

Piotr PIESIK, Jan WĘGLARZ
Politechnika Poznańska

WIELOKRYTERIALNE WSPOMAGANIE PROJEKTOWANIA ELASTYCZNYCH SYSTEMÓW PRODUKCYJNYCH Z WYKORZYSTANIEM TECHNIK SYMULACYJNYCH

Streszczenie. W pracy przedstawiono moduł programowy służący do wielokryterialnego wspomaganie decyzji projektowych dotyczących elastycznych systemów produkcyjnych (ESP). Moduł ten spełnia rolę interfejsu pomiędzy symulatorem ESP a zaimplementowaną metodą wielokryterialnego wspomaganie decyzji. Program pozwala użytkownikowi na wprowadzanie interesujących go kryteriów, wybiera zbiór rozwiązań niezdominowanych i wspomaga decydena w wyborze rozwiązania kompromisowego.

INTERACTIVE DECISION SUPPORT FOR DESIGN OF FLEXIBLE MANUFACTURING SYSTEMS USING SIMULATION TECHNIQUES

Summary. A software package is presented which is designed for multicriteria decision support for design of Flexible Manufacturing Systems. This software acts as an interface between a simulator of FMS and implemented multicriteria decision support methods. It enables the user to enter desired criteria, chooses a set of nondominated solutions, and finally supports the decision maker to choose a compromise solution.

1. Wstęp

Elastyczne systemy produkcyjne (ESP) od kilkunastu lat pozostają w kręgu zainteresowań naukowców i inżynierów z różnych dyscyplin. Głównym wymaganiem, jakie stawia się takim systemom, jest możliwość produkowania szerokiej gamy wysokiej jakości produktów w małych partiach i małym nakładem kosztów. Takie wymagania powodują, że projektowanie i sterowanie takich systemów jest bardzo złożonym zadaniem optymalizacyjnym. Zwykle problem ten dzieli się na cztery podproblemy: projektowanie, planowanie, szeregowanie i sterowanie [1]. Często zdarza się, że ESP nie jest tak elastyczny i efektywny, jak tego oczekiwano w założeniach jego inwestorów i projektantów. Jednym z powodów tego stanu rzeczy jest fakt, że na etapie projektowania systemu trudno jest przewidzieć faktyczne warunki pracy tego systemu i wymagania dotyczące różnorodności

produkowanych elementów. Z kolei decyzje podjęte na etapie projektowania mają kapitalny wpływ na właściwości systemu i z reguły nie mogą być skorygowane na etapach późniejszych. Jednym z możliwych sposobów polepszenia jakości projektu jest użycie metod symulacyjnych w celu zbadania zachowania się systemu w różnych warunkach [9].

W niniejszej pracy wykorzystujemy program symulacyjny SARA [5], który może być stosowany w każdym etapie projektowania i sterowania ESP. Program ten może ponadto symulować różne systemy produkcyjne poprzez zmianę jedynie danych wejściowych, bez zmiany kodu programu. W porównaniu z programami symulacyjnymi proponowanymi w literaturze [2,3,6] SARA pozwala łatwiej zdefiniować konfigurację systemu, umożliwia użycie różnych algorytmów szeregowania i ma bardziej przyjazny interfejs użytkownika.

Biorąc pod uwagę wielokryterialną naturę problemów projektowania i sterowania ESP, proponujemy specjalny moduł do interakcyjnego wielokryterialnego wspomaganie decyzji z wykorzystaniem programu SARA. Moduł ten jest interfejsem pomiędzy wybraną interakcyjną metodą wspomaganie decyzji a symulatorem, który pracuje tutaj jako generator rozwiązań. Interfejs ten jest maksymalnie elastyczny. Korzystając z niego, decydent wprowadza kryteria i wybiera parametry, które należy optymalizować, a użytkownik może nawet zaprogramować własną metodę wspomaganie decyzji i podłączyć ją do symulatora.

Zarówno program symulacyjny, jak i moduł wspomaganie decyzji zostały zaimplementowane w środowisku graficznym X Window System, pod kontrolą systemu operacyjnego Unix.

W rozdziale 2 zawarto krótki opis programu symulacyjnego oraz jego zastosowania na wszystkich etapach projektowania i działania ESP. W rozdziale 3 opisano moduł wspomaganie decyzji. Pracę zakończono komentarzami i wskazówkami dotyczącymi dalszych badań w rozdziale 4.

