Evolutionary and Genetic Algorithms

THE PRODUCTION SCHEDULING IN ASSEMBLY SYSTEM WITH EVOLUTIONARY ALGORITHM

Galina Setlak

Abstract: In this paper an evolutionary algorithm is proposed for solving the problem of production scheduling in assembly system. The aim of the paper is to investigate a possibility of the application of evolutionary algorithms in the assembly system of a normally functioning enterprise producing household appliances to make the production graphic schedule.

Keywords: Artificial intelligence, flexible assembly systems, evolutionary algorithm, production scheduling.

ACM Classification Keywords: I. Computing methodologies I.1.Symbolic and algebraic manipulation I.1.3.Evaluation strategies I.2.Artificial Intelligence I.2.8.Problem solving Control Methods and Search – Scheduling J.6.Computer Aided Engineering - Computer Aided Manufacturing (CAM).

Conference: The paper is selected from Second International Conference "Intelligent Information and Engineering Systems" INFOS 2009, Varna, Bulgaria, June-July 2009

Introduction

The competition in the world market requires that machine and technological equipment manufactures try new effective tools to maximally shorten the preparation time to start production and meet quality standards. The unusually intensive changes in the market result in the need to realize a growing number of production objectives in the shortest time possible. All this makes the automatization of the assembly process more and more decisive for the competitiveness of a modern enterprise [Sawik, 1996].

The problem of how to optimize planning methods, including production scheduling, has drawn the attention of quite a number of scientists since as early as mid-fifties [Conway, 1967]. The complexity of assembly scheduling and its highly significant effect on the functioning of assembly systems and their production impact necessitate looking for and developing advanced methods and algorithms for solving scheduling related problems. In recent years many research centers have been using artificial intelligence methods including evolutionary algorithms to find out how to optimize production plans.

The aim of the paper is to investigate a possibility of the application of evolutionary algorithms in the assembly hall of a normally functioning enterprise producing household appliances to make the production graphic schedule.

Production scheduling in an assembly system is connected with taking particular decisions about the passage of objects that are being assembled through the system. Here, two basic kinds of scheduling tasks can be distinguished [Sawik, 1996]:

- Initial scheduling of products, which consists in timing the introduction of the successive base parts of different types of assembled products into the system. Decisions on the choice of the product for assembly and the moment when it can leave the factory are taken with regard to operational and tactical plans (assembly hall capacity, preplanned technological routes).

- Scheduling assembly and transport operations involving the assigning of assembly and time limits to particular machines. That kind of scheduling is curbed by the inflow of the particular types of base parts to the system, i.e. the results of the assumed initial schedule.

Thus, the two basic schedules above are closely connected. While making detailed production schedules it is common practice to aim at attaining a specific objective which is later used as an evaluation criterion for the schedule. The chosen targets may be, e.g. time minimization of all the tasks, reduction of delays in meeting buyers' expectations, making full use of the production potential and others.

What are evolutionary algorithms

The basic idea of evolution modeling has always been intriguing for researchers in many fields of science: "Let us replace the process of modeling human race by modeling its evolution".

Evolutionary algorithms are computer-aided problem solution systems. They are based upon the principles that can be observed in the evolution of living organisms [Goldberg, 1989]. The idea of evolutionary algorithms is founded on the processes observed in nature such as the selection of specimens and evolution of species, reproduction mechanism and inheriting characteristics.

Evolutionary algorithms include also such methods as genetic algorithms, evolutionary programming and evolutionary strategies [Goldberg, 1989], [Michalewicz, 1996]. The paper, following the recent trends, makes use of the general term and generally accepted name: evolutionary algorithms (EA). Because of the size limits of the paper the differences between the methods have not been explained here.

The working of evolutionary algorithms can be described in a simple, step-by-step way as follows:

1. <u>Initiation</u>-the formation of the initial population of specimens, which means a random choice of the necessary number of chromosomes (specimens). In such a population each specimen represents an acceptable solution.
2. <u>Evolution of chromosome adaptation</u> in the population-involves calculating the value of the adaptation function (fp) for each chromosome of a particular population. The value of (fp) depends on what kind of problem is being solved.
3. <u>Checking if the termination criterion has been met</u>-that depends upon a particular application of EA. If the criterion has been fulfilled, we pass on to the final step which is <u>extracting</u> the the best chromosome. If it does not happen this way, the next step is selection.
4. <u>Chromosome selection</u>, which consists in choosing the chromosomes which will participate in the creation of offsprings for the new generation. The process follows the principles of natural selection. The chromosomes with the greatest fp value have the best chance to participate in the creation of new offsprings. There are many selection methods [Goldberg, 1989], [Pawlak, 1999], but the simplest and the most popular one is the roulette method as its randomness is like that of a roulette. The probability of picking a particular chromosome is greater with an increase in the fp value. The selection results in creating a parent population whose size equals that of the current population.
5. <u>Application of genetic operators to the selected chromosomes</u>, which leads to the creation of a new population which is made up of the populations of offsprings obtained from the selected population of their parents. In EA two basic genetic operators are applied:
   - crossover,
   - mutation operator.

