ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: AUTOMATYKA z.117 Nr kol. 1337

Waldemar KACZMARCZYK
Akademia Górniczo–Hutnicza w Krakowie

PODZIAŁ PARTII UŁATWIAJĄCY ZNALEZIENIE DOBREGO ROZWIĄZANIA ZAGADNIENIA PRZEPŁYWOWEGO

Streszczenie. Podział partii zwiększa obszar rozwiązań dopuszczalnych. Dotychczas był on jednak wykorzystywany głównie do skracania czasu wykonywania jednej danej partii. A przecież istnieje szansa, że pomoże on również uzyskiwać lepsze uszeregowania całkowite wielu partii.

W pracy przedstawiono kilka różnych sposobów podziału partii oraz algorytmy dla zagadnienia przepływowego wykorzystujące z powodzeniem tę koncepcję.

PART SPLITTING THAT FACILITATES FINDING A GOOD SOLUTION OF THE FLOW SHOP PROBLEM

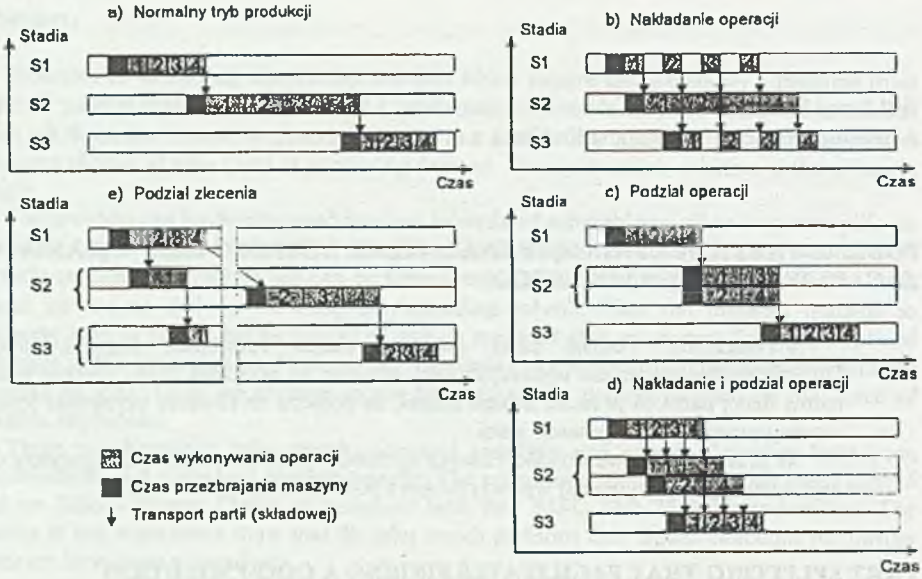
Summary. Part splitting increases the feasible solution set. So far it is used primarily to decrease flow time of one given part. Event though there exist a chance that it helps to improve entire schedules of many parts.

In this paper I present some kinds of part splitting and algorithms for the hybrid flow shop problem using successful this idea.

1. Wprowadzenie

Jeśli w danym systemie produkcyjnym wytwarza się wiele części takiego samego typu, to łączone one są zazwyczaj w partie. Upraszcza to znacząco zarządzanie produkcją, a w szczególności zmniejsza liczbę koniecznych przezbrojeń maszyn. Czasem korzystny może się jednak okazać ponowny, przejściowy podział niektórych partii na mniejsze partie składowe. W niniejszej pracy przedstawiłem rodzaje podziału partii i niektóre powody ich stosowania. Zaproponowałem także algorytmy szeregowania operacji w dyskretnym, przepływowym systemie produkcyjnym z maszynami równoległymi wykorzystujące podział partii do poprawienia jakości harmonogramów.

Każda partia grupująca wiele wyrobów tego samego typu może być szeregowana tak, jakby to była jedna część. W takim przypadku wszystkie wyroby z jednej partii przetwarzane będą w każdym stadium na jednej i tej samej maszynie, będą też zawsze wszystkie razem transportowane pomiędzy stadiami (rys. 1.a). Możliwy jest też jednak *podział wybranej partii* na kilka mniejszych (ang. splitting), ich osobny transport (rys. 1.b), a także przetwarzanie na różnych maszynach w niektórych stadiach (rys. 1.c). Smith [5] mówi w pierwszym przypadku



Rys. 1. Różne sposoby podziału partii

Fig. 1. Different sorts of part splitting

o *szeregowaniu z nakładaniem operacji* (ang. overlap scheduling), a w drugim o *podziale operacji* (ang. operation splitting). Możliwe jest również równoczesne stosowanie obu tych technik (rys. 1.d).

