obliczono błąd względny

$$\eta^{HS} = \frac{C^S - C^{NS}}{C^S - C^{HS}} \tag{6.67}$$

dla każdego przykładu takiego, że $C^S > C^{HS}$. Stwierdzono zachodzenie nierówności $C^S > C^{HS}$ w 94 procentach przypadków (3750 przykładów). Odpowiednia średnia wartość błędu względnego dla tych przykładów wynosiła 0,86. Otrzymaliśmy $\eta^{HS} = 1$ (tzn. $C^{NS} = C^{HS}$) dla 2840 przykładów (76 procent z 3750). Ponieważ $C^P \geq C^{HS}$, zatem $\eta^P \geq \eta^{HS}$. Oznacza to, że algorytm NS generuje rozwiązania bliższe tym z algorytmów P i HS niż rozwiązaniu z algorytmu S chociaż ma n-krotnie mniejszą złożoność obliczeniową.

6.9 Algorytm C

Algorytmy dokładne w formie metod przeglądu opartych na schemacie podziału i ograniczeń, dla rozważanego problemu, były analizowane m.in. w pracach Bakera and Su [10], McMahona and Floriana [100], Carliera [28], Grabowskiego i innych [65]. Rozpoczniemy prezentację głównych wyników od algorytmu (C) Carliera [28].

6.9.1 Podstawowe własności

Opisywany algorytm C generuje dla każdego węzła w drzewie rozwiązań kompletne rozwiązanie, używając do tego algorytmu S. Na bazie tego rozwiązania, określa się zasadę podziału, dolne i górne ograniczenia oraz reguły eliminacji, korzystając z pewnych własności π^S opisanych poniżej.

Niech π^S będzie permutacją wygenerowaną algorytmem S z drogą krytyczną (a, b) odpowiednio do definicji podanej w rozdz. 6.4.

Twierdzenie 6.4 *1. Jeśli π^S nie jest optymalna, to istnieje zadanie $c \in N$ oraz zbiór $K \subseteq N$ takie, że $h(K) > C_{\max}(\pi^S) - p_c$ oraz w rozwiązaniu optymalnym c jest wykonywane przed lub za wszystkimi zadaniami z K.*

2. Jeśli istnieje $I \subseteq N$ taki, że $C_{\max}(\pi^S) = h(I)$, to π^S jest optymalne.

Dowód. Bez straty ogólności możemy założyć, że $\pi^S = \pi$, gdzie $\pi(i) = i$, $i = 1, \ldots, n$. Zgodnie ze wzorem mamy $C_{\max}(\pi) = r_a + p(I) + q_b$, gdzie $I = \{a, a+1, \ldots, b\}$.

Wykażemy następnie kolejno trzy własności. Po pierwsze żadne zadanie nie jest wykonywane w przedziale $(r_a - 1, r_a)$. Przypuśćmy, że istnieje zadanie i rozpoczynające się w $S_i < r_a$ oraz $S_i + p_i > r_a$. Stąd $S_i + p_i + p(I) + q_b > C_{\max}(\pi)$, co przeczy temu, że $C_{\max}(\pi)$ jest maksymalne.

Po drugie $r_a = r(I)$. Jest tak istotnie, ponieważ maszyna jest wolna w przedziale czasu $(r_a - 1, r_a)$. Stąd w algorytmie gdy szeregujemy a w chwili $t = r_a$ mamy $t = r_a = \min_{j \in N \setminus U} r_j$ oraz $r_a = \min_{j \in I} r_j = r(I)$, ponieważ $I \subseteq N \setminus U$.

Po trzecie, jeśli $q_b = q(I)$, to długość ścieżki krytycznej jest równa $C_{\max}(\pi) = r_a + p(I) + q_b = r(I) + p(I) + q(I) = h(I)$, zatem rozwiązanie jest optymalne. Występuje to zawsze, gdy spełniona jest zależność "jeśli $r_i \leq r_j$ to $q_i \geq q_j$". Na przykład, jeśli wszystkie r_j są równe lub wszystkie q_j są równe.

W przypadku przeciwnym do poprzedniego, to znaczy gdy $q_b > q(I)$, istnieje zadanie i takie, że $i < b$ oraz $q_i < q_b$; oznaczmy przez c największy indeks takiego zadania. Zdefiniujmy zbiór $K = \{c+1, \ldots, b\}$. Zgodnie z definicją mamy $q_c < q_j$, $j \in K$. Łatwo zauważyć, że zachodzi $S_c < r_j$, $j \in K$. Istotnie, gdyby zachodziło $S_c \geq r_i$ dla pewnego $i \in K$, to zadanie i byłoby gotowe do wykonywania w chwili czasowej $t = S_c$, w której podejmowano decyzje o uszeregowaniu zadania c, i zadanie to zostałoby uszeregowane w chwili t zamiast c ze względu na warunek $q_j > q_c$. Otrzymujemy zatem nierówność prawdziwą dla każdego $j \in K$

