ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: ATOMATYKA z.119 Nr kol. 1339

Jan WĘGLARZ, Jarosław NABRZYSKI
Politechnika Poznańska

SYSTEM EKSPERCKI Z WIELOKRYTERIALNYM ALGORYTMEM TABU DLA PROBLEMÓW ROZDZIAŁU ZASOBÓW

Streszczenie. W pracy przedstawiono szkieletowy system ekspercki dla obszernej klasy deterministyczych problemów rozdziału zasobów. Klasa ta jest scharakteryzowana przez następujące główne cechy: wiele rodzajów zasobów wszystkich kategorii, wiele sposobów wykonywania każdego zadania i wiele kryteriów optymalności. Opisano przede wszystkim równoległy, wielokryterialny algorytm przeszukiwania Tabu znajdujący uszeregowania niezdominowane ze względu na wszystkie kryteria jednocześnie oraz jego interakcję z systemem eksperckim.

EXPERT SYSTEM WITH MULTICRITERIA TABU SEARCH FOR MULTIOBJECTIVE PROJECT SCHEDULING

Summary. We present a decision support system for multiobjective project scheduling problems under multiple category resource constraints. It handles quite a general class of nonpreemptive scheduling problems with renewable, nonrenewable and doubly-constrained resources, multiple performing modes of activities, precedence constraints in the form of an activity network and multiple project performance criteria of time and cost type. The system uses a parallel, multicriteria Tabu Search (TS) method for solving the above class of problems. We concentrate ourselves on the interactions between TS algorithm and a knowledge base of the expert system, containing knowledge that concerns different aspects of TS technology, resources and schedules.

1. Sformułowanie problemu

Projektowany szkieletowy system ekspercki ma wspomagać podejmowanie decyzji dla obszernej klasy problemów rozdziału zasobów, scharakteryzowanej przez następujące, główne cechy:

- wiele rodzajów zasobów dla każdej z następujących kategorii: odnawialne, nieodnawialne i podwójnie ograniczone oraz dyskretne i ciągłe,
- wiele sposobów wykonywania każdego zadania w wypadku dyskretnych żądań zasobowych (w wypadku żądań ciągłych nie będziemy mówili o sposobach, których jest nieprzeliczalnie wiele, ale o przedziałach, z których można dowolnie przydzielać zasoby),

- wiele, w ogólności konfliktowych, kryteriów oceny uszeregowań.

Chcemy ponadto, by system wspomagał konstrukcję zarówno uszeregowań statycznych (*predictive schedules*), jak i dynamicznych (*reactive* lub *real time schedules*). Oczywiście, w miarę pełna ocena wielokryterialna będzie dotyczyła uszeregowań statycznych. Chcemy jednak wykorzystać wiedzę dotyczącą konstrukcji i oceny tych uszeregowań przy podejmowaniu decyzji odnośnie do uszeregowań dynamicznych.

Zadowalające rozwiązanie tak postawionego zadania wymaga spełnienia przez projektowany system szeregu postulatów, z których najważniejsze to:

- możliwość korzystania z wiedzy wielu ekspertów,
- możliwość stosowania różnych metod reprezentacji wiedzy i różnych mechanizmów wnioskowania,
- możliwość strukturalizacji i hierarchizacji opisu i procesu rozwiązywania problemów,
- możliwość złożonego sterowania procesem rozwiązywania problemów,
- efektywne środowisko obliczeniowe.

Skomentujemy krótko powyższe postulaty. Dwa pierwsze wynikają z faktu, że w złożonych sytuacjach praktycznych zwykle trudno o jednego eksperta posiadającego wystarczającą wiedzę o wszystkich aspektach rozwiązywanego problemu. Mówiąc o ekspertach, mamy tu zatem na myśli ekspertów dysponujących wiedzą cząstkową, dotyczącą np. zasobów, operacji i uszeregowań. Z tymi ekspertami będą związane osobne źródła wiedzy o różnej w ogólności reprezentacji wiedzy i/lub mechanizmach wnioskowania. Postulat trzeci wynika z dużego stopnia strukturalizacji (czasowej, logicznej) zarówno opisu rozwiązywanych problemów, jak i samego procesu ich rozwiązywania. Postulat czwarty, bodaj najważniejszy, dotyczy możliwości głębokiej ingerencji decydenta w proces konstruowania rozwiązań, a w szczególności możliwości sterowania tym procesem w zależności od dynamicznie wyrażonych preferencji decydenta. Nie zakładamy bowiem preegzystencji globalnego modelu preferencji decydenta na początku procesu decyzyjnego (podejście deskryptywne), a budowę tego modelu w iteracyjnej procedurze dialogowej (podejście konstruktywne). I wreszcie postulat piąty, dotyczący środowiska obliczeniowego. Ogromna złożoność obliczeniowa problemu oraz konieczność szybkich odpowiedzi systemu (praca interaktywna oraz uszeregowania dynamiczne) wymagają efektywnego środowiska obliczeniowego.