Smith [5] wyróżnia ponadto jeszcze jeden rodzaj podziału partii tzw. *podział zlecenia* (ang. order splitting) różniący się jednak, z punktu widzenia zadania szeregowania, od podziału operacji tylko okolicznościami powstawania. Stosowany jest on bowiem wtedy, gdy nie jest możliwe planowe wykonanie całego zlecenia produkcyjnego, np. z powodu nagłej awarii maszyny. Wtedy ta jego część, która musi zostać wykonana szybko, oddzielana jest od reszty partii i wykonywana niezależnie w pierwszej kolejności. Tak rozumiany podział zlecenia nie ma dla rozważań zawartych w tej pracy żadnego znaczenia.

W pozostałych przypadkach podział partii stosowany jest dla *skrócenia czasu wykonywania całej partii*, tj. czasu od momentu, gdy partia zostanie załadowana do systemu, aż do momentu, gdy zostanie ona cała wykonana. Na ogół jest to równoznaczne ze *zmniejszeniem liczby części znajdujących się wewnątrz systemu*. Smith [5] zwraca uwagę, że szczególnie korzystne może to być w stosunku do bardzo drogich lub bardzo nieporęcznych wyrobów.

Jest jednak jeszcze inne możliwe zastosowanie podziału partii. Przyczynia się on bowiem do *zrównoważenia czasów przetwarzania partii w poszczególnych stadiach*. Ułatwia to zdecydowanie sporządzanie harmonogramów. I takim właśnie wykorzystaniem podziału

partii poświęcona jest ta praca. Więcej informacji o stosowaniu podziału partii znaleźć można m.in. w pracach Wagnera, Ragatza [1], Smitha [5].

Problem szeregowania operacji w dyskretnym, przepływowym systemie produkcyjnym z maszynami równoległymi określany jest zazwyczaj jako *zagadnienie przepływowe z maszynami równoległymi* (ang. hybrid [flexible] flow shop). Wytwarzanych jest w nim na przemian na tych samych maszynach wiele różnych produktów. Składa się on z wielu stadiów, a każde z nich zawierać może wiele równoległych maszyn. Kolejność stadiów jest dla wszystkich produktów zawsze taka sama, choć część z nich może omijać niektóre stadia. Maszyny w każdym stadium mogą być dowolne. Oznacza to, że wykonują one wprawdzie takie same operacje, ale z różnymi prędkościami, a prędkości te mogą być odmienne dla każdej operacji. Rozważany tu model opracowany został dla zakładu chemicznego wytwarzającego płynne produkty opisanego w pracy Kaczmarczyka [2]. Dla zagadnienia przepływowego z maszynami równoległymi i z dowolną liczbą stadiów znanych jest niewiele algorytmów. Więcej informacji znaleźć można m.in. w pracy Kaczmarczyka [3].

Tabela 1

Wielkości charakterystyczne rozpatrywanego systemu produkcyjnego

P	=	$\{1,\ldots,n\}$ – zbiór wszystkich produktów,
L_j	–	zbiór partii składowych produktu j,
S	=	$\{1,\ldots,m\}$ – zbiór wszystkich stadiów produkcyjnych,
S_j	–	zbiór stadiów, w których przetwarzany być musi produkt j,
M_i	=	$\{1,\ldots,m_i\}$ – zbiór wszystkich maszyn w stadium i,
M_{ij}	–	zbiór maszyn ze stadium i zdolnych do przetwarzania produktu j,
p_{ijk}	–	czas przetwarzania produktu j w stadium i na maszynie k.

W niniejszej pracy przyjęto, że wszystkie *partie składowe danego produktu*, tj. powstałe w wyniku podziału *partii pełnej*, są identyczne, tzn. zawierają tyle samo części danego typu. Proponowane algorytmy bazują na pewnej heurystyce iteracyjnej, przedstawionej bliżej w następnym podrozdziale. W pracy przedstawione są również wyniki eksperymentów obliczeniowych obrazujące jakość otrzymanych harmonogramów i wpływ różnych sposobów podziału partii na długość harmonogramów i liczbę potrzebnych przezbrojeń maszyn.

