ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1986
Seria: AUTOMATYKA z. 84 Nr kol. 894

Krzysztof Ficoń
Wojskowa Akademia Techniczna

SZEREGOWANIE ZADAŃ METODĄ KOLOROWANIA GRAFU KONFLIKTÓW

Streszczenie. W pracy przedstawiono próbę wykorzystania teorii grafów do rozwiązania dyskretnego problemu szeregowania zadań. Metodę rozwiązania oparto na kolorowaniu wierzchołków pewnego grafu zwykłego zwanego grafem konfliktów, do którego należy przekształcić pierwotny problem szeregowania. Wierzchołki grafu konfliktów obrazują zbiór zadań oczekujących na realizację. Liczność zbioru maszyn jest dowolna i może być określona parametrycznie. Zbiór krawędzi przedstawia relację konfliktów, oznaczającą niemożność jednoczesnego wykonania odpowiedniej pary zadań.

Do rozwiązania problemu zaproponowano nową, binarną metodę suboptymalnego kolorowania wierzchołków grafu. Poszukiwane uszeregowanie będziemy utożsamiać z podziałem zbioru wierzchołków na podzbiory niezależne /jednakowo kolorowalne/.

1. Wstęp

Niech dany będzie zbiór zadań Z, które należy wykonać na pewnym zbiorze maszyn, którego liczność jest dowolna i może być ustalona parametrycznie. Zbiór zadań Z jest zbiorem niepustym i skończonym, a jego elementy $z_i \in Z$ stanowią tzw. zadania niepodzielne, tj. zadania, których realizacja nie może być przerwana. Na zbiorze zadań Z określona jest pewna dwuczłonowa relacja, obrazująca konflikt czasowy lub przestrzenny między poszczególnymi parami zadań. Relacja konfliktu wyrażać może np. uwarunkowania technologiczne /kolejnościowe/ istniejące między niektórymi parami zadań i oznacza niemożność jednoczesnego wykonywania tych zadań - konflikt czasowy lub konieczność wykonywania ich na różnych maszynach - konflikt przestrzenny.

Należy znaleźć taką kolejność realizacji zbioru $|Z|$ zadań, aby spełnić ustaloną funkcję kryterium. Dopuszczalność uszeregowania będziemy odnosić do warunku "zero konfliktów" - zgodnie z przyjętym grafem konfliktów.

W tak sformułowanym problemie pomijamy chwilowo kwestie maszyn, ich liczby oraz uwarunkowania kolejnościowe /technologiczne/. Przykładowo możemy założyć istnienie zbioru maszyn adekwatnego do potrzeb generowanego harmonogramu. Założenie to eliminuje z modelu czas, a tym samym czas realizacji poszczególnych zadań t_{ij} na odpowiednich maszynach. Zgodnie z powyższym problem optymalizacji uszeregowania został sprowadzony do zadania pokolorowania grafu konfliktów minimalną liczbą kolorów - równą

liczbie chromatycznej grafu. Optymalne pokolorowanie grafu konfliktów oznaczać może np. minimalizację liczby maszyn niezbędnych do "obsłużenia" zbioru $|Z|$ zadań. Problem sterowania liczbą wykonawców występuje stosunkowo często w tzw. harmonogramach o ustalonym czasie realizacji [3].

Ze względu na NP-zupełną złożoność obliczeniową ogólnego problemu kolorowania grafu i małą efektywność optymalnego algorytmu Maghout-Weissmana wynikającą m.in. z konieczności minimalizacji pewnych formuł algebraicznych [8], szeroko stosuje się różne algorytmy i metody przybliżone, jak np.: metoda Welsha-Powella [1], metoda Wooda [12] czy metoda Burlagi [4]. Do efektywnego szeregowania zadań zaproponowano w pracy nowy, suboptymalny algorytm kolorowania wierzchołków grafu, bazujący na binarnej macierzy przyległości wierzchołków, którego przydatność oceniono między innymi za pomocą badań komputerowych.

