▩ 연구논문

A Heuristic for Part Sequencing on a Flexible Machine

- 유연생산기계의 제품 생산 순서 결정을 위한 발견적 기법 -

Song, Chang Yong[1)]
송 창 용

요 지

본 연구에서는 한 유연생산기계에서 생산해야 되는 제품들의 공구교환횟수의 합을 최소로 하는 생산순서를 결정하는 문제를 다룬다. 이 문제를 풀기 위한 방법으로 외판원문제와 관련된 발견적 기법을 적용할 수 있다. 이 때 연속으로 생산해야 될 두 제품사이의 공구교환횟수는 외판원문제에서의 방문해야 될 두 지점사이의 거리에 해당된다. 만약 각 제품이 필요로 하는 공구의 갯수가 공구장착장치의 용량과 같다면 두 제품사이의 공구교환횟수를 정확히 계산할 수 있지만 그렇지 않다면 각 제품이 필요로 하는 공구의 수와 종류가 다르고 제품을 생산하기 전에 공구장착장치에 장착되어 있는 공구의 종류에 따라서 두 제품사이의 공구교환횟수가 달라지므로 정확하게 추정하기는 힘들다. 이러한 공구교환횟수를 추정하는 방법으로 기존의 방법들은 단지 두 제품사이의 생산에 필요한 공구만을 고려하였으나 본 연구에서는 제품생산순서의 전체적인 관점에서 두 제품사이의 공구교환횟수의 상한값을 기초로 추정하는 새로운 방법을 제시한다. 이 새로운 방법의 우수성을 많은 예제를 통하여 기존에 제시된 다른 방법들과 비교하여 보여준다.

1. Introduction

This article deals with a Tool Loading Problem (TLP) on a single flexible machine. A series of n parts of different types have to be successively processed on the machine. Each part requires a subset of tools of m different types, which have to be placed in the tool magazine of the machine before the part can be processed. The tool magazine can accomodate at most c tools, and, in general, the total number of tools required for all part types exceeds the capacity of the magazine. As a result, it is sometimes necessary to change tools between two part types in a sequence. The TLP consists of finding the sequence in which to process the parts and the tools to place on the machine before each part is processed in order to minimize the total number of tool switches.

The problem is prominent when the time needed to change a tool is significant relative to processing time. All tools are kept in a tool crib located close to the machine. Before

1) 탐라대학교 사회과학부

each part is processed, the corresponding tools have to be placed in the tool magazine. If no tools are available in the tool magazine, they must be transferred from the tool crib to the tool magazine. A lot of machine processing time can be wasted on the loading and unloading operations of tools. Since the processing time of each part is sequence independent, we are only concerned with the time associated with tool switches in order to minimize the completion time of all parts [2].

This problem has been previously considered by some authors. Crama *et al.* [1] proved that the TLP is NP-hard, and so suboptimal solution methods are possible. However, once the part sequence is known, optimally loading the tools in the magazine is accomplished by a Keep Tool Needed Soonest (KTNS) policy as discussed by Tang and Denardo [4]. This policy prescribes that when some tools should be removed for tools required by the next part, those tools that are needed the soonest for remaining parts should be kept first in the magazine. In a recent article of Hertz *et al.* [3], they pointed out that though the heuristics for the TLP was studied extensively, all known algorithms based on Traveling Salesman Problem (TSP) were myopic in the sense that they account for interactions of two parts at a time without a global view of the entire part sequence. So they suggest two more adequate 'tooling' distances between two parts. Here we present a new distance on a global view of the entire part sequence in section 2. In section 3, we present computational results showing the relative efficiency of a proposed distance over the distances suggested by Hertz *et al.* [3]. The conclusions follow in section 4.

2. New definition for tooling distance between two parts

The TLP reduces to a TSP with distances $d(i, j)$ between two parts i and j. If all parts use full capacity, the distances can be correctly estimated. However, in general, as not all parts require full capacity, the distances can be not correctly defined [1]. So, Hertz *et al.* [3] considered the five definitions for $d(i, j)$. The first two distances are:

$$d_1(i, j) = c - |T_i \cap T_j|$$

and

$$d_2(i, j) = |T_i \cup T_j| - |T_i \cap T_j|$$

where T_i is the set of tools required by part i.