2. Opis heurystyki bazowej

Opis proponowanych w niniejszej pracy metod wygodnie jest rozpocząć od przedstawienia heurystyki bazowej, która nie uwzględnia możliwości łączenia części w partie. Szczegółowy opis jej nieco bardziej złożonej wersji znaleźć można w pracy Kaczmarczyka [3]. Jest ona stosunkowo prostą heurystyką dla *zagadnienia przepływowego z dowolnymi maszynami równoległymi*. W każdej iteracji tej heurystyki harmonogram częściowy, powstały w poprzednich iteracjach, zostaje "wydłużony" przez dodanie na jego końcu planu wykonania

jednego z produktów (jednej części), który nie został w nim jeszcze uwzględniony. Poniżej przedstawiony jest ogólny schemat tego algorytmu i słowny opis niektórych jego kroków.

Ogólny schemat algorytmu bazowego

σ	–	aktualny harmonogram częściowy,
$\sigma\sigma_j$	–	harmonogram σ z dołączonym na jego końcu harmonogramem σ_j produktu j
$\overline{P}$	–	zbiór produktów, nie włączonych jeszcze do harmonogramu σ

Krok 1. Przyjmij $\sigma = \emptyset$, $\overline{P} = P$.

Krok 2. [**Wybór maszyn**] Określ dla każdego produktu $j \in \overline{P}$ najkorzystniejszy harmonogram σ_j^* na końcu harmonogramu σ: $\sigma_j^* = \arg\operatorname{opt}_{\sigma_j}(\sigma\sigma_j)$.

Krok 3. [**Wybór produktu**] Znajdź produkt j^*, który najlepiej daje się dołączyć do harmonogramu σ: $j^* = \arg\operatorname{opt}_{j \in \overline{P}}(\sigma\sigma_j^*)$.

Krok 4. Dołącz harmonogram $\sigma_{j^*}^*$ wybranego produktu j^* do harmonogramu σ: $\sigma = \sigma\sigma_{j^*}^*$, $\overline{P} = \overline{P} \setminus \{j^*\}$.

Krok 5. Jeśli $\overline{P} = \emptyset$, to STOP, jeśli nie, to wróć do kroku 2.

W algorytmie tym zadanie ogólne rozbite zostało na dwa podproblemy. Pierwszym z nich jest określenie najkorzystniejszego harmonogramu dla pojedynczego produktu na końcu aktualnego harmonogramu częściowego. Zadanie to sprowadza się do wyboru jednej maszyny w każdym ze stadiów. Uwzględniane są przy tym leksykograficznie dwa kryteria:

M1. Przetwarzanie produktu należy zakończyć jak najwcześniej.

M2. Powinny powstać jak najmniejsze dodatkowe przestoje maszyn.

Liczba wszystkich możliwych kombinacji maszyn może być jednak bardzo duża. Dlatego też zadanie wyboru maszyn zostało potraktowane jak wieloetapowy problem decyzyjny. Zastosowany algorytm wyboru maszyn (*krok 2.*) dokonuje przeglądu możliwych rozwiązań i wyboru jednego z nich według schematu znanego z programowania dynamicznego. Z każdą jego iteracją (etapem) związany jest wybór maszyny w jednym stadium.

Po określeniu "optymalnych" harmonogramów dla każdego z pozostałych do zaplanowania produktów należy wybrać jeden z nich (*krok 3.*) i dołączyć go do aktualnego harmonogramu częściowego. Wybrać należy produkt, który najlepiej spełnia poniższe kryteria:

P1. Przyczynia się do powstania najmniejszych dodatkowych przestojów maszyn.

P2. Obciąża najwcześniej dostępne maszyny.

Ocena produktu równa jest ważonej sumie ocen obu tych kryteriów we wszystkich stadiach.

3. Opis algorytmów z podziałem partii

Wszystkie cztery przedstawione poniżej algorytmy są rozszerzeniem opisanej powyżej heurystyki. Rozszerzenie to polega na zastąpieniu pojedynczych części przez całe partie części tego samego typu (produktu). Tak więc, aby otrzymać ogólny schemat tych algorytmów, wystarczyłoby w heurystyce bazowej słowo *produkt* zastąpić słowami *pełna partia produktu*. Poszczególne algorytmy różnią się jedynie sposobem traktowania partii składowych.