2. Kolorowanie grafu konfliktów

Jak zostało wspomniane zastosowanie metody kolorowania wierzchołków grafu wymaga uprzedniej transformacji problemu szeregowania do tzw. grafu konfliktów technologicznych G. Jest to graf zwykły bez pętli i łuków oraz wierzchołków izolowanych [10]

$$G = \langle Z, U \rangle \qquad /1/$$

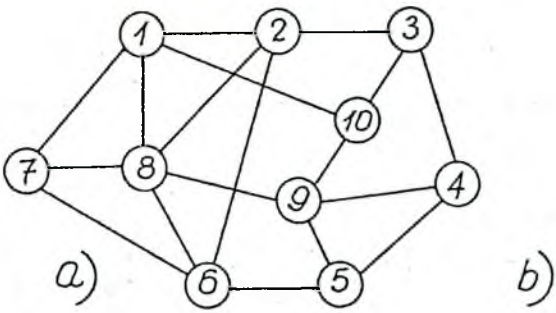
gdzie: $Z = \left\{ z_i;\ i = \overline{1,I} \right\}$ - zbiór wierzchołków,

$U = \left\{ u_n;\ n = \overline{1,N} \right\}$ - zbiór krawędzi.

Zbiór wierzchołków Z reprezentuje zbiór zadań $\left\{ z_i \right\}$, oczekujących na obsługę, dla których należy znaleźć optymalne uszeregowanie /kolejność realizacji/. Zbiór krawędzi U obrazuje konflikty czasowo-przestrzenne wynikające z technologii wykonywania zbioru zadań Z. Formalnie zbiór krawędzi /konfliktów/ jest relacją dwuczłonową określoną na produkcie kartezjańskim $Z \times Z$

$$U \subset Z \times Z = \left\{ \langle z_i, z_j \rangle \right\} = \left\{ u_n;\ n = \overline{1,N} \right\} \qquad /2/$$

którego elementami $u_n \in U$ są pary zadań $\langle z_i, z_j \rangle$, będące w konflikcie np. czasowym oznaczającym niemożność jednoczesnego wykonywania zadania z_i oraz z_j w tej samej chwili. Relacje konfliktów U determinuje najczęściej technologia wykonywania zbioru zadań Z czy też inne względy np. organizacyjne a także dostępność materiałów, surowców, narzędzi itp.



Z_1	Z_2	Z_6	X
Z_3	Z_4	Z_9	X
Z_5	Z_7	X	Z_8
X	Z_{10}	X	X

Rys.1. Przykładowy graf konfliktów /a/ i odpowiadający mu harmonogram /b/
Example of conflict graph /a/ and its schedule /b/.

Jeżeli potrafimy znaleźć podział zbioru zadań Z

$$\left\{ Z_1, Z_2, \ldots, Z_k, \ldots, Z_K \right\} \qquad /3/$$

taki, że spełniony jest warunek

$$\bigwedge_{z_i, z_j \in Z} \; \bigwedge_{1 \leq k \leq K} \left(z_i, z_j \in Z_k \right) \Rightarrow \left(z_i, z_j \in U \right) \qquad /4/$$

to zbiory $\left\{ Z_k \right\}$ składają się z zadań, które mogą być wykonywane w tej samej chwili. Wobec tego K jest liczbą jednostek czasowych realizacji wszystkich zadań $|Z|$. Metodami kolorowania grafu można wyznaczyć minimalny podział /3/, a K jest wówczas minimalną liczbą jednostek czasowych potrzebnych do wykonania $|Z|$ zadań.

Niech dany będzie graf G i liczba naturalna K. Mówimy, że graf G jest K-barwny jeżeli jego wierzchołki można pomalować K-różnymi barwami w taki sposób, że żadne dwa wierzchołki sąsiednie nie są pokolorowane tą samą barwą. Najmniejszą liczbę K, dla której graf G jest K-barwny nazywamy liczbą chromatyczną grafu $\mathcal{X}(G)$. Zatem K-pokolorowanie grafu G jest równoważne wyznaczeniu rodziny podzbiorów /3/ takiej, że spełnione są warunki:

$$1^{\circ} \quad \emptyset \neq Z_k \subseteq Z \; ; \qquad 1 \leqslant k \leqslant K \qquad /5/$$

$$2^{\circ} \quad Z_i \cap Z_j = \emptyset \quad \text{dla } i \neq j \; ; \; i,j = \overline{1,K} \qquad /6/$$

$$3^{\circ} \quad \left(z_i, z_j \in Z \right) \Rightarrow \left(\langle z_i, z_j \rangle \in U \right) \qquad /7/$$

Jeżeli ponadto zachodzi warunek

$$\bigcup_{k=1}^{\mathcal{K}} Z_k = Z \qquad /8/$$

to pokolorowanie /3/ nazywamy pełnym. Pokolorowanie pełne, takie że

$$K = \mathcal{H}(G) \qquad /9/$$

nazywamy optymalnym. Podzbiór wierzchołków Z_k grafu G spełniający warunek /7/ nazywamy niezależnym.

