Production Planning in Manufacturing/Remanufacturing Environment using Genetic Algorithm

Chihoon Lim
Department of Industrial Engineering, Seoul National University,
San 56-1, Shillim-Dong, Kwanak-Gu, Seoul, Korea, 151-742
+82-2-880-7182
ciga98@gmail.com

Eoksu Sim
Mechatronics & Manufacturing Technology Center, SAMSUNG ELECTRONICS CO., LTD
416, Maetan-3Dong, Yeongtong-Gu, Suwon-City, Gyeonggi-Do, Korea, 443-742
+82-31-200-2207
eoksu.sim@gmail.com

Categories and Subject Descriptors
J.7 [**Computers in other systems**]: *process control*

General Terms
Algorithms, Management, Measurement, Performance

Keywords
remanufacturing, production planning, genetic algorithm

1. Production Planning in Mfg./Rmfg. environment

Because of increasing consumptions, finite resources, and disposal capacities, remanufacturing is receiving more attention [2]. Remanufacturing is the most important process in recovery: is an environmentally and economically sound way to achieve many of the goals of sustainable development; closes the material-use cycle and forms an essentially closed-loop manufacturing system; focuses on value-added recovery, rather than just materials recovery, i.e., recycling. Production planning and scheduling of remanufacturing should be handled differently from that of manufacturing because of high uncertainty of that.

Clegg et. al. proposed a linear programming model for production planning and control in an aggregate manufacturing environment [1]. However, they could not suggest a real method to resolve the problem.

In this study, we propose a new method for production planning in a hybrid manufacturing system. In the production planning, we want to determine the production schedule in mid-term reflecting long term planning from sales as well as operation planning. In each planning period and in each resource, the production quantity and return quantity of used products should be determined. We considered multiple products consisting of multiple product structures.

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

2. Solution Methodology

The kernel of this algorithm schedules parts, subassemblies, products levels ,separately, not in a local optimal solution but in a global optimal solution. The most movable section of the overall schedule is the manufacturing/remanufacturing schedule of a part's level. After one population of GA determines only this section, the disassembly/assembly schedule is determined according to that schedule. Then, the total cost and fitness function value is determined, and through selection, crossover and mutation, the optimal solution is obtained. The overall flow of the algorithm can be described in figure 1. Next, each process of the algorithm is explained, in detail.

First, decision variables that are represented as the chromosomes of GA must be selected. The decision variables for the part level will be encoded as chromosomes. But variables for all planning periods do not have to be encoded because in some periods, workcenters cannot manufacture or remanufacture. In the first two periods and last two periods, part level cannot be remanufactured because of the time for disassembly and assembly. Similarly, part level cannot be manufacturing in the last two periods because assembly is not required.

We generate initial populations. It means that we decide the production level for each variable. Once initial populations are generated, we can calculate the production quantities, number of setups, and the overtime working hours. Overall production quantities are divided to each period along initial populations. The number of setups for each part should be calculated to extract the setup costs. We assume one setup for no production in the earlier period but production in the present period. Overtime working hours are calculated for each period and each workcenter. If the processing time for each workcenter exceeds the capacity of the workcenter, the extra time will be calculated as overtime working hours.

Using remanufacturing schedule for part level, we can calculate process requirements (PRs) of subassemblies. PR means that if we want to remanufacture x item of part p in period t, subassemblies as many as required for part p should have been already disassembled. PR calculation is done by the following formula.

$$pr_{pt} = \max_{p}\left\{\frac{\sum_{i=1}^{t+1} x_{pi}}{sub_p}\right\} - \sum_{j=1}^{t-1} pr_{pj}$$

If the PRs are determined, the optimal disassembly plan of subassemblies can be solved by linear programming. Because LP is used only for subassemblies, many decision variables may not be required.

Similarly, the concept of process provisions (PPs) for an assembly plan of subassemblies can be used. PP means that if one decides to manufacture or remanufacture x item of a part for subassembly p in period t, the number of subassembly p as many as that will be produced in period t+1 can be assembled. PPs are calculated by the following formula.

$$ps_{pt} = \min_{p}\left\{\frac{\sum_{i=1}^{t-1} (x_{pi} + y_{pi})}{sub_p}\right\} - \sum_{j=1}^{t-1} ps_{pj}$$

In the same manner, the optimal assembly plan of subassemblies can be solved by linear programming.