Najprostsze do opisania są heurystyki pierwsza i druga. W *heurystyce 1.* nie uwzględnia się w żaden sposób możliwości podziału partii. Natomiast w *heurystyce 2.* uwzględnia się możliwość podziału partii, ale partie składowe traktowane są jak całkowicie niezależne produkty. Tak więc algorytmy te są dokładnie takie same jak heurystyka bazowa, pewnej zmianie ulegają jedynie dane wejściowe. W *heurystyce 1.* za produkt z heurystki bazowej przyjmowana jest pełna partia, natomiast w *heurystyce 2.* jak osobne produkty traktowane są partie składowe.

Heurystyka 3. uwzględnia już możliwość podziału partii, ale wszystkie partie składowe przetwarzane być muszą w każdym stadium na tej samej maszynie (rys. 1.b). Dla tej heurystyki pewnej modyfikacji ulec musi krok 2 heurystyki bazowej:

Krok 2. **[Wybór maszyn]** Określ dla każdego produktu $j \in \overline{P}$ najkorzystniejszy harmonogram σ_j^* na końcu harmonogramu σ:

a) Określ dla pierwszej partii składowej produktu j najkorzystniejszy harmonogram σ_{j1}^* na końcu harmonogramu σ: $\sigma_{j1}^* = \arg\operatorname{opt}_{\sigma_{j1}}(\sigma\sigma_{j1})$.

b) Określ kolejno dla wszystkich pozostałych partii składowych $p \in L_j \setminus \{1\}$ produktu $j \in \overline{P}$ ich harmonogramy σ_{jp}^* na końcu harmonogramu $\sigma\sigma_{j1}\ldots\sigma_{jp-1}$, tak by przetwarzane one były na tych samych maszynach co pierwsza partia składowa.

Tak więc w rozumieniu heurystyki bazowej teraz w kroku 2. planowane jest przetwarzanie kolejno wielu „produktów". W ogólnym schemacie algorytmu w krokach 3. i 4. przez harmonogram produktu j rozumieć należy teraz sumę harmonogramów partii składowych tego produktu $\sigma_j^* = \sigma_{j1}^*\sigma_{j2}^*\ldots\sigma_{jL_j}^*$.

Heurystyka 4. jest dalszym rozwinięciem heurystyki trzeciej. Uwzględnia ona mianowicie możliwość przetwarzania partii składowych w niektórych stadiach na różnych maszynach (rys. 1.c). Względem heurystyki 3. dalszej modyfikacji ulec musi krok 2. Teraz dopuszczalne będzie wybieranie dowolnych maszyn we wszystkich stadiach prócz jednego wybieranego każdorazowo w kroku drugim dla każdego z produktów osobno.

Zmianie ulega tu też sposób agregacji kryteriów *M1* i *M2*, w celu otrzymania ogólnej funkcji celu przy wyborze maszyn. Dotychczas kryteria te uwzględniane były w sposób leksykograficzny, teraz tworzona będzie ważona suma ocen obu tych kryteriów we wszystkich wcześniejszych stadiach.

Celem tych zmian jest ograniczenie liczby maszyn angażowanych w przetwarzanie wszystkich partii składowych. W stadiach, w których czasy przetwarzania są krótkie, minimalizacja przestojów wymusi na ogół wybór maszyny wykorzystanej przez pierwszą partię składową. Natomiast w stadiach wolniejszych minimalizacja terminu wykonania wymusi sięgnięcie również po inne maszyny z tych stadiów.

Krok 2. **[Wybór maszyn]** Określ dla każdego produktu $j \in \overline{P}$ najkorzystniejszy harmonogram σ_j^* na końcu harmonogramu σ:

a) Określ dla pierwszej partii składowej produktu j najkorzystniejszy harmonogram σ_{j1}^* na końcu harmonogramu σ: $\sigma_{j1}^* = \arg\operatorname{opt}_{\sigma_{j1}}(\sigma\sigma_{j1})$.

b) Znajdź stadium i^* z najmniejszym czasem przetwarzania pierwszej partii składowej produktu j: $i^* = \arg\min_{i \in S_j} p_{ik_{ij}^*j}$, gdzie k_{ij1}^* oznacza maszynę wybraną w stadium i dla pierwszej partii składowej produktu j.

c) Określ kolejno dla wszystkich pozostałych partii składowych $p \in L_j \setminus \{1\}$ produktu $j \in \overline{P}$ ich harmonogramy σ_{jp}^* na końcu harmonogramu $\sigma\sigma_{j1}\ldots\sigma_{jp-1}$, tak by w stadium i^* przetwarzane one były na tej samej maszynie co pierwsza partia składowa.