The first is an upper bound on the number of tool switches between i and j. The next distance:

$$d_3(i, j) = \max\{0, |T_i \cup T_j| - c\}$$

used by Crama *et al.* [1] represents a lower bound the number of tool switches between i and j. The next two distances are:

$$d_4(i, j) = \max\left\{0, |T_i \cup T_j| - \left[\theta \frac{\Lambda(i, j)}{(n-2)|T_i \cup T_j|}\right] c\right\}$$

and

$$d_5(i, j) = \left(\left[\frac{c+1}{c}\right] |T_i \cup T_j| - |T_i \cap T_j|\right) \left(\frac{(n-2)|T_i \cup T_j|}{\max\{\Lambda(i, j), 0.5\}}\right)$$

where $\lambda_k(i, j)$ is the number of parts, apart from i and j, requiring tool $k \in T_i \cup T_j$

$\Lambda(i,\ j)=\sum_{k\in T_i \cup T_j}\lambda_k(i,\ j)$ and θ is parameter in [0, 1]. The above two distances take into account the $c - |T_i|$ tools present in the magazine when going from i to j nor those required by parts following j.

We will present a new distance that improves upon $|T_j \setminus T_i|$, which is a valid upper bound on the number of switches from i to j as suggested by Hertz *et al.* [3]. This improves on $|T_j \setminus T_i|$ by subtracting a quantity, that is, the number of tools required by only j and not i are likely to be required just before i and then kept in the magazine during processing the part i. Hence we define:

$$d_{new}(i,\ j) = |T_j \setminus T_i| - \left(\frac{c - |T_i|}{c}\right)\left(\frac{\Delta(i,\ j)}{n-1}\right)$$

where $\delta_k(i,\ j)$ as the number of parts, apart from i and j, requiring tool $k\in(T_j \setminus T_i)$, and $\Delta(i,\ j)=\sum_{k\in(T_j \setminus T_i)}\delta_k(i,\ j)$. The factor $(c - |T_i|)\ /\ c$ gives a larger value if the size of the magazine is small, i. e., if more tool changes are probable.

3. Computational experiments

In order to investigate the effectiveness of the new distance for the TLP. we generate sixteen types of problem instances as in Crama *et al.* [1] and Hertz *et al.* [3]. Each instance type is characterized by the following parameters, where

n = number of parts
m = number of tools
min = lower bound on the number of tools per part
max = upper bound on the number of tools per part
c = tool magazine capacity.

The various instances types generated are described in Table 1. For each type, 10 instances were randomly generated resulting in a total of 160 instances.

Table 1. Instance types

n	m	min	max	c
10	10	2	4	4, 5, 6, 7
15	20	2	6	6, 8, 10, 12
30	40	5	15	15, 17, 20, 25
40	60	7	20	20, 22, 25, 30

Hertz *et al.* [3] gave several interesting heuristics, but they suggested the use of FI2 when good solutions have to be computed quickly. So we applied FI2 heuristic to compare with six distances defined in Section 2. FI2 successively applies Farthest Insertion Heuristic [5] using each part as a starting point, applies KTNS to each of the n solutions, and selects one with the least number of tool switches among n solutions. Farthest Insertion Heuristic is performed as follows:

Step 1. Start with part i only.

Step 2. Find part p such that $d(i,\ j)$ is maximal and form the subtour $(i,\ p,\ i)$.

Step 3. (Selection) Given a subtour, find part k not in the subtour and part l in the current subtour such that $d(l, k) = \max_j \{ \min_i d(i, j) \}$, where j denotes a part not in the current subtour and i denotes a node in the current subtour.

Step 4. (Insertion) Find the edge (i, j) in the subtour which minimizes $d(i, k) + d(k, j) - d(i, j)$ Insert k between i and j.

Step 5. Go to Step 3 unless all parts are on the tour.

The computational results are summarized in Table 2. The value $\theta = 0.25$to be best was used in d_4. For each distance type, we report averages and (in parentheses) standard deviations over 10 instances in terms of deviation, in percent, of the value of the TLP objective function (number of tool setups, equal to c plus the number of tool changes) over the best value.