Poszczególne etapy powstawania systemu były przedstawione w pracach [1,2,3,4,5]. W niniejszej pracy przedstawiamy nowe elementy systemu, głównie wielokryterialny algorytm

przeszukiwania Tabu oraz implementację równoległą w środowisku metakomputera (rozproszone środowisko komputerów dużej mocy).

W kolejnych rozdziałach omówimy główne elementy składowe systemu eksperckiego. Skoncentrujemy się na wielokryterialnym algorytmie Tabu (TS) oraz jego interakcji z bazami wiedzy systemu, jak również na dialogowej procedurze sterowania procesem konstrukcji rozwiązań (uszeregowań) za pomocą preferencji decydenta.

2. Konstrukcja uszeregowań kompromisowych

W przedstawianym systemie rozwiązanie problemu rozdziału zasobów jest pomyślane jako interaktywny proces przeszukiwania, w skład którego wchodzą wybór i szeregowanie zadań w czasie. Proces konstrukcji uszeregowań podzielony został na dwa etapy: fazę przeszukiwania i fazę interaktywną. W ogólności system generuje pewien zbiór rozwiązań niezdominowanych z wykorzystaniem wielokryterialnego algorytmu TS i następnie na zbiorze tych rozwiązań operuje procedura interaktywna, znajdująca rozwiązania kompromisowe na podstawie preferencji decydenta.

System składa się z trzech modułów:

1. system ekspercki o architekturze tablicowej stanowiący główny moduł,
2. wielokryterialny algorytm TS,
3. procedura interaktywna wspomagająca podejmowanie decyzji użytkownika.

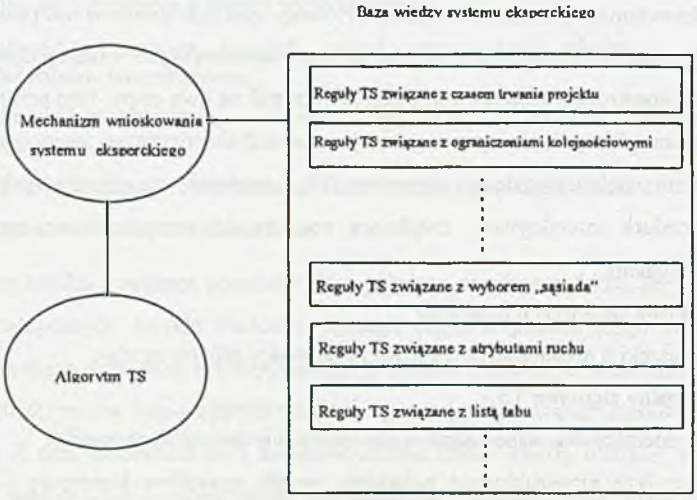
Przedstawiając fazę przeszukiwania pokażemy przede wszystkim koncepcję i możliwości współpracy systemu eksperckiego z algorytmem TS. Rozwiązanie takie stanowi dla użytkownika bardzo elastyczne narzędzie do definiowania jego własnej strategii tabu.

2.1. Faza przeszukiwania

Faza przeszukiwania oparta jest na specjalizowanym algorytmie TS, który współpracując z systemem eksperckim znajduje uszeregowania dopuszczalne dla wszystkich kryteriów danego problemu rozdziału zasobów. Wszystkie elementy specyficzne dla algorytmu dotyczące strategii tabu są zawarte w bazach wiedzy w postaci reguł i obiektów i sterują przebiegiem algorytmu. Algorytm uruchamiany jest dla wszystkich kryteriów jednocześnie z użyciem poszczególnych reguł. TS składa się z następujących modułów: (1) definicja problemu (konfiguracja), (2) definicja ruchu i (3) moduł generujący sąsiadów - właściwa procedura. Interakcja pomiędzy TS i bazami wiedzy systemu eksperckiego jest sterowana przez mechanizm wnioskowania systemu eksperckiego (rys 1).