4. Eksperymenty obliczeniowe i omówienie ich wyników

Za pomocą przedstawionych powyżej algorytmów wykonano szereg eksperymentów obliczeniowych zarówno dla przypadku z identycznymi maszynami równoległymi w stadiach, jak i dla przypadku z maszynami dowolnymi. Obliczenia wykonano przyjmując na przemian n=20 lub 40 oraz $\max|L_j| = 3$ lub 5. W ten sposób obliczono harmonogramy dla czterech grup zadań. Każda z nich składała się z 20 przykładów i została scharakteryzowana w tabeli 2.

Część danych wejściowych wygenerowana została losowo. Wykorzystano w tym celu rozkład równomierny. I tak, czasy wykonywania operacji p_{ikj} w stadiach nieparzystych wygenerowane zostały losowo, ale z uwzględnieniem pewnej korelacji pomiędzy prędkościami wykonywania wszystkich operacji danego produktu, a także pomiędzy prędkościami wykonywania wszystkich operacji na danej maszynie. Najpierw wygenerowano liczby A_j z przedziału $[1,10]$ i liczby B_{ik} z przedziału $[1, 3*|M_i|]$. Następnie wygenerowano czasy operacji p_{ikj} z przedziału $[A_j * B_{ik}, 3 * A_j * B_{ik}]$. W rozwiązanych przykładach stadia parzyste pełnią rolę buforów, a wszystkie produkty musiały przejść przez wszystkie stadia. Liczbę partii składowych wygenerowano dla każdego z produktów z osobna z przedziału $[1, \max|L_j|]$.

W tabeli 2. przedstawiono wyniki eksperymentów dla n=20, $\max|L_j| = 5$ oraz identycznych maszyn równoległych we wszystkich stadiach.

$\bar{u}$ – oznacza *dla heurystyki 1.* średni stosunek długości uzyskanego harmonogram do minimalnego obciążenia najbardziej obciążonego stadium. Obliczenie tego obciążenia sprowadza się do określenia "optymalnego" przyporządkowania operacji do maszyn w każdym

ze stadiów i wybrania tego największego. W tym celu wykorzystano heurystykę LPT dla przypadku maszyn identycznych oraz heurystykę ECT wraz z dwoma heurystykami ulepszającymi opisanymi w pracy Hariri, Pottsa [4] dla przypadku maszyn dowolnych.

$\bar{u}$ – oznacza *dla pozostałych heurystyk* średni stosunek długości harmonogramu uzyskanego daną heurystyką do długości harmonogramu uzyskanego heurystyką 1.

x – oznacza *dla heurystyki 1.* stosunek liczby przezbrojeń maszyn we wszystkich stadiach do minimalnej, teoretycznej liczby tych przezbrojeń obliczonej jako $\sum_{i \in S}(n-|M_i|)$

x – oznacza dla pozostałych heurystyk stosunek liczby przezbrojeń potrzebnych po zastosowaniu danej heurystyki do liczby przezbrojeń otrzymanych w heurystyce 1.

$\bar{t}$ – to średni czas obliczeń podany w sekundach.

Wszystkie obliczenia wykonano na komputerze PC/486 SX.

Wyniki obliczeń dla pozostałych przypadków zostaną scharakteryzowane jedynie ogólnie. I tak można stwierdzić, że $\bar{u}$ dla heurystyki 1. przyjmuje małe wartości, gdy średnia liczba operacji przypadająca na jedną maszynę jest stosunkowo duża, największe są też wtedy korzyści z podziału partii. Dla n=20 uzyskano względną poprawę długości harmonogramu, średnio aż o 20%, podczas gdy dla n=40 jedynie o 11%.