2. Opis programu symulacyjnego

2.1. Moduły składowe programu

Program składa się z następujących modułów:

- moduł definiowania ESP,
- moduł definiowania zadań dla ESP,
- moduł sprawdzający poprawność danych,
- symulator,

- moduł obróbki danych wyjściowych.

Wymienione moduły są napisane w języku obiektowym C++ i połączone w jedną aplikację pracującą pod X Window System. Poniżej krótko scharakteryzujemy poszczególne moduły.

Moduł definiowania ESP służy do wprowadzania, edycji i zapamiętywania danych opisujących projektowany system. Na dane te składają się:

- typy maszyn (nazwa, pojemność głowicy, liczba buforów wejściowych i wyjściowych),
- stanowiska (nazwa, typ, odległość od stacji za- i wyładowczej, zamontowane narzędzia),
- palety (nazwa (typ), pojemność (liczba detali), liczba dostępnych palet danego typu),
- uchwyty (nazwa (typ), liczba dostępnych uchwytów danego typu),
- narzędzia (nazwa (typ), liczba dostępnych narzędzi danego typu),
- inne zasoby,
- wózki (liczba, prędkość),
- transportery narzędzi (liczba, prędkość),
- stacje za- i wyładowcze (liczba, czas operacji),
- możliwe połączenia transportowe pomiędzy maszynami (maszyna 1, maszyna 2, odległość).

Moduł definiowania zadań dla ESP służy do wprowadzania, edycji i zapamiętywania danych opisujących zadania (tzn. detale do obróbki). Na dane te składają się:

- nazwa (typ) detalu,
- wielkość partii do jednoczesnej obróbki,
- wymagany typ palety,
- operacje (numer kolejnej operacji, nazwa operacji, czas potrzebny na wykonanie, wymagany typ uchwytu, alternatywne maszyny, na których operacja może być wykonana, narzędzia potrzebne do wykonania operacji).

Moduł badania poprawności danych sprawdza kolejno, czy spełnione są następujące warunki:

- istnieje choć jedna maszyna,
- istnieje choć jedna paleta,
- istnieje choć jeden uchwyt,
- istnieją jakiegokolwiek narzędzia,
- zdefiniowano detale do obróbki,

- zaden detal nie wymaga nie istniejącej palety,
- istnieje choć jedna operacja dla każdego detalu,
- żadna operacja nie wymaga nie istniejącego uchwytu bądź narzędzia,
- istnieją maszyny potrzebne do wykonania wszystkich operacji,
- żadna wielkość partii do równoczesnej obróbki nie przekracza pojemności odpowiedniej palety,
- podłączono odpowiedni algorytm sterujący,
- marszruty technologiczne odpowiadają możliwościom systemu transportowego.

Program ma dwa (wybierane przez użytkownika) tryby reagowania na błędy wykryte w czasie testowania poprawności danych:

- zatrzymuje się na pierwszym wykrytym błędzie,
- sprawdza całość, po czym pokazuje błędy po kolei na liście.

W obu trybach możliwe jest automatyczne wejście w to miejsce w danych, w którym wystąpił błąd.

Symulator służy do uruchomienia symulacji zdefiniowanego systemu produkcyjnego.

W tym celu potrzebne jest (oprócz zdefiniowania systemu) także zdefiniowanie zadań oraz podłączenie algorytmu sterującego. Podczas symulacji generowane są (jeśli użytkownik włączył odpowiednią opcję) awarie (proces o rozkładzie normalnym z parametrami zdefiniowanymi przez użytkownika).