Jak już wspomniano istnieje szereg algorytmów kolorowania grafu zarówno dokładnych /optymalnych/ [5], jak też przybliżonych /suboptymalnych/ [7], [11], [12] jednak ich sprawność i efektywność odniesiona do takich parametrów, jak: czas obliczeń i zajętość pamięci operacyjnej jest ciągle daleka od zadowalającej. Poniżej zostanie przedstawiony nowy algorytm suboptymalnego kolorowania wierzchołków grafu G bazujący na binarnej macierzy przyległości wierzchołków $\mathcal{A}(G)$.

3. Niezależne zbiory wierzchołków

Zgodnie z definicją problemu kolorowania /5/ ÷ /8/ kolor k-ty mogą tworzyć tylko wierzchołki /zadania/ $z_i \in Z_k$, nie będące wzajemnie w relacji konfliktu /2/. Zbiory $Z_k \subseteq Z$ nazywać będziemy zbiorami niezależnymi /jednakowo kolorowalnymi/. Proces konstruowania zbiorów niezależnych $Z_k \subseteq Z$ rozpoczyna się od wskazania wierzchołka inicjującego dany kolor $z_k \in Z$, który dzieli pozostałe wierzchołki grafu G na dwa rozłączne podzbiory Z_o i Z_1

$$Z = z_k \cup Z_o(z_k) \cup Z_1(z_k) \qquad /10/$$

gdzie: $Z_o(z_k)$ - zbiór wierzchołków nieprzyległych do z_k,

$Z_1(z_k)$ - zbiór wierzchołków przyległych do z_k.

Do zbioru niezależnego $Z_k \subseteq Z$ mogą należeć tylko wierzchołki ze zbioru $Z_o(z_k)$, przy czym żaden wierzchołek $z_i \in Z_k$ nie może pozostawać w relacji konfliktu z jakimkolwiek innym wierzchołkiem tworzącym dany kolor Z_k. W ogólności procedura generowania zbiorów niezależnych polega na sukcesywnym włączaniu do zbioru Z_k wierzchołków nieprzyległych ze stopniowo redukowanego zbioru $Z_o(z_k)$

$$Z_k = z_k \cup \left\{ Z_o(z_k) \setminus \bigcup_{k=1}^{\mathcal{K}} Z_{k+i} \right\} \qquad /11/$$

Konstruowanie danego koloru Z_k kończy się w momencie wyczerpania się zbioru $Z_o(z_k)$

$$\left(Z_k = Z_k^{max}\right) \Longleftrightarrow \left(Z_o(z_k) = \emptyset\right) \qquad /12/$$

Kolejny nowy kolor k+1 jest inicjowany przez następny wierzchołek bazowy z_{k+1} należący do zbioru $Z_1(z_k)$, czyli

$$\left(z_{k+1} \in Z_{k+1}\right) \Longleftrightarrow \left(Z_o(z_k) = \emptyset\right) \wedge \left(Z_1(z_k) \neq \emptyset\right) \qquad /13/$$

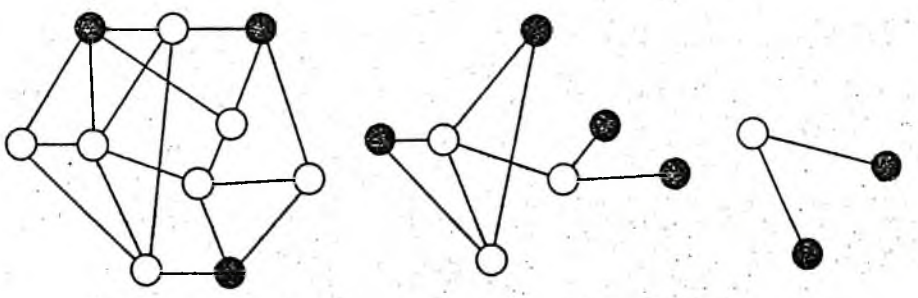
który podobnie jak wierzchołek z_k dzieli zbiór pozostałych wierzchołków na dwa rozłączne zbiory Z_o i Z_1

