116

Flow Shop Scheduling Using Evolutionary Computations

(Hokkaido university) ¤ Ibrahim Kebbe, Hiroshi Yokoi, Keiji Suzuki, Yukinori Kakazu

Abstract: Flow Shop Scheduling has been addressed by many methods of the deterministic nature which proved obsoleteness due to solution period length and hardness with respect to problem size. Evolutionary computation seems to be a proper candidate to surpass this obstacle as it can handle the size but on the expense of determination of optimal solutions to some extent. This paper is intended to formulate the problem according to two Evolutionary Computational methods, namely, Vibrational Potential Method (VPM) and Genetic Algorithms (GA). Solutions of some case studies are considered through which evaluation of the effectiveness of both methods is sought and discussed.

1. Introduction

Flow Shop Scheduling is a manufacturing problem that has been treated by many methods for solution. The problem's big picture is the fact the tasks are to pass over all machines in the same sequence (preferably optimal) from which originates the relative simplicity of the problem as compared to the Job Shop Scheduling.

Eventually, the nature of the solution depends largely on the size of the problem in terms of machines and tasks. Deterministic methods of the like of Branch and Bound [1, 2] can successfully handle small-sized problems otherwise difficulty associated with large solution time arises. This calls for methods of the like of Neural Networks, Genetic Algorithms (GA), and Vibrational Potential Method (VPM). In fact, although the aforementioned methods might not provide absolute optimal solutions, especially for large sizes, they tend to offer near-optimal solutions with a satisfactory span of computation time.

2. Genetic Algorithms (GA)

Being an indirect method corresponding to its involvement with the given problem, GA requires special encoding and decoding for the physical problem. The usual approach is to represent the problem space using a binary configuration of the space entities (chromosomes).

In the case of the problem addressed herein, many have attempted a similar binary representation that would only add to the complexity of the problem in terms of information handling. On the other hand, some used the direct non-binary representation of the chromosomes which simplified things but added to the complexity of the GA engine.

Goldberg [3] tackled the problem as well as others [4] using this non-binary representation and treated the special case that would arise when attempting to effect crossovers on the chromosomes. The uniqueness of the individual gene within the chromosome implied that a special process should be followed to ensure the non-repetitiveness of genes. This gave rise to Partially Matched Crossover (PMX) which was first used to solve the blind Traveling Salesman Problem (TSP) [3].

Some earlier works inhibited crossings between non-homologous strings to avoid genes repetitiveness but that was obviously "unnatural" and limited the abilities of GA. Later, many approaches were adopted to avoid this repetitiveness and several chunking methods were attempted (single point, two points, ..) [4].

In this work, the crossover point is single and repetition of genes is avoided by a double sweep for the genes of the swapped sub-strings. In fact, since crossover is affected between two strings, one can anticipate the worst scenario in that the swapped sub-strings would both contain identical genes. The problem arises when these identical genes are not at the same position with respect to the corresponding sub-string. In that case, individual genes of the sub-strings complementary to the swapped sub-strings in each parent are compared to each swapped gene of the corresponding parent: if a match occurs, a pointer is set in the swapped field matching gene flagging the corresponding gene in the other parent. Afterwards, a second comparative scan is affected on the swapped sub-string of the original parent for the value of the gene pointed in the other. If a match occurs then the gene corresponding to it in the other parent is chosen, else, the originally pointed out gene is adopted. This special crossover, along with an appropriate objective function, are added to a simple GA engine to solve for Flow Shop Problems.

3. Vibrational Potential Field

VPM is an evolutionary computational method differing from GA in that the solution space is integrated into a single field of interaction among individual fields representing the solution entities rather than being a competitive sample of that space [5].

Each task and machine are attributed with a potential field characterizing the transfer between combinatorial (physical) and field (computational) media. A common wave function differentiates between the identities of individual fields and, under its guidance, these fields interact among each other producing attraction and repulsion (the amount of which measured by energy level of the field) towards the final state of stability which is the optimal.

4. Objective Problems

Due to space and time limitation, only two representative problems will be dealt with herein. The first is a well-determined, real-life application for a simple line of three machines where six tasks are to be sequentially fed into the line [2] and the solution is sought for the optimal time which is characteristic of the optimal solution(s). The second problem is an artificially-generated one where twenty five tasks are assigned to fifteen machines at random make spans.

5. Results

First using GA, both problems showed characteristic behavior in their solution.

The first one converged at once due to the fact that, being a realistic example, it had many optional solutions as the individual make span of every part on a single machine is relative to the machine function: in our case, the second machine had shorter spans than the other two which gave rise to multiple solutions.