Symulacja zrealizowana jest przy użyciu metody przeglądania zdarzeń, przy czym jest tu ona dostosowana do obiektowej filozofii programu. Poszczególne elementy systemu są reprezentowane za pomocą odpowiadającej im hierarchii obiektów, co znacznie ułatwia ich organizację. Program główny (będący również obiektem) komunikuje się z nimi za pomocą komunikatów (message). Jest to metoda znana i zalecana w programowaniu obiektowym. Każdy rodzaj elementów systemu (maszyny, wózki, palety itd.) jest reprezentowany przez odpowiednią klasę (typ) obiektu. Tworzone są odpowiednie podklasy do reprezentacji np. maszyn czy zasobów określonego typu. Odpowiednie obiekty (instancje) reprezentują konkretne maszyny, wózki transportowe i zasoby (tzn. każdy pojedynczy element systemu jest reprezentowany przez instancję obiektu odpowiedniej klasy). Każdy obiekt reaguje na dwa rodzaje komunikatów: polecenia (np. dla maszyny załadowanie detalu z bufora na stół roboczy jest poleceniem) oraz zapytania (np. maszynę można "zapytać", czy ma zajęty bufor, czy jest gotowa do rozpoczęcia wykonywania zadania itp.). Wszystkie dane na temat stanu (zarówno aktualnego, jak i "historii" od początku obróbki) każdego elementu systemu są pamiętane w

polach odpowiadającego mu obiektu (instancji). Zatem po zakończeniu symulacji możliwe jest przeprowadzenie analizy całego przebiegu.

Moduł obróbki danych wyjściowych służy do przetworzenia danych ("historii") pamiętanych w poszczególnych obiektach. Wstępne przetworzenie danych polega jedynie na zebraniu informacji z ww. obiektów. Oprócz tego możliwe jest zapamiętywanie danych w pliku oraz generowanie raportów tekstowych i wykresów graficznych. Następnym etapem jest analiza porównawcza wybranych przebiegów symulatora. Przebiegi te mogą być wykonane w różnym czasie (dlatego potrzebne jest wspomniane wyżej zapamiętywanie danych w pliku) albo równoległe przez dwie instancje programu symulującego (należy pamiętać, że program jest przeznaczony do uruchomienia w wielozadaniowym systemie operacyjnym). Możliwe jest też zaprogramowanie wielu przebiegów symulacji ze zmieniającym się zadaniem parametrem (np. w celu zbadania wpływu liczby palet na pracę systemu). Wówczas najodpowiedniejszym sposobem przedstawienia wyników jest wykreślenie wydajności systemu (lub wykorzystania zasobów) w funkcji tego parametru.

2.2. Zastosowanie programu SARA w projektowaniu i sterowaniu ESP

Przedstawiony program symulacyjny może być używany jako narzędzie wspomaganie decyzji na wszystkich etapach projektowania i sterowania ESP. Pozwala on porównywać różne rozwiązania, dając decydentowi charakterystykę pracy systemu dla każdego wariantu.

Na etapie projektowania systemu program pozwala łatwo przetestować i porównać różne konfiguracje systemu. Przykładowo, można wprowadzić jako dane liczbę i typy maszyn, a następnie wykonać symulację dla losowo generowanych zbiorów zadań.

Na etapie planowania stosowanie metod analitycznych jest często niemożliwe lub daje niejednoznaczne wyniki. Dzięki symulatorowi możliwe jest porównanie wydajności systemu przy różnych wariantach decyzji dotyczących obróbki, w tym:

- tworzenia partii do jednoczesnej obróbki,
- tworzenia grup maszyn równoległych,
- przydzielania operacji do maszyn,
- przydzielanie palet i uchwytów do poszczególnych zadań.

Na etapie szeregowania i sterowania symulator może służyć do porównywania przebiegów systemu z różnymi algorytmami szeregowania w celu znalezienia najlepszego

algorytmu. Ponadto może być wykorzystywany do monitorowania systemu i testowania decyzji podejmowanych w wypadku awarii lub nagłej zmiany asortymentu, co może nastąpić w przypadku, gdy do systemu przybywa zadanie o wyższym priorytecie.

3. Interakcyjny moduł wspomaganie decyzji

Moduł ten jest zaprojektowany w celu stworzenia programowego interfejsu pomiędzy symulatorem a interakcyjną metodą wspomaganie decyzji. Korzystając z tego modułu, użytkownik może wprowadzić różne kryteria, w tym m.in.:

- czas zakończenia zbioru zadań C_{max} ,
- maksymalne opóźnienie zadania L_{max} ,
- średni czas przepływu zadania F ,
- wydajność produkcji,
- liczba maszyn,
- liczba wózków,
- liczba zasobów (narzędzi, uchwytów, palet),
- wykorzystanie maszyn, wózków i zasobów.

Dla każdego z kryteriów podaje się akceptowalny przedział.