$$Z \setminus Z_k = z_{k+1} \cup Z_o(z_{k+1}) \cup Z_1(z_{k+1}) \qquad /14/$$

Proces generowania kolejnych nowych kolorów $k+i \in K$ prowadzony jest aż do momentu, gdy wyczerpie się zbiór wierzchołków dotychczas niepokolorowanych

$$\left(k + i = K\right) \Longleftrightarrow \left(Z \setminus \bigcup_{k=1}^{K} Z_k = \emptyset\right) \qquad /15/$$

co oznacza zakończenie procedury kolorowania grafu G. Graficzną interpretację procedury kolorowania grafu metodą redukowanych zbiorów niezależnych przedstawia rys.2.



Rys.2. Procedura generowania zbiorów niezależnych
Algorithm of generation independent sets

Jeżeli uzyskana liczba kolorów K /podzbiorów Z_k/ spełnia warunek /9/, to otrzymaliśmy pokolorowanie minimalne /optymalne/, co na gruncie teorii harmonogramów odpowiada np. minimalnej liczbie maszyn lub minimalnej długości uszeregowania /przy nieograniczonej liczbie maszyn/.

4. Binarna metoda kolorowania grafu

Procedurę kolorowania grafu metodą wyznaczania zbiorów niezależnych można sprowadzić do operacji logicznego dodawania odpowiednich wierzchołków nieprzyległych. Do jej numerycznej realizacji wykorzystamy binarną

macierz przyległości wierzchołków grafu $A(G)$

$$A(G) = \left[a_{ij}\right]_{|Z|\times|Z|} \rightarrow \left\{0,1\right\} \qquad /16/$$

której elementy $\left[a_{ij}\right]$ określamy następująco

$$a_{ij} = \begin{cases} 1 & - \text{ gdy } \langle z_i, z_j \rangle \in U \\ 0 & - \text{ przeciwnie.} \end{cases} \qquad /17/$$

Jak wynika z definicji /16/ jest to macierz kwadratowa o wymiarze $|Z| \times |Z|$, odznaczająca się m.in. własnością symetri względem głównej przekątnej, co zostało wykorzystane w algorytmie komputerowym, bazującym tylko na górnej podmacierzy trójkątnej.

Rozpatrzmy dowolny n-ty wiersz macierzy $A(G)$, zawierający elementy zero-jedynkowe. Wszystkie elementy zerowe $/a_{ij} = 0/$ tworzą zbiór wierzchołków nieprzyległych $Z_o(z_k)$

$$Z_o(z_k) = \left\{ a_{ij} = 0 \; ; \; i=n, \; n \leqslant j \leqslant N \right\} \qquad /18/$$

a elementy jednostkowe $/a_{ij} = 1/$ tworzą zbiór wierzchołków przyległych $Z_1(z_k)$

$$Z_1(z_k) = \left\{ a_{ij} = 1 \; ; \; i=n, \; n \leqslant j \leqslant N \right\} \qquad /19/$$

Procedura konstruowania k-tego koloru /zbioru $Z_k \subseteq Z$ / może być wyrażona jako suma logiczna odpowiednich wierszy macierzy $A(G)$. Inicjuje ją wybór wierzchołka bazowego $z_k \in Z$ rozpoczynającego dany kolor, co jednocześnie determinuje zbiór wierzchołków nieprzyległych $Z_o(z_k)$, a tym samym składniki sumy logicznej /11/. Tak uzyskana suma /11/ wyznacza z kolei dalsze postępowanie. Zgodnie z wyrażeniem /18/ elementy zerowe $/a_{ij}=0/$ w wierszu sumarycznym określają numery wierszy /wierzchołków/, które są potencjalnymi składnikami kolejnej operacji sumowania. Proces dołączania wierzchołków nieprzyległych /sumowania logicznego wierszy $a_{i(\cdot)}$ /, kontynuowany jest aż do momentu wyczerpania się zbioru $Z_o(z_k)$. Spełnienie warunku /15/ inicjuje następny k+1 kolor, o ile prawdziwe jest wyrażenie /13/. Każdy z tak uzyskanych zbiorów niezależnych $Z_k \subseteq Z$ jest jednocześnie zbiorem maksymalnym.

Binarny algorytm kolorowania wierzchołków grafu G można przedstawić za pomocą następującej sekwencji.