The second problem was pretty elusive to GA as it imposed a highly competitive space due to the randomness of make spans where it becomes hard to single out the optimal solution less one tends to include the total space of population, a process futile in itself as this space is of the order of 1023. This tendency is outlined by the fact that, as population size grows, more optimal solutions arise (figs 1 and 2) and, moreover, matching sequences at the beginning and end of the sub-optimal strings increase correspondingly and are propagated in that manner into higher sizes and the similarity is increased.

By examining fig. 1, one sees clearly the effect of varying parameters of GA where the optimal combination is assigned to using the elite method which propagates the best string onto generations, a pressure factor of unity, and 0.8 crossover ratio. Other crossover ratios are quite redundant and higher pressure ratios decrease optimality.

In fig. 2, as mentioned before, optimality is emphasized in terms of population size. Notice how the improvement in the solution slows down towards larger population sizes, a phenomenon suggesting closeness to optimality.

As for VPM, it converged almost instantaneously for the first problem (P1 in fig. 3) but for the other one it behaved around the solution such that, for lower damping factors and speed increments and higher position increments, performance is improved. This means that slower and thorough search will catch on to local minima better than otherwise. The P2 series in fig. 3 depicts that in the corresponding sequence of the three parameters: damping factor, speed, and position increments.

6. Conclusions

GA has shown better results than VPM on large-scale problems but on the expense of larger execution time. However, VPM has yielded its best results within 5% of those of GA with much less execution time.

Further elaboration on VPM would help tune up its parameters to better suit the Floor Shop Problem. Until then, GA offers a less demanding tool to solve it for valid sub-optimal solutions. Both methods represent a better alternative to deterministic ones.

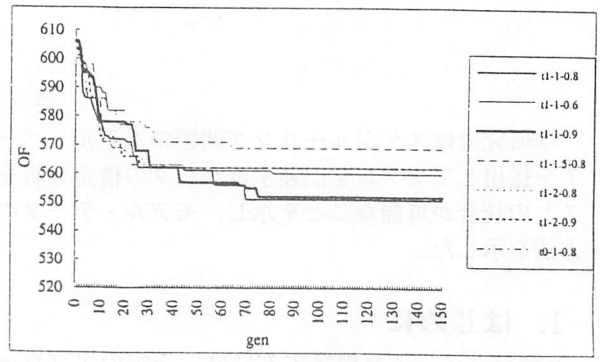

Fig.1 GA: effect of varying parameters for a population of 500

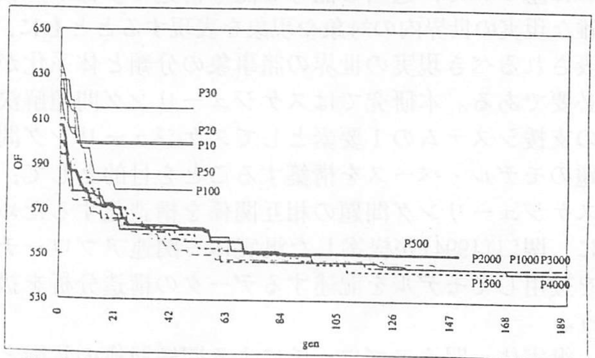

Fig.2 GA: effect of varying population size

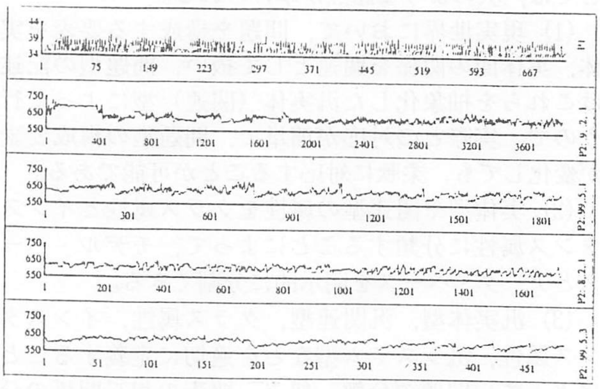

Fig.4 VPM: P1 (top); P2,triple parametric solution runs

References

[1] McMillan, Claude Jr., "Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machine".

[2] Kusiak, Andrew, *"Intelligent Manufacturing Systems"*, Prentice-Hall International Editions Inc., 1990.

[3] Goldberg, David E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company Inc., 1989.

[4] Cleveland, Gary A., and Smith, Stephen F., "Using Genetic Algorithms to Schedule Flow Shop Releases", International Conference on Genetic Algorithms, 1989.

[5] Yokoi, Hiroshi, "Theories and Applications on Vibrating Potential Method", PhD Thesis, Hokkaido University, 1992.