Użytkownik może wprowadzić typy detali do obróbki, a dla każdego typu zdefiniować następujące dane: wielkość partii do jednoczesnej obróbki, wymagany typ palety, operacje (nazwa, czas, typ uchwytu, alternatywne maszyny, wymagane narzędzia). Podczas procesu symulacji każda z tych wartości może być losowo zmieniana w zakresie podanym przez użytkownika.

Komunikacja pomiędzy tym modułem a symulatorem jest realizowana poprzez wymianę odpowiednich plików tekstowych, zawierających informacje o konfiguracji systemu, detalach oraz ostatecznie wynikach procesu symulacji. Wszystkie pliki mogą być modyfikowane bezpośrednio przez użytkownika za pomocą wbudowanego edytora. W plikach zawarte są następujące dane:

- konfiguracja systemu:
 - maszyny (typy maszyn i liczba maszyn danego typu),
 - system transportowy (liczba i typy wózków oraz graf połączeń),
 - zasoby (narzędzia, palety, uchwyty),
- dane na temat detali do obróbki:

- typy detali,
- operacje na każdym typie detalu (z grafem ograniczeń kolejnościowych).

Biorąc pod uwagę wymienione wcześniej kryteria, moduł wspomagania decyzji generuje zbiór rozwiązań dopuszczalnych. Każde rozwiązanie jest konkretną konfiguracją ESP, która potrafi obrabiać wymagane detal zgodnie z podanymi warunkami. Rozwiązania są generowane w następujący sposób:

- liczba maszyn, wózków i uchwytów każdego typu jest zmieniana o 1,
- połączenia transportowe są generowane stosownie do wybranego procesu technologicznego,
- liczba innych zasobów (narzędzi, palet) jest zmieniana o 1 lub więcej, w zależności od wybranego zakresu zmienności każdego zasobu oraz wymaganej precyzji; aby uniknąć zmniejszenia precyzji, po pierwszym przebiegu można wykonać drugi ze zmniejszonym zakresem zmienności,
- w zależności od wybranych kryteriów, typu systemu i procesu technologicznego, program wybiera odpowiedni algorytm z posiadanej bazy algorytmów.

Wszystkie wygenerowane w ten sposób rozwiązania są wartościowane z użyciem symulatora, a następnie spośród nich wybiera się rozwiązania niezdominowane, które są dalej brane pod uwagę w interakcyjnej procedurze wspomagania decyzji.

Do opisanego modułu można podłączać różne metody wspomagania decyzji. Użytkownik może zdefiniować format pliku potrzebnego do dostarczenia niezbędnych danych dla programu będącego implementacją odpowiedniej metody. Moduł obsługuje zarówno pliki tekstowe, jak i binarne. Pomimo znacznej elastyczności tego modułu przewidujemy, że dla wielu zewnętrznych programów będzie potrzebna również postać źródłowa, aby zapewnić poprawną automatyczną wymianę danych.

Jako przykład zastosowano metodę proponowaną w [7]. Procedura interakcyjna jest tu realizowana w trzech krokach.

Step 1. (Krok startowy). Na zbiorze rozwiązań niezdominowanych konstruujemy tablicę wypłat Z o wymiarze $k \times k$. Element z_{ij} macierzy Z to wartość kryterium i dla rozwiązania najlepszego z punktu widzenia kryterium j . Przekątna Z definiuje rozwiązanie idealne, które z reguły nie jest dopuszczalne. Wektor złożony z najgorszych wartości z_{ij} dla danego kryterium tworzy 'nadir'.

Step 2. (Krok obliczeniowy). Znajdujemy rozwiązanie niezdominowane będące najbliższym idealnym w sensie funkcji skalaryzującej (powiększona ważona norma Czebyszewa).

Step 3. (Krok decyzyjny). Rozwiązanie znalezione w kroku 2 prezentujemy decydentowi. Jeśli decydent uzna je za satysfakcjonujące dla wszystkich kryteriów, to zatrzymujemy się, w przeciwnym razie pytamy decydenta, które kryterium można pogorszyć i o ile, aby zyskać na innych kryteriach. To pogorszenie jest przetwarzane na funkcję kary, która zostaje dołączona do funkcji skalaryzującej, i wracamy do kroku 2.