   The chromosomes of the chosen parents are combined to produce offsprings. The process is also called recombination.

   Mutation is done after crossover and involves the introduction of some random alterations in the chains of descendant chromosomes created before. Like in nature, mutation occurs extremely rarely: the probability of its occurrence of is very small ($0 \le pm \le 0.01$).

6. Evaluation of all the offsprings that form the population.

   Not only the newly obtained descendant chromosomes but also all the others are evaluated.

7. Creation of a new population.

   At all times the offsprings that are the fittest stand the best chance of getting into the new population. Here various methods can be applied [Michalewicz, 1996], [Pawlak, 1999].

   Now we pass on to step 3.

   Steps 4-7 are performed loop wise until the criterion of EA termination has been met.

8. Derivation of the best chromosome.

   If the EA has been stopped, the effect of the algorithm should be derived, i.e. the solution to the problem should be found. The best solution is the chromosome of the greatest fp value.

An example of the evolution algorithm application to assembly process scheduling

The chapter will present the results of the practical application of evolution algorithms to scheduling the production at the vacuum cleaner assembly department of the ZELMER Household Appliances Factory. Currently the factory is producing seven basic models of vacuum cleaners. Their technological designs are different, but the number of modules and the number of the parts that make them up is basically the same. A manual and the assembly instruction materials for the vacuum cleaner which was probably the prototype of one of currently produced models were used for the research.

The following assumptions are made in order to work out the production schedule:

- Work schedule applies to the completion of monthly tasks of the assembly department in the two-shift system.
- Nine tasks are realized in the production process and each of them means putting together a defined number of modules making up a finished product as well as the final assembly of a fixed number of one particular model.

As sub-assembly and complete product assembly operations are regarded as tasks, there are some sequence restrictions between them. Fig 1. shows the basic components singled out in the assembly and disassembly operations of a vacuum cleaner.

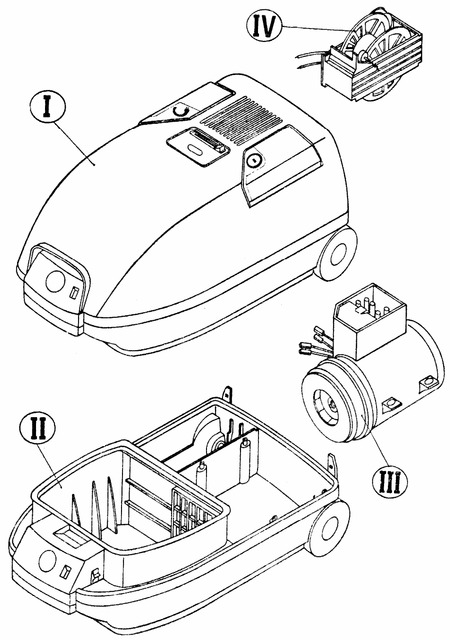

Fig 1. Basic assembly components of a vacuum cleaner [Instruction, 1998].

A finished vacuum cleaner (I) consists of 4 basic subassemblies

- II-complete body consisting of 18 parts-components,
- III-a sucking set composed of 9 elements,
- IV-a complete power cord reel made up of 18 parts,
- Apart from II-IV sets, a complete vacuum cleaner includes 24-26 other pats.

One of the most essential decisions while applying evolutionary modeling methods is specifying the space of the solutions to be searched by the evolutionary algorithm. This is attained through defining the mapping between a point in the solutions space (schedule) and a point in the representation space, i.e. chromosome. There are two known approaches to solving schedule problems with evolutionary algorithms which apply two kinds of representation [Pawlak, 1999]:

- Direct representation-involves using the schedule as the chromosome. Here, the schedule for a particular machine is an arranged list of time limits of starting the operations performed with the machine. The method needs special crossovers to guarantee that the sequence of the operations necessary to carry out the task is not disturbed.
- Indirect representation evolves using the sequence of the tasks to be done as the chromosome. At the chromosome level the tasks have no fixed technological plans, reserves or the beginning and conclusion times. The chromosome is made acceptable by means of a special decoder (schedule making procedures). In the assembly system example under consideration, due to the particular task completion sequence, a simplified variant of the approach presented in [Setlak, 2004] was used where the chromosome is represented by the sequence:

$$(Z_1, s_1)\ (Z_2, s_2)\ \dots\ (Z_n, s_n)$$

where Z_i - i-task, s_i-obligatory arrangement following the sequence of performing i-task.