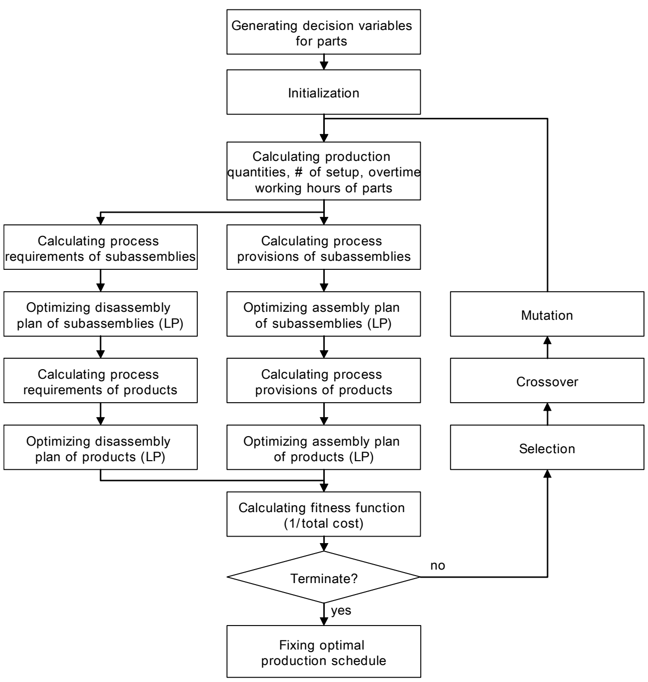

Figure 1. Overall flow of the algorithm

After completing the disassembly and assembly plan in subassembly level, similar process is applied to end product level. Likewise, we can calculate PR and PP and optimize disassembly and assembly plan of the end product level.

Then total costs is calculated. Total costs consist of setup costs of part level and overtime costs for each workcenter. Setup cost is calculated as (the number of setup)*(setup unit cost). Overtime cost is calculated based on calculated overtime working hours.

We use 1/(total cost) for the fitness function. Based on fitness function values of all populations, we certify the termination condition of the algorithm. The termination conditions are two types. One requires that the average of the fitness function values of the present generation is twice that of the first generation. The other requires that the total number of generations equals 100.

If the algorithm doesn't terminate, selection, crossover and mutation of GA evolve the present generation into the next generation. Selection is the process that selects high fitness value populations among the parent populations. Survival probability of each population is normalized by the fitness value.

Crossover is the process that generates child populations from survived parent populations. In this study, 2 parent populations are ready and 2 crossover points are randomly generated. Between two points, parent populations interchange their genes. As high fitness value populations are selected through the selection process, child populations with higher fitness values can be obtained. Mutation is done to randomly selected genes from each population. In 0/1 encoding, mutation changes gene 0 to 1, or 1 to 0. In case of 0/1/2 encoding, 0 to 1, 1 to 2, or 2 to 0.

Through the processes, populations of the next generation are generated. The algorithm returns and repeats the same processes.

3. Computational Experiments

Because this study attempts to solve the problem that has been uninvestigated by any other research, our results could not be compared. Therefore, we investigated the adequacy of the settings for encoding, crossover rate and mutation rate and whether the fitness values of the algorithm improve through further generations.

First, we searched the optimal setting combination for a randomly generated problem set. That is, the gene encoding method 0-1 or 0-1-2, crossover rate 0.1 or 0.3, and mutation rate 0.01 or 0.05. There are 8 setting combinations for 3 factors. We made an experiment on each setting combination and selected the optimal setting combination based on the global maximum fitness values and trend of the maximum fitness value for each generation.

Of all results, we can say that the best combination is 0/1/2 encoding, crossover rate 0.3, and mutation rate 0.01. With these settings, we carried out the second experiment.

Second, we checked if the optimal setting combination obtained from first experiment is applicable for other problem sets. The combination of 0/1/2 encoding, crossover rate 0.3, and mutation rate 0.01 was applied to randomly generated 10 problem sets. Like experiment 1, we observed the global maximum fitness values and trend of maximum fitness value for each generation and etc. Overall, good results were obtained.

4. REFERENCES

[1] Clegg, A.J., Williams, D.J., Uzsoy, R., Production planning for companies with remanufacturing capacity. Proceedings of the 1995 IEEE International Symposium on Electronics and the Environment, Orlando, FL, 186-191

[2] Fleischmann, M., Quantitative models for reverse logistics. Springer-Verlag, Berlin, 2001