5. Podsumowanie

W niniejszej pracy przedstawiono interakcyjny, wielokryterialny system wspomaganie decyzji dla elastycznych systemów produkcyjnych. System ten składa się z symulatora oraz interfejsu do podłączenia metody wspomaganie decyzji. Korzystając z tego systemu, użytkownik może stosować istniejące lub tworzyć własne algorytmy szeregowania i metody wspomaganie decyzji. System może wspomagać wielokryterialne decyzje w projektowaniu, planowaniu, szeregowaniu i sterowaniu ESP.

Obecnie zaimplementowane są tylko podstawowe algorytmy szeregowania. Planujemy utworzenie bazy algorytmów zawierającej także nowe algorytmy dające jednocześnie uszeregowanie maszyn wraz z odpowiadającym mu dopuszczalnym uszeregowaniem transportowym. Planujemy także zaadaptować inną procedurę interakcyjną bazującą na słabszej informacji preferencyjnej.

LITERATURA

1. Błażewicz J., Ecker K., Schmidt G., Węglarz J.: Scheduling in Computer and Manufacturing Systems. Springer Verlag, Berlin 1993.
2. Carrie A.S., Adhami E.: Introducing FMSs by Simulation. Working Paper, University of Strathclyde, 1984.
3. Hurrion R.D. (ed.): Simulation. Applications in Manufacturing. Springer Verlag, Berlin 1986.
4. Józefowska J., Piesik P., Węglarz J.: RUNSIM - a Tool for Simulation of Flexible Manufacturing Systems. Proceedings of the Twelfth European Meeting on Cybernetics and System Research, Wiedeń 1994, pp. 1237-1244.
5. Laskowski C.: Symulator elastycznych systemów produkcyjnych (praca magisterska). Instytut Informatyki Politechniki Poznańskiej, Poznań 1996.

6. Ravi T., Lashkarl R.S., Dutta S.P.: Selection of Scheduling Rules in FMSs - a Simulation Approach. *The International Journal of Advanced Manufacturing Technology*, 6, 1991, pp.246-262.
7. Słowiński R., Soniewicki B., Węlarz J.: DSS for multiobjective project scheduling. *European Journal of Operational Research* 79, pp. 220-229 (1994).
8. Stecke K.E.: Design, Planning , Scheduling and Control Problems of Flexible Manufacturing Systems. *Annals of Opns. Res.*, vol.3, Baltzer, Basel 1985.
9. Suri R., Dille J.W.: A Technique for On-line Sensitivity Analysis of Flexible Manufacturing Systems. *Annals of Opns. Res.*, vol.3, Baltzer, Basel 1985.

Recenzent: Prof. dr hab. inż. Ryszard Gessing

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

Flexible Manufacturing Systems (FMSs) have remained in the field of interest of engineers and researchers of various disciplines for over a decade. The main requirement addressed towards these systems: ability to produce a wide range of high quality products in small batches at a low cost, makes their design and control a difficult optimisation problem. Usually this problem is decomposed into four subproblems, i.e. design, planning, scheduling and control, corresponding to consecutive stages in the FMS's design and operation. In many cases FMSs are not as flexible and efficient as they were expected to be by the investors and designers. One of the reasons is that it is extremely difficult to predict actual demand and product mix at the stage of the system design. Decisions made at this stage have a great impact on the system performance and usually can not be corrected by decisions made at the next stages. One of the possibilities to improve the quality of the system design is to use simulation tools to test the system under various operating conditions.

In this paper we use a simulation program SARA which can be applied at all stages of the design and control of a FMS. Moreover, it can simulate many various production systems by changing only the input data, not the code itself. In comparison with simulation programs proposed in the literature, SARA enables easier definition of a system configuration (as input data), introduction of new scheduling algorithms which may be used during the simulation run, and has a more user-friendly interface.

With regard to the multicriteria nature of the design and control problems in the FMS, we propose a special module for interactive multicriteria decision support. In fact, this is a software interface between an existing interactive method and the simulator which works as a solving module. This interface is designed to be as flexible as possible. Using the interface, the user can even create his own decision support methods and connect it to the simulator. With this module, the user can enter the criteria and choose desired parameters to optimize.

The simulation program SARA has been implemented in the Unix X Window System environment, using the C++ language.