Thus, sequence limitations between tasks are coded in the chromosome. Fig.2 shows the hierarchical structure of the finished product, which basically determines the sequence of the tasks to be performed. This way, there is also a possibility to code the assembly priorities of particular vacuum cleaner types to meet the buyer's expectations.

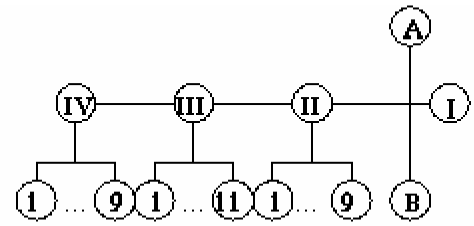

Fig.2. Structure of finished product, arranged to follow he sequence of assembly operations.

In the structure in Fig.2, 1-9 or 1-11 mark the operations performed while putting together a particular sub-assembly described above as A - assembly of sucking pipes involving 4 operations and B - assembly of a small suction nozzle involving one operation. The assembly of the whole vacuum cleaner, marked as I, involves 19 operations that are not shown in the picture.

The assembly of the particular types of the vacuum cleaners, which slightly differ in fittings or subassemblies, follows the same sequence, as shown in Fig.2. In the case of only one type the order is different as it involves one more subassembly to perform one more function. This has been considered by placing one element (sentence) in a chromosome with another number of sub-assemblies and operations.

In the schedule creation process, i.e. arranging the particular elements of the chromosome, a principle was adopted that all the next task operations are fixed in the schedule at the earliest possible dates of their commencement. Thus, assembly operations of sub-assembly IV come first-from IV.1 to IV.9. Then, successively, such operations of sub-assembly III, from III.1 to III.11, which is in agreement with the assembly order of the finished product given in manual [Instruction, 1998]. The next chromosome element are all the II 1-II 9 fitting operations of sub-assembly II, to be finally followed by operations I1.-I. 19, A.1-A.4 and B1 to arrive at the finished product. Consequently, the chromosome will be represented by sequence:

(Z_1,IV_1,1,IV.2,2,…,IV.9,9, III.1,10, III.2,11,…, III.11,21, II.1,22, II.2,23,…,II.9.30, I.1.31,…, I.19,50, A.1,51,…, A.4,54, B,55) (Z_2, IV_2-B_2)… (Z_9, IV-B_9).

For the sake of simplicity detailed operations are not included in the chromosome representation and the exemplifying chromosome is written as:

(Z_7,1)(Z_5,2)(Z_2,3)(Z_9,4)(Z_3,5)(Z_4,6)(Z_1,7)(Z_8,8) (Z_6,9)

In respect of the defined production quota and the two-shift working system (48 shifts per month) it was necessary to divide the production tasks into parties, which increased the number of components in the chromosome up to 37 elements.

The structure of the evolutionary algorithm can be described as follows:

Step 1. The creation of the initial population was carried out by the heuristic method described in [Conway, 1967].

Step 2. Creation of the acceptable chromosome, due to restrictions.

Step 3. Chromosome evaluation. While working on the presented system, one of the basic evaluation criteria, i.e. minimization of the time necessary to finish all the tasks (Cmax) was made use of. Because of this there is a need to transform the evaluation into the adaptation function which will be maximized. Let us rewrite the adaptation function following the formula:

$$F(x) = \frac{\max_x [C_{max}(x)] - C_{max}(x) + \gamma}{\max_x [C_{max}(x)] - \min_x C_{max}(x) + \gamma} \quad (1),$$

where F(x)-value of adaptation function,

$\max_x [C_{max}(x)]$] - maximum importance of completion data in the particular generation,

$C_{max}(x)$-value of completion date for individual x,

$\min_x [C_{max}(x)]$ - minimum time limit value in the particular generation,

γ- coefficient which can perform two functions.

Depending on the situation occurring during the solution of the problem it can take small values from interval (0,1) so that the quotient denominator does not equal zero. In the other case, coefficient y is used for scaling congruence function and leveling the differences between individuals in the population. In such a case the value of y must be correspondingly greater and it is a parameter of the system [Mattfeld, 1996].

Step 4. Memorizing the best chromosome.