Gdy podczas przeszukiwania zostaną spełnione pewne warunki zawarte w bazie wiedzy, mechanizm wnioskujący „odpala" regułę lub zbiór reguł. Konkluzje płynące z tych reguł są następnie przekazywane do globalnej bazy danych o stanie systemu. W następnym kroku system powraca do algorytmu TS, który kontynuuje przeszukiwanie z nowymi danymi o stanie systemu.

Struktura Tabu jest w naszym systemie widziana jako postać list wielokrotnych zmieniających się dynamicznie. Dla każdego zadania istnieje dokładnie jedna lista, której rozmiar zależy od możliwych dróg (sposobów wykonania), jakimi dane zadanie może zostać uszeregowane.



Rys. 1. Współpraca algorytmu TS z bazami wiedzy systemu eksperckiego
Fig. 1. Interactions of TS algorithm with knowledge bases of the expert system

Poniżej przedstawiamy kilka przykładowych reguł sterujących przebiegiem algorytmu. Poza regułami dotyczącymi poszczególnych kryteriów przedstawiamy reguły charakterystyczne dla algorytmu TS, definiujące strategię wykorzystania pamięci, atrybuty ruchu, status tabu, kryteria aspiracji etc...

Jako pierwsze przedstawimy reguły reprezentujące atrybuty ruchu dla przejścia od *x_now* do *x_trial.*

RULE #1

IF *Changed*(x_j, *0, 1)* THEN *MoveAttribute:=A1;*

RULE #2

IF *Changed(x_k, 1, 0)* THEN *MoveAttribute:=A2;*

RULE #3

IF *Changed(x_j, 0, 1)* AND *Changed(x_k, 1, 0)* THEN *MoveAttribute:=A3;*

RULE #4

IF *Changed(c(x), c(x__now), c(x_trial))* THEN *MoveAttribute:=A4;*

RULE #5

IF *Changed(g(x), g(x__now), g(x_trial))* THEN *MoveAttribute:=A5;*

Funkcja *g(x)* jest funkcją strategiczną niezależną od *c(x)*, i reprezentującą odległość pomiędzy danym rozwiązaniem i ostatnio znalezionym dobrym rozwiązaniem.

Atrybuty ruchu są wykorzystywane do reprezentacji ograniczeń tabu, zabraniających dokonania pewnych ruchów, jeśli odwracałyby one zmiany dokonane przez te atrybuty. Poniżej przedstawiamy kilka ograniczeń tabu w postaci reguł:

RULE #6

IF *Changed(x_j, 0, 1)* AND *PreviousChanged(x_j, 1, 0)* {x_j - poprzednia zmiana z 1 na 0}

THEN *Move(x_now, x_next):=Tabu;*

RULE #7

IF *Changed(x_k, 1, 0)* AND *PreviousChanged(x_k, 0, 1)* THEN *Move(x_now, x_next):=Tabu;*

RULE #8

IF *(Changed(x_j, 0, 1)* AND *PreviousChanged(x_j, 1, 0))*

OR *(Changed(x_k, 1, 0)* AND *PreviousChanged(x_k, 0, 1))*

THEN *Move(x_now, x_next):=Tabu;*

Ta reguła jest bardziej restrykcyjna niż reguły 6 lub 7 - powoduje ona, że więcej ruchów zostaje oznaczonych jako Tabu.

RULE #9

IF *(Changed(x_j, 0, 1)* AND *PreviousChanged(x_j, 1, 0))*

AND *(Changed(x_k, 1, 0)* AND *PreviousChanged(x_k, 0, 1))*

THEN *Move(x_now, x_next):=Tabu;*

Powyższa reguła jest z kolei mniej restrykcyjna niż reguły 6 lub 7 - powoduje ona, że mniej ruchów zostaje oznaczonych jako Tabu.

RULE #10

IF *c(x)=t'* AND *c(x')=t'* THEN *Move(x_now, x_next):=Tabu;*

c(x') jest tutaj wartością funkcji *c(x)* dla pewnego, odwiedzonego wcześniej rozwiązania x'.

Ograniczenia związane z problemami rozdziału zasobów są w systemie pomyślane właśnie jako ograniczenia tabu. Użytkownik może zdefiniować dowolną liczbę uszeregowań (lub ich fragmentów) jako uszeregowania tabu. Na przykład może on zażądać, by żadne zadanie z grupy A nie było wykonywane w tym samym czasie co dowolne zadanie z grupy B. Takie ograniczenia w bardzo łatwy sposób mogą być zapisane jako reguły.