1° Dla danego grafu konfliktów G zbuduj odpowiadającą mu binarną macierz przyległości wierzchołków $A(G)$, w której należy wydzielić górną podmacierz trójkątną.

2° W podmacierzy trójkątnej $A(G)$ wybierz wierzchołek /wiersz $a_{i(\cdot)}$ / inicjujący dany kolor Z_k.

3° Numer kolejnego wierzchołka dołączonego $z_n \in Z$ ustal na podstawie numerów elementów zerowych $a_{ij}=0$ w wierszu inicjującym /sumarycznym/ $a_{n(\cdot)}$

4° Dodaj logicznie wierzchołek bazowy z_k /multiwierzchołek Z_k/ do wybranego wierzchołka dołączonego z_n.

5° W nowopowstałym wierszu sumarycznym Z_k^n ustal aktualny zbiór wierzchołków nieprzyległych $Z_o(z_{k+n})$.

Przypadek $Z_o(z_{k+n}) = \emptyset$ oznacza wyczerpanie się możliwości dalszego posługiwania się k-tym kolorem. W ten sposób został utworzony maksymalny zbiór niezależny typu Z_k. Jeżeli wszystkie wierzchołki grafu G zostały pokolorowane /włączone do odpowiednich zbiorów niezależnych Z_k/, to algorytm kończy swoje działanie. W przeciwnym przypadku inicjowany jest następny k+1 kolor począwszy od kroku 2°.

5. Podsumowanie.

Przedstawione w pracy podejście do problemu szeregowania zadań wymaga transformacji zagadnienia pierwotnego do grafu konfliktów G, zawierającego obraz czasowo-przestrzennych kolizji /uwarunkowań/ technologicznych opisanych na zbiorze zadań Z. Dwa zadania $z_i, z_j \in Z$ połączone wspólną krawędzią $u_n \in U$ nie mogą być wykonywane jednocześnie /w tym samym czasie lub na tej samej maszynie/. Dopuszczalne szeregowanie zbioru zadań zostało sprowadzone do problemu kolorowania wierzchołków grafu konfliktów. Wierzchołki /zadania z_i/ tworzące jeden kolor /podzbiór niezależny Z_k/ zawierają zadania, które mogą być realizowane jednocześnie, gdyż między tymi zadaniami nie występuje konflikt technologiczny. Liczba uzyskanych kolorów K odpowiada liczbie interwałów czasowych wykonywania wszystkich $|Z|$ zadań. Natomiast liczność poszczególnych podzbiorów niezależnych $|Z_k|$ jest równa liczbie maszyn niezbędnych aktualnie do realizacji zadań.

Problem optymalizacji generowanego harmonogramu można rozpatrywać bądź w kontekście minimalizacji długości uszeregowania /minimalizacja liczby użytych kolorów/, bądź według kryterium minimalizacji liczby maszyn /minimalizacja maksymalnego podzbioru niezależnego/. Podejście oparte na metodzie kolorowania grafu konfliktów daje także możliwość konstruowania dwukryterialnych uszeregowań minimalizujących jednocześnie ich długość oraz liczbę wykonawców, co wymaga jednak odpowiedniego sterowania procedurą kolorowania grafu np. na etapie ustalania wierzchołka inicjującego dany kolor a także w trakcie wybierania wierzchołków dołączonych. Widać więc, że możliwości podejścia opartego na kolorowaniu wierzchołków grafu są bardzo szerokie a ewentualne ograniczenia przez nią implikowane mogą być znacznie złagodzone. Przykładowo w modelu można obok relacji konfliktu

uwzględnić również ograniczenia kolejnościowe w postaci relacji czasowego poprzedzania /następstwa/, a także włączyć do zadań czasy ich realizacji czy system wag bądź priorytetów. Wszystkie te parametry mogą być rozpatrywane na etapie generowania zbiorów niezależnych i łączenia ich w ciągi tworzące podział zbioru zadań. Bardzo szerokie możliwości stwarza metoda w przypadku transformacji problemu pierwotnego do postaci hipergrafu konfliktów, stanowiącego uogólnienie pojęcia grafu. Istnieje wówczas możliwość uwzględnienia złożonych zależności i uwarunkowań technologicznych za pomocą wieloczłonowych relacji konfliktów.

