to maximize the fitness of the schedule subject to the hard constraints that exist within the system. The cost mapping function evaluates the relative fitness of a particular solution or schedule. It is this process that drives that heuristic algorithm towards better solutions. The objectives of the algorithm is a combination of several factors that include the urgency of job production, reduction in the make span of the schedule, reduction of the labor requirement of the system and reduction of inventory. The operation of the machine is continuous not discrete. The priority of production must be considered by the system. This priority is set by the required supply date. Releasing all known jobs into the scheduling algorithm may result in better machine efficiency over a given make span but early production results in greater levels of inventory. Thus the optimum schedule for a given time period is difficult to define with regards to a single variable that is to be a measure of the fitness of the schedule.

As the cost mapping function takes many different objectives into account it is a better measurement of the fitness of a schedule than the efficiency alone for the purposes of schedule optimization. The fitness value of a schedule is a subject measure of device performance as it is influenced by the desires of the operator of the algorithm with respect to the urgency of jobs and is relative to the initially created random solution. In practice the long-term performance of the machine is measured in efficiency and is subject to all schedules being feasible to implement.

6 NEIGHBORHOOD FUNCTION

The neighborhood selection function plays a key role in the overall efficiency of local search algorithms as described by Hao (1996). For SA to work it is required that the entire solution domain that is to be searched is accessible through neighborhood transitions. This requirement causes the size of the neighborhood to be larger than is required for other search methods such as iterative improvement.

The function mapping between solutions of these experiments is achieved by swapping the positions of two parts on the production sequence. This swapping operator is broken down into three operations; primary part selection, secondary part selection and swapping.

The neighborhood function is a form of limited random selection. The only restriction in generation of transitions is that the part found in secondary selection cannot be of the same job type as the part found by primary selection. Typically for this problem the size of this neighborhood function is 4500 possible transitional steps from any given solution.

7 EXPERIMENTS

The objective of the experiments was to test the feasibility of SA based optimization as applied to practical scheduling problems. Three different job lists were generated where each lists represents more than one week of production for the device. The three job-lists named JBL1, JBL2 and JBL3 were selected to represent difficult scheduling problems as encountered on the factory floor and were extracted from historical management data.

Simulation is usually the most computationally expensive component of simulation optimization problems. The number of simulations that can be afforded may be the limiting factor with regards to algorithm selection. The simulation was modeled in C++ using UML. This method of modeling was selected primarily for speed of execution. The device that was investigated is an NC machine where most operation are timed and were modeled deterministically. The manual operations that may not reasonably be represented deterministically have only minor impact upon the overall cycle time. Verification of the model showed that for a planning interval of 1 week the fidelity of the model is acceptable for planning purposes. For this reason process times are modeled in a non-stochastic manner. Non-stochastic modeling eliminates the need for multiple sampling of models and thereby decrease computational effort. Verification of the model was performed by other researchers and the simulation model was is use prior to this optimization work for production schedule testing and resource planning using what if analysis.

The amount of computational effort for each run of the algorithm was set to an approximate limit of 2×10^6 simulations. The experiments are performed on a Pentium III processor running Windows 2000 and each run requires approximately 1.5 hours of computation. Each run was repeated 100 times for each job list in order to generate statistical data.

Data on the machine efficiency is shown as a measure of the general performance of the algorithm. Statistical data collected from cost mapping function is also presented. Cost mapping is not a direct function of efficiency however it servers the purpose of allowing insight into the behavior of the algorithm. Please see Table 1 for statistical data collected from experimental measurements at termination of the algorithm.

Figure 3 shows trace measurements for a typical SA run for JBL2. It shows the decreasing cost of a schedule as the K value increases, or the algorithm progress. The gradient of the K value line is indicative of the length of the Markov chain for each temperature decrement of the algorithm.

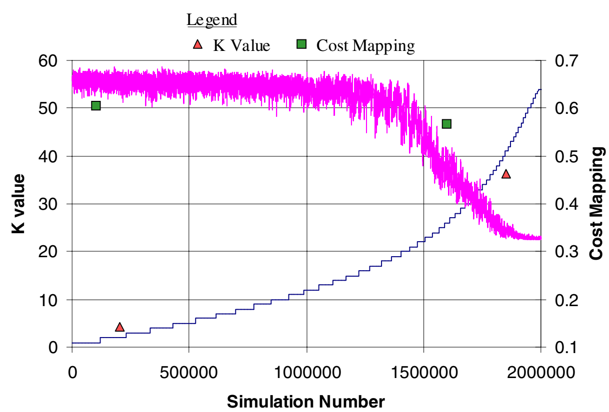

Figure 3: Cost Mapping and K Values vs Simulation Number of Typical Run of JB2

Table 1: Experimental Results

	Efficiency			Cost Mapping		
	Best	Mean	StDev	Best	Mean	StDev
JBL1	75.15	72.825	0.96	0.303	0.3237	0.0063
JBL2	74.456	71.708	1.25	0.300	0.3234	0.0061
JBL3	76.363	74.436	0.94	0.300	0.3434	0.0051

8 CONCLUSIONS

In this paper, algorithmic scheduling optimization was investigated. An example was given on the synthesis of the neighborhood function, solution representation, cost mapping and correct parameter selection for successful implementation of simulated annealing.