W pewnych warunkach ograniczenia tabu mogą zostać przekroczone. W naszym przypadku można sobie wyobrazić taką sytuację, że status tabu danego ruchu może być pominięty, jeżeli ruch ten prowadzi do rozwiązania dopuszczalnego i jeśli użytkownik wskaże warunki, w jakich dane uszeregowanie chwilowe może zostać wybrane do dalszej oceny. Służą do tego kryteria aspiracji, określające sposób, w jaki ograniczenia tabu mogą zostać przekroczone. Właściwe wykorzystanie kryteriów aspiracji jest bardzo ważne dla jakości i wydajności algorytmu TS. Wyróżniamy dwa rodzaje tych kryteriów: aspiracje ruchu i aspiracje atrybutu. Kiedy aspiracja ruchu jest spełniona, to zachodzą zmiany w klasyfikacji atrybutów tabu dla wszystkich ruchów. Spełnienie aspiracji atrybutu powoduje zmiany w atrybutach aktywnych. Istnieje specjalna grupa reguł odpowiedzialnych za definicje aspiracji. Użytkownik może zdefiniować wszystkie znane z teorii aspiracje (aspiration by default, aspiration by objective, aspiration by influence). Poniżej pokazujemy przykład aspiracji przedstawionej poprzez cel jako najczęściej stosowanej.

RULE#11

IF c(x_trial) < *best_cost*

THEN *Move(x_now, x_trial)* Ruch aspiracyjny spełniony, gdy c(x_trial)<best_cost)

Innym bardzo ważnym aspektem algorytmu Tabu jest strategia wyboru najlepszego sąsiada. Strategia ta jest określona przez definicję najlepszego ruchu sporządzoną przez decydenta. Może on na przykład ustanowić pewne relacje pomiędzy stanem końcowym (rozwiązaniem) i najlepszym sąsiadem danego rozwiązania częściowego w następujący sposób. Niech x będzie danym rozwiązaniem bieżącym, $v(x)$ miarą niedopuszczalności rozwiązania x, a x' niech będzie sąsiadem, dla którego $c(x')-c(x)$ stanowi minimum (tak więc $v(x')>0$). Niech x'' będzie natomiast sąsiadem, dla którego minimalizujemy $v(x'')-v(x)$. Wybór najlepszego sąsiada może być tutaj zdefiniowany przez następujący ciąg reguł:

RULE#12

IF v(x)>0 THEN *BestNeighbour(x)*:=x"

RULE#13

IF v(x)=0 AND v(x")>0 THEN *BestNeighbour(x)*:=x'

RULE#14

IF v(x)=0 AND v(x")=0AND c(x")≤ *t* THEN *BestNeighbour(x)*:=x"

RULE#15

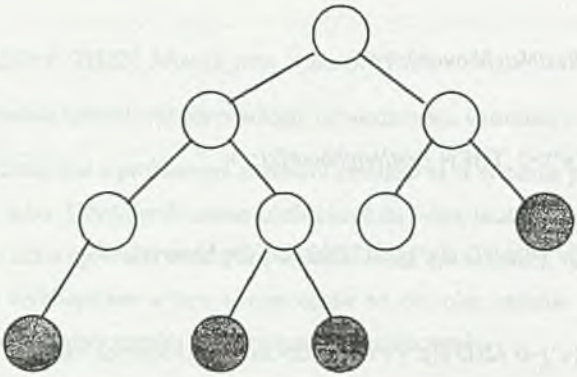
IF v(x)=0 AND v(x")=0 AND c(x") > *t* THEN *BestNeighbour(x)*:=x'

Strategia zawarta w tym przykładzie mówi, że jeżeli dane, chwilowe uszeregowanie jest niedopuszczalne, to najlepszym sąsiadem takiego uszeregowania jest takie, które w maksymalnym stopniu redukuje tę niedopuszczalność.