Step 5. Chromosome selection was made with a roulette wheel and by the tournament method.

Step 6. Crossover. For that purpose a special method was worked out. It is based on the PMX (Partially Mapped Crossover) method presented in [Uckun, 1993]. The modification involves the introduction of the element of searching extra solution space.

Step 7. Mutation. It did not always occur while solving a problem posed.

Step 8. Repairing effect of chromosomes.

Step 9. Putting the best chromosome in the population.

Step 10. Evaluation of chromosomes.

Step 11. Memorizing the best chromosome.

In the algorithm steps 1-4 are initial ones and made once only, whereas steps 5-11 are performed many times until the condition of evolution termination has been meet.

Results of calculation experiments

A few series of test were made while testing the evolution algorithm for scheduling assembly tasks that is described in the paper. The first test series was performed at the following assumptions:

- population size-50
- crossover by the modified PMX method.
- application of order-based mutation
- chromosome evaluation made with help of adaptation function described by formula (1), the analyzed various at different y coefficient values: y=0, y=0.1
- In each evolution process 500 generations were considered.
- Application of evolution strategy by choosing the best chromosome and putting it in the next generation to replace the randomly chosen one.
- Simulations for various crossover and mutation probability values were carried out. Each evolution process was repeated ten times and the average result was calculated.

  To analyze and estimate the schedules obtained by applying the worked-out evolution algorithm, a heuristic algorithm of active schedule and no-delay schedule were created.

  In the building process of no-delay schedules the priority rules considered were as follows:

  - the shortest operation, i.e. from among the available alternative assembly operations we select the one whose completion time is the shortest (SPT rule [Conway, 1967])
  - the shortest completion time limit rule.

Table 1. shows the results obtained by the methods discussed in the paper.

Table 1. The results obtained by the methods

Method	Evolutionary Algorithm	No-delay schedule	Active schedule
$C_{max}(x)$ (h)	368,2	412,86	434,12
Calculation time	1,45 h	14 sek	15 sek

Conclusion

The results obtained by the evolutionary algorithm method are much better than those by heuristic scheduling or the no-delay method. However, we must notice the relatively long calculation time by the evolutionary algorithm which is 1.45 hours.

Based on the tests carried out, whose results have only partially been presented in the paper, it can be pointed out that evolutionary algorithms are a very effective tool that enables solving complicated practical optimization problems including NP-hard production scheduling problems in assembly systems. An important characteristic of evolutionary algorithms is their simplicity and versatility. Their main drawback is a long calculation time, which, however, is not a serious disadvantage nowadays with advanced computer technology and does not limit their use for searching for almost optimal solutions.

Bibliography

[Conway, 1967] Conway R.W., Maxwell W.L., Miller L.W. Theory of scheduling. Addison-Wesley, Reading, Massachusetts, 1967, 382 str.

[Goldberg, 1989] Goldberg D.E. Genetic algorithms in search. Optimization and Machine Learning, Addison-Wesley P.C., 1989.

[Mattfeld, 1996] Mattfeld D.C. Evolutionary search and the job-shop. Physica - Verlag, Heidelberg, 1996.

[Michalewicz, 1996] Michalewicz Z. Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT, Warszawa, 1996, 432 str..

[Pawlak, 1999] Pawlak M.: Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji, PWN, Warszawa, 1999

[Sawik, 1996] Sawik T.: Planowanie i sterowanie produkcji w elastycznych systemach montazowych, WNT, Warszawa, 1996

[Setlak, 2004] G.Setlak. Harmonogramowanie produkcji na wydziale montażu za pomocą algorytmów ewolucyjnych // Ogólnopolski kwartalnik naukowo-techniczny „Technologia i automatyzacja montażu", 2004, № 2, P..3-9.

[Instruction, 1998] The instruction of disassembly and repairs, the catalogue of teams and the spare parts for vacuum cleaner type 900.0 i 930.0 ZZSD „PREDOM-ZELMER", Rzeszow, 1998.

[Uckun, 1993] Uckun S., Bagchi S., Mijabe Y.: Managing genetic search in Job-shop scheduling //IEEE Expert-Intelligent Systems & Their Applications, Vol.8, 1993, pp.15-24.

Authors' Information

Galina Setlak – D.Sc, Ph.D., Eng., Associate Professor, Rzeszow University of Technology, Department of Computer Science, Str. W. Pola 2 Rzeszow 35-959, and The Bronislaw Markiewicz State School Of Higher Vocational Education in Jaroslaw, Czarneckiego Street 16, Poland, Phone: (48-17)- 86-51-433,
e-mail: gsetlak@prz.edu.pl