O atrakcyjności podejścia decyduje przede wszystkim sprawność obliczeniowa i globalna efektywność binarnego algorytmu kolorowania wierzchołków grafu /hipergrafu/ konfliktów. Zaproponowany algorytm wyróżnia się takimi cechami, jak: skończoność, niezawodność, duża szybkość i precyzja działania a także wysoka podatność na implementację komputerową, o czym świadczą uzyskane wyniki z symulacji komputerowej. W przypadku "czystego" kolorowania grafu algorytm generuje zawsze rozwiązanie optymalne, w którym liczba użytych kolorów jest równa liczbie chromatycznej grafu. Algorytm jest szczególnie przydatny w problemie szeregowania zadań, gdyż obok wprowadzonych ograniczeń umożliwia on uwzględnienie dodatkowych parametrów opisujących zbiór zadań, co daje możliwość wielokryterialnego sterowania generowanym rozwiązaniem.

LITERATURA

[1] Błażewicz J. i inni: Badania operacyjne dla informatyków. WNT, Warszawa 1983.

[2] Błażewicz J. i inni: O niektórych problemach szeregowania zadań wieloprocesowych. Zeszyty Naukowe PŚL, ser. Automatyka z.74 Gliwice 1984.

[3] Burlaga H.: Minimalizacja ilości wykonawców operacji o ustalonym harmonogramie. Biuletyn WAT, nr 12/241/, Warszawa 1975.

[4] Burlaga H.: Suboptymalne metody kolorowania grafów. Biuletyn WAT, nr 12/246/, Warszawa 1975.

[5] Christofides N.: An algorithm for the chromatic number of a graph. Computer Journal vol.14 nr 1, 1971.

[6] Coffman E.G.: Teoria szeregowania zadań. WNT, Warszawa 1980.

[7] Jerszow A.P.: Wprowadzenie do teorii programowania, WNT,Warszawa 1981.

[8] Korzan B.: Elementy teorii grafów i sieci. Metody i zastosowania. WNT, Warszawa 1978.

[9] Kubale M.: Niektóre zagadnienia szeregowania prac zespołowych. Materiały na konferencję "Cybernetyka w gospodarce morskiej" t.1 z.2 PTC Gdańsk, WSM, Gdynia 1985.

[10] Wilson R.J.: Wprowadzenie do teorii grafów. PWN, Warszawa 1985.
[11] Welsh D., Powell M.: An upper bound for the chromatic number of a graph. Computer Journal vol. 10 nr 1, 1967.
[12] Wood D.: A technique for colouring a graph applicable to large scale timetabling problems. Computer Journal vol.12 nr 4, 1969.
[13] Węglarz J.: Sterowanie w systemach typu kompleks operacji. PWN, Warszawa - Poznań 1981.

Recenzent: Doc.dr h.inż.Józef Grabowski
Wpłynęło do Redakcji do 1986.04.30

РАСПИСАНИЕ ЗАДАЧ ПО МЕТОДУ РАСКРАСКИ ГРАФА КОНФЛИКТОВ

Р е з ю м е

В работе предоставлен метод дискретного расписания задач базующий на раскраске графа конфликтов. Вершины этого графа представляют множество задач ожидающих выполнения. Численность множества машин произвольна. Множество рёбер представляет реляцию конфликта, которая обозначает невозможность одновременного решения соответственной пары задач.

Для разрешения проблемы дискретного расписания задач предложен новый алгоритм субоптимальной расцветки вершин графа.

TASKS SCHEDULING BY THE USE OF THE CONFLICTS GRAPH COLOURING METHOD

S u m m a r y

In the paper the method of discrete scheduling task on the basic of colouring vertices of certain graph has been presented. The method needs transformation of scheduling problem in so called conflict graph to be done. In the graph all independent / of the same colour / sets of vertices ought to be determined. The nodes of the conflicts graph represent the set of tasks waiting for realisation. Number of elements of the set machines is not limited and can be determined in the parametric way. The set of edge represents the conflicts relation. The relation denotes impossibility of simultaneous performance of the suitable pair of tasks. For solving the problem a new binary method of suboptimum colouring of egdes of the conflicts graph has been proposed. The strait basic of the method action is a binary matrix of the incidence of vertices. The requested schedule is understood as the division of the set of vertices into the independent subsets. At the end of the paper the possibility of ignoring certain bounds and significant generalization of the range of applications of the method have been shown.