2.2. Równoległość obliczeń

Algorytm TS jest implementowany w środowisku metakomputera Poznańskiego Centrum Superkomputerowo Sieciowego. W celu zrealizowania podstawowych postulatów związanych z opracowaniem systemu, jako główną ideę podczas jego implementacji przyjęto przetwarzanie równoległe. Ogólnie rzecz ujmując, algorytm TS polega na przeszukiwaniu drzewa w celu znalezienia odpowiednich rozwiązań niezdominowanych. Równoległy algorytm TS jest skonstruowany w ten sposób, że początkowo jest tworzone jedno zadanie (w sensie modelu programowania równoległego *task-channel*) reprezentujące korzeń drzewa. Zadanie to oblicza dany węzeł, i, jeżeli nie jest on rozwiązaniem (dopuszczalnym) niezdominowanym, to tworzy on nowe zadania dla każdego wywołania procedury *T_search (poddrzewo)*. Dla każdego nowego zadania tworzony jest kanał w celu przekazania każdemu „rodzicowi" nowych zadań wszystkich rozwiązań zlokalizowanych w poddrzewie. W ten sposób wraz z progresją przeszukiwania drzewa rozwiązań w dół są tworzone nowe zadania z odpowiadającymi im kanałami (rys 2.)

Każde kółko reprezentuje węzeł w drzewie rozwiązań i jednocześnie wywołanie procedury *T search*. W dowolnej chwili kilka zadań aktywnie uczestniczy w rozwiązywaniu problemu (na rysunku są to kółka zaciemnione).



Rys. 2. Struktura zadań podczas przeszukiwania drzewa rozwiązań
Fig. 2. Task structure for the TS parallel algorithm

W czasie konstrukcji algorytmu silny nacisk położono na cztery aspekty związane z algorytmami równoległymi: współbieżność (wykonywanie wielu czynności jednocześnie), skalowalność (dostosowywanie się algorytmu do liczby procesorów), lokalność obliczeń (wysoki współczynnik odwołań do pamięci lokalnej względem odwołań do pamięci zdalnej), modularność (dekompozycja problemu na mniejsze moduły).

3. Sterowanie preferencjami decydenta

W ogólności, na drodze zastosowania wieloktryterialnego algorytmu Tabu, zostaje wygenerowany zbiór uszeregowań dopuszczalnych. Spośród tych uszeregowań do dalszych rozważań bierzemy zbiór uszeregowań niezdominowanych, dla którego stosujemy interaktywną procedurę złożoną z trzech kroków [6].

Krok początkowy

Dla rozwiązań niezdominowanych tworzy się macierz wypłat Z (pay-off table) o wymiarach kxk, której elementy z_{ij} oznaczają wartość kryterium i dla najlepszego rozwiązania z punktu widzenia kryterium j ($i,j=1,...k$, gdzie wartość k jest liczbą rozpatrywanych kryteriów). Przekątna tej macierzy definiuje zatem rozwiązanie *idealne*, w ogólności nierealizowalne w ramach przyjętych ograniczeń, przy czym wektor najgorszych rezultatów dla poszczególnych kryteriów jest rozwiązaniem *nadir*.

Krok obliczeniowy

Znajdujemy uszeregowanie najbliższe *idealnemu* w sensie wybranej funkcji skalaryzującej, którą jest rozszerzona, ważona norma Czebyszewa.

Krok decyzyjny

Uszeregowanie znalezione w kroku obliczeniowym przedstawiane jest decydentowi do oceny i, jeśli ten uzna je za satysfakcjonujące, to procedura się zatrzymuje. W przeciwnym razie decydent określa kryterium, na którym decyduje się stracić o określoną wartość Δg, żeby zyskać na innych kryteriach. Wartość Δg jest przeliczana następnie na wartość funkcji kary, dodawana do funkcji skalaryzującej i procedura wraca do kroku obliczeniowego.

4. Podsumowanie

W niniejszej pracy przedstawiono szkieletowy system ekspercki dla obszernej klasy deterministyczych problemów rozdziału zasobów. Opisano m.in. równoległy, wielokryterialny algorytm Tabu znajdujący uszeregowania niezdominowane ze względu na wszystkie kryteria jednocześnie oraz jego interakcję z systemem eksperckim.

Dalsze prace pójdą w kierunku integracji poszczególnych elementów składowych systemu opisanych w pracach [1,2,3,4,5] i w niniejszej pracy. Przeprowadzone zostaną również doświadczenia polegające na tworzeniu różnych baz wiedzy dotyczących wielu rzeczywistych problemów rozdziału zasobów i badaniu efektywności przedstawionego podejścia.