The experimental results were compared against average data collected during the operation of the system. The results show that SA can be used to resolve practical scheduling problems. SA produces quality results with a low degree of variance.

Further work tailoring the algorithm to the system to be optimized should yield high quality results with less computational effort.

REFERENCES

Aarts, E. H. L., P. J. M. Van Laarhoven, J. K. Lenstra, N. and L. J. Ulder. 1994. A computational Study of Local Search Algorithms for Job Shop Scheduling. *ORSA Journal on Computing* 5:118-125.

Choi, C. W. 2000. An Overview of Solving Job Shop Scheduling Problem by Local Search Techniques. National University of Singapore, Singapore.

Hao, J. and R. Dorne. 1996. Empirical Studies of Heuristic Local Search for Constraint Solving. In *Lecture Notes in Computer Science*: Springer

Hopp, W. J. and M. L. Spearman. 1996. *Factory Physics*: Irwin.McGraw-Hill.

Huchison, J. 1990. Current and Future Issues Concerning FMS Scheduling. *OMEGA International Journal of Management Science* 19 (6):529-537.

Jain, A. and S. Meeran. 1998. A state of the art review of job-shop scheduling Techniques. University of Dundee, Dundee, Scotland.

Jones, A. and L. C. Rabelo. 1998. Survey of Job Scheduling Techniques. Gaithersburg, MD: National Institute of Standards and Technology.

Kirkpatrick, S., C. D. Gelatt Jr. and M. P. Vecchi. 1982. Optimization by Simulated Annealing. *IBM Research Report RC 9355*.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller. 1953. Equation of State Calculations by Fast Computing Machines. *J. of Chem. Physics* 21.

Parthasarathy, S. and C. Rajendran. 1998. Scheduling to Minimize Mean Tardiness and Weighted Mean Tardiness in Flowshop and Flowline-based Manufacturing Cell. *Computers and Industrial Engineering* 34 (2): 531-546

Raine, F., C. Raczy and A. Artiba. 1999. Hybrid Auto-Adaptable Simulated Annealing Base Heuristics. *Computers and Industrial Eng.* 37 (1-2):277-280

Ruiz-Torres, A., E. Ensore and R. Barton. 1997. Simulated Annealing Heuristic for the Average Flow-Time and teh Number of Tardy Jobs Bi-Criteria Identical Parallel Machine Problem. In *21st International Conference on Computers and Industrial Engineering* 33 (1-2):257-260

Vaessens, R. J. H., E. H. L.Aarts and J. K. Lenstra. 1995. Job Shop Scheduling by Local Search. Eindhoeven: University of Technology.

van Laarhoven, P. M., E. H. Aarts. 1987. *Simulated Annealing: Theory and Applications*: D. Reidel Publishing Co.

AUTHOR BIOGRAPHIES

ALEX CAVE is a researcher at Deakin University, in Geelong, Australia. He received his engineering degree from Deakin University in the area of Mechatronics. His research interests include simulation modeling, optimization algorithms, and process optimization. His email address is <apcave@deakin.edu.au>.

SAEID NAHAVANDI received BSc (Hons), MSc and a PhD in Automation from Durham University (UK). In 1991 he joined Massey University (NZ) where he taught and led research in robotics. In1998 he became an Associate Professor at Deakin University and the leader for the Intelligent Systems research group and also Manager for the Cooperative Research Center for CAST Metals Manufacturing. Dr. Nahavandi has published over 100 reviewed papers and delivered several invited plenary lectures at international conferences. He is the recipient of four international awards, best paper award at the World Automation Congress (USA) and the Young Engineer of the Year award. Dr. Nahavandi is the founder of the World Manufacturing Congress series and the Autonomous Intelligent Systems Congress series. Dr. Nahavandi is a Fellow of IEAust and IEE.

ABBAS KOUZANI is a lecturer with the School of Engineering and Technology of Deakin University, Australia. He received his PhD degree from Flinders University and his MEngSc degree from the University of Adelaide, both in Australia. Abbas has taught in electronic and computer engineering areas. He has authored or coauthored more than 50 publications, and is involved in more than ten industry and university research grants. Abbas is member of the IEEE and the ACM. His current research interests include visual information processing, intelligent systems, and microelectronics.