Tabela 2

Wyniki obliczeń

l p.	$m_1,\ldots,m_m$	Heurystyka 1.			Heurystyka 2.			Heurystyka 3.			Heurystyka 4.		
		$\bar{u}$	x	$\bar{t}$	$\bar{u}$	x	$\bar{t}$	$\bar{u}$	x	$\bar{t}$	$\bar{u}$	x	$\bar{t}$
1	3,3,3,3,3	1.51	1.90	1.06	0.79	1.77	4.47	0.84	1.08	1.41	0.81	1.38	1.57
2	3,6,6,6,3	1.75	1.96	1.62	0.69	2.23	4.99	0.90	1.26	2.53	0.71	1.73	3.38
3	6,6,3,6,6	2.10	2.01	1.66	0.59	1.82	4.06	0.73	1.15	2.10	0.61	1.53	2.42
4	6,6,6,6,6	2.25	2.05	2.53	0.55	2.15	7.14	0.66	1.20	3.30	0.61	1.62	4.09
Średnie:		1.90	1.98	1.72	0.66	1.99	5.17	0.78	1.17	2.34	0.69	1.57	2.86

Podział partii ma też o wiele większe znaczenie dla przypadku z identycznymi maszynami równoległymi w stadiach, dla którego poprawa wyniosła średnio 19%, niż dla przypadku z maszynami dowolnymi, dla którego poprawa wyniosła jedynie 12%.

Duże znaczenie ma,rzecz jasna, maksymalna liczba partii składowych. Podniesienie tej liczby z 3 na 5 pozwala uzyskać podwojenie poprawy do 18%. Zmienia się przy tym również oczywiście liczba potrzebnych przezbrojeń. Dla $\max|L_j|=5$ i heurystyki 4. przyrost liczby przezbrojeń względem heurystyki 1. jest przeciętnie trzy razy większy niż dla heurystyki 3., a dla $\max|L_j|=3$ tylko dwa razy.

5. Podsumowanie

W niniejszej pracy przedstawione są heurystyki dla zagadnienia przepływowego z dowolnymi maszynami równoległymi uwzględniające możliwość dzielenia partii poszczególnych produktów na mniejsze partie składowe, ich osobny transport oraz przetwarzanie na różnych maszynach.

Algorytm bazowy tych wszystkich heurystyk jest iteracyjną, jednoprzejściową heurystyką. W każdej iteracji tego algorytmu istniejący już częściowy harmonogram wydłużony zostaje przez dodanie do niego na jego końcu planu wytwarzania wszystkich części z jednej partii. Do określenia "optymalnego" harmonogramu pierwszej partii składowej wykorzystywany jest schemat programowania dynamicznego.

Przedstawione wyniki eksperymentów obliczeniowych pokazują, że uwzględnienie podziału partii może wydatnie skrócić długość generowanych harmonogramów. Wpływ ten jest największy, gdy średnia liczba operacji przypadająca na jedną maszynę jest stosunkowo duża, a maszyny w stadiach są identyczne.

LITERATURA

1. Wagner B. J., Ragatz G. L.: The impact of lot splitting on due date performance, Journal of Operational Management, vol. 12, 1994, pp.13-25.
2. Kaczmarczyk W.: Sterowanie operatywne w przepływowym systemie produkcyjnym. Zeszyty Naukowe Politechniki Śląskiej, Seria: Automatyka, z. 115, 1994, s. 35-45
3. Kaczmarczyk W.: Metoda szeregowania operacji w przepływowym systemie produkcyjnym z dowolnymi maszynami równoległymi. W: Zarządzanie w przemyśle: teoria i praktyka podejmowania decyzji, Wydział Zarządzania AGH, Kraków 1995, str. 16-25
4. Hariri A. M. A., Potts C. N.: Heuristics for scheduling unrelated parallel machines. Computers Ops Res. Vol. 18, No. 3, 1991, pp 323-331.
5. Smith S.B.: Computer–based production and inventory control, Prentice Hall, Englewood Cliffs, New Yersey, 1989.

Recenzent: Dr hab.inż. Jan Kałuski, prof. Pol.Śl.

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

In most production systems parts of the same product type are united to lots. The reason is that the number of setup times can by decreased this way. Such lots are manufactured as it were one part. All parts that belong to one lot are together transported and manufactured on the same machines.

But sometimes it is very useful to divide these lots in smaller sublots, independent transporting and manufacturing on different machines in some production stages. This

technique is cold part splitting. It is usually used to decrease the flow time of expensive or bulky products

In this paper is presented a scheduling method which use part splitting to improve operation schedules in a multistage production system. This system is made of a certain number of processing stages and each stage has a few parallel machines. Proposed method is a part–by–part heuristic based on priority rules and an approximation to dynamic programming. During every iteration a complete processing schedule is determined for one lot of parts.

There are proposed algorithm versions for different types of lot splitting. Results of computational experiments are reported. It show the quality of the schedules and the influence of types of lot splitting on the schedule.