Table 2. Comparison of six heuristics

(n, m, min, max, c)	d_1	d_2	d_3	d_4	d_5	d_{new}
(10, 10, 2, 4, 4)	4.18(4.18)	1.67(2.67)	4.27(4.27)	1.67(2.67)	0.83(1.50)	1.00(1.80)
(10, 10, 2, 4, 5)	2.82(3.95)	2.91(4.07)	8.78(3.60)	1.91(3.05)	1.00(1.80)	2.74(3.84)
(10, 10, 2, 4, 6)	2.11(3.38)	1.11(2.00)	5.00(6.00)	1.11(2.00)	1.11(2.00)	1.11(2.00)
(10, 10, 2, 4, 7)	0.00(0.00)	0.00(0.00)	2.11(3.38)	0.00(0.00)	0.00(0.00)	0.00(0.00)
(15, 20, 2, 6, 6)	7.75(3.39)	5.16(3.65)	7.14(3.66)	3.32(2.66)	3.10(2.48)	2.45(2.45)
(15, 20, 2, 6, 8)	5.00(1.77)	3.60(2.16)	15.73(5.80)	2.54(3.05)	0.00(0.00)	1.84(2.20)
(15, 20, 2, 6, 10)	3.54(2.13)	4.00(1.60)	17.67(6.47)	2.03(2.44)	1.01(1.62)	0.95(1.52)
(15, 20, 2, 6, 12)	1.00(1.60)	1.00(1.60)	9.05(5.95)	0.00(0.00)	0.50(0.90)	0.50(0.90)
(35, 40, 5, 15, 15)	6.21(1.71)	2.32(1.27)	3.74(1.90)	0.75(0.75)	1.93(1.03)	0.68(0.82)
(35, 40, 5, 15, 17)	5.65(1.30)	3.11(0.96)	8.21(3.13)	0.85(1.02)	2.72(1.42)	1.66(1.33)
(35, 40, 5, 15, 20)	5.86(2.22)	4.64(2.33)	22.51(3.93)	1.16(0.93)	2.21(1.72)	1.85(1.59)
(35, 40, 5, 15, 25)	5.98(1.39)	3.71(1.59)	26.82(4.34)	1.89(1.56)	1.87(0.98)	1.22(1.22)
(40, 60, 7, 20, 20)	4.92(1.34)	1.81(0.60)	3.28(0.88)	0.47(0.57)	1.21(0.58)	0.23(0.32)
(40, 60, 7, 20, 22)	5.73(1.26)	2.02(1.38)	6.05(2.35)	0.87(0.82)	1.87(1.02)	1.07(0.86)
(40, 60, 7, 20, 25)	5.40(2.27)	3.33(1.38)	15.20(6.16)	1.43(1.39)	1.81(0.99)	0.26(0.52)
(40, 60, 7, 20, 30)	4.36(2.11)	2.85(1.20)	25.05(3.22)	1.52(1.16)	1.33(1.20)	1.23(1.09)
average	4.41(1.60)	2.70(1.12)	11.29(3.22)	1.35(0.69)	1.41(0.71)	1.17(0.60)

It can be seen from this table that of all distance functions, d_3 is clearly the least interesting and on average and standard deviation, d_{new} is better than d_1, d_2, d_4, and d_5. As the size of (n, m, min, max, c) increases, the ranking of solutions delivered by the heuristics seems to become more stable. Namely, d_{new} yields the best results. Next comes d_4 and d_5 and the worst solution is produced by d_3. Also, the size of c is small, i. e., more tool changes are probable, d_{new} performs well in comparison with the other distance functions. It can be explained that d_{new} tends to be good estimate of the number of tool switches. In case of large values for c, every solutions contain fewer tool switches. And so it is difficult to compare with each other.

4. Conclusions

We have suggested a new distance function to guide the search in TSP-based heuristics for a tool loading problem. The distance is based on a global view of the entire part sequence. The performance of this distance is tested with the other distances proposed by Hertz *et al.* [3] on several sets of randomly generated instances. It turns out that the new distance yields solution values that rank among the best available. The further research is to explore the new distance on other TSP-based heuristics.

References

[1] Crama, Y., Kolen, A.W.J., Oerlemans, A.G. and Spieksma, F.C.R.; "Minimizing the Number of Tool Switches on a Flexible Machine," International Journal of Flexible Manufacturing Systems, 6: 33-54, 1994.

[2] Gray, A.E., Seidmann, A. and Stecke, K.E.; "A Synthesis of Decision Models for Tool Management in Automated Manufacturing," Management Science, 39(5): 549-567, 1993.

[3] Hertz, A., Laporte, G., Mittaz, M. and Stecke, K.E.; "Heuristics for Minimizing Tool Switches When Scheduling Part Types on a Flexible Machine," IIE Transactions, 30: 689-694, 1998.

[4] Tang, C. and Denardo, E.V.; "Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches," Operations Research, 36(5): 767-777, 1988.

[5] Lawler, E.L., Lenstra, J.K., Rinnooy Kan A.H.G. and Shmoys, D.B. (eds.); The Traveling Salesman Problem: A Guide Tour of Combinatorial Optimization, John Wiley & Sons, Chichester, U.K., pp. 207-249, 1985.