LITERATURA

1. Nabrzyski, J., Węglarz, J.: On a Knowledge-based System for Multiobjective Project Scheduling, Materiały 4 Międzynarodowego Kongresu Inżynierii Przemysłowej (4th International Congress of Industrial Engineering), Marsylia 5-17.12.1993, s.131-139.
2. Nabrzyski, J., Węglarz, J.: Koncepcja szkieletowego systemu eksperckiego dla problemów rozdziału zasobów. Zeszyty Naukowe Politechniki Śląskiej, Seria Automatyka, z. 115, 1994, s. 348-358.
3. Nabrzyski, J., Węglarz, J.: A Knowledge-based Project Scheduling System, Raport Instytutu Informatyki Politechniki Poznańskiej, RA-94/004, październik 1994.
4. Nabrzyski J., Węglarz J.: On a Knowledge Based Project Scheduling" w: Journal of Decision Systems , (w druku).
5. Nabrzyski J., Węglarz J.: On an Expert System with Tabu Search for Multiobjective Project Scheduling, Materiały konferencji INRIA/IEEE (Conference on Emerging Technologies and Factory Automation ETFA'95), Paryż 10-13 październik, 1995.
6. Słowiński, R., Soniewicki, B. Węglarz, J.: DSS for Multiobjective Project Scheduling, w: European Journal of Operational Research, Volume 79, Numer 2, 8 grudzień, 1994, str.220-229.

Recenzent: Prof.dr hab.inż. Jerzy Klamka

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

We present a decision support system for multiobjective project scheduling problems under multiple category resource constraints. It handles quite a general class of nonpreemptive scheduling problems with renewable, nonrenewable and doubly-constrained resources, multiple performing modes of activities, precedence constraints in the form of an activity network and multiple project performance criteria of time and cost type. The system uses a multicriteria decision making and an expert system approaches together with a Tabu Search (TS) method for solving the above class of problems. We concentrate ourselves on the interactions between TS algorithm and a knowledge base of the expert system, containing knowledge that concerns different aspects of TS technology, resources and schedules. We also show an interactive procedure organized on a set of nondominated schedules.

Multiobjective Project Scheduling problems are invariably very complex; they require taking into account such different and conflicting factors as due date requirements, cost restrictions, alternative performing modes or process plans etc. Computationally even the simplest of such problems are hard.

In this paper we propose a multicriteria Tabu Search (TS) approach. The specificity of the TS (tabu lists, history of the search) gives a possibility of using the expert system to control the search. The knowledge base of the expert system contains all the knowledge which is necessary to provide the user with an efficient process of finding compromise solutions.

Simply stated the problems addressed are how to schedule precedence and resource-constrained tasks of a project in order to find the best compromise solution, based on decision maker's preferences, or a set of "good" compromise solutions. Resource constraints can concern: (i) the amount available at every moment, and (ii) the consumption over a given time period. If only constraint (i) is imposed we speak about renewable resources, if only (ii) is defined - about nonrenewable resources, and if both (i) and (ii) - about doubly-constrained resources. The set of project performance measures is composed of time objectives (e.g. project duration, mean weighted flow time, mean weighted lateness) and cost objectives (e.g. weighted resource consumption, net present value). The specific form of the scheduling problems examined here permits each task to be performed in one of several ways called operating modes or simply modes. Each mode represents a different way of combining resources to accomplish a given task. The duration and resource requirements of each task mode are known a priori, and thus we can speak about resource-duration interactions. In this model a variety of multiple resource-duration interactions, such as using different technologies or types of labor to accomplish the same task, can be expressed and evaluated.

In our system the process of constructing compromise schedules is thought as an interactive search process involving the selection and sequencing of activities and resources over time to achieve the problem objectives, subject to a set of constraints. The process is divided into two phases: the search phase and the interactive one. In general, the system generates a set of plenty of feasible schedules using parallel TS for multiple criteria. From among these schedules nondominated ones are then found. On the set of them the interactive procedure is organized yielding compromise solutions depending on Decision Maker's (DM) preferences. The system is composed of three modules: (1) distributed expert system which runs as a kernel of the whole system, (2) parallel TS algorithm for multiobjective scheduling, (3) interactive decision support procedure for multiobjective scheduling. Exploring the search phase we show the main concepts and possibilities of employing an expert system to control the behavior of the TS algorithm. This solution gives DM a very flexible tool for defining his own tabu strategy.

In the paper we don't focus on the TS algorithm itself, but on its interactions with a knowledge base of the expert system.