$$r_j > S_c = r_a + \sum_{t=a}^{c-1} p_t \tag{6.68}$$

implikującą oczywistą nierówność

$$r(K) = \min_{j \in K} r_j > S_c = r_a + \sum_{t=a}^{c-1} p_t. \tag{6.69}$$

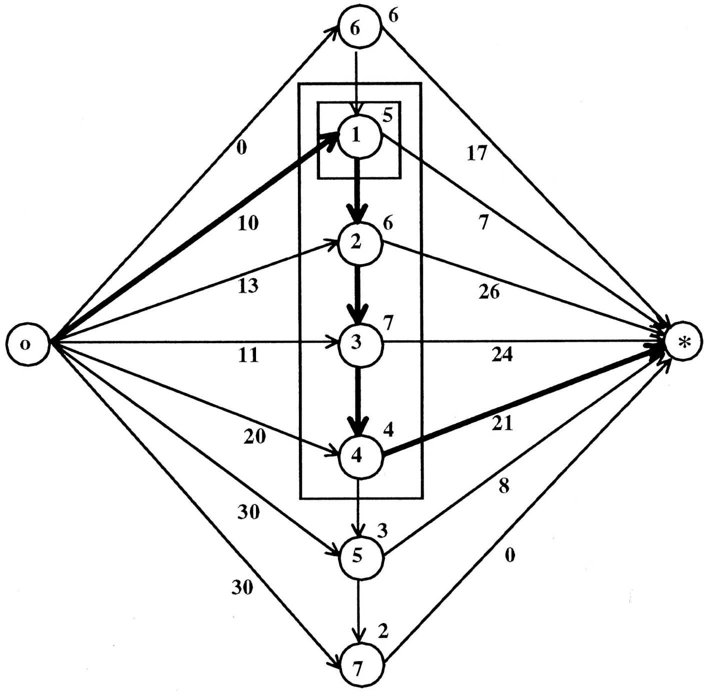
Zgodnie z definicją zbioru K mamy $q_b = \min_{j \in K} q_j = q(K)$, zatem ostatecznie

$$h(K) = r(K) + p(K) + q(K) > r_a + p(I) + q_b - p_c = C_{\max}(\pi) - p_c \tag{6.70}$$

co kończy dowód twierdzenia. ∎

Twierdzenie 6.4 jest zmodyfikowaną wersją bazowego twierdzenia z pracy Carliera [28], z częścią (2) poprawioną w oparciu o spostrzeżenia Nowickiego i Zdrzałki [124]. Odpowiednio, zadanie c oraz zbiór K zdefiniowany w rozdz. 6.4 spełniają warunki twierdzenia 6.4(1).

W przykładzie opisanym w tablicy 6.5 mamy $c = 1$, $K = \{2, 3, 4\}$, $h(K) = 49$, patrz także rysunki 6.3 i 6.4.



Rys. 6.4. Interpretacja zadania c oraz zbioru K na grafie.

6.9.2 Zasada podziału

Z każdym węzłem drzewa rozwiązań kojarzymy:

1. Problem $1|r_j, q_j|C_{\max}$ z danymi $(r_j, p_j, q_j, j = 1, \ldots, n)$; wartości r_j, q_j będą ulegały modyfikacji w czasie pracy algorytmu.

2. Permutację π^S wygenerowaną algorytmem S wraz z odpowiadającą mu wartością funkcji celu $C_{\max}(\pi^S)$, drogą krytyczną (a, b), zadaniem interferencyjnym c oraz zbiorem krytycznym K, zgodnie z definicjami podanymi w rozdz. 6.4.

3. Dolne ograniczenie wartości celu LB dla wszystkich rozwiązań problemu z danymi charakterystycznymi dla tego węzła.

4. Górne ograniczenie UB wartości funkcji celu dla wszystkich rozwiązań sprawdzonych przed analizą rozpatrywanego węzła.

Przyjmujemy początkowo $UB = \infty$. Wyznaczamy π^S poprzez zastosowanie algorytmu S, a następnie odpowiadającą mu wartość funkcji celu $C_{\max}(\pi^S)$, drogę krytyczną (a, b), zadanie interferencyjne c oraz zbiór krytyczny K, zgodnie z definicjami podanymi w rozdz. 6.4. Jeśli c nie istnieje, to otrzymane rozwiązanie jest optymalne dla rozważanego problemu na mocy twierdzenia 6.4 (2). W przeciwnym przypadku kreowane są 2+1 problemy potomne (następniki w drzewie rozwiązań), z których 2 podlegają sprawdzeniu, zaś jeden jest zawsze eliminowany.

Dwa podstawowe problemy potomne są określone następująco:

(A) "Zadanie c jest wykonywane przed wszystkimi zadaniami ze zbioru K". Warunek ten jest uwzględniany poprzez modyfikację danych problemu według zasady

$$q_c := \max\{q_c, p(K) + q_{\pi^S(b)}\}. \tag{6.71}$$

(B) "Zadanie c jest wykonywane za wszystkimi zadaniami ze zbioru K". Warunek ten jest uwzględniany poprzez modyfikację danych problemu według zasady

$$r_c := \max\{r_c, r(K) + p(K)\}. \tag{6.72}$$

Zauważmy, że $p(K) = C_{\pi^S(b)} - C_c$, czyli może być wyznaczony w czasie $O(1)$. Podobnie mamy $q(K) = q_{\pi^S(b)}$.

6.9.3 Górne ograniczenie

Jako górne ograniczenie przyjmuje się $UB := \min\{UB, C_{\max}(\pi^S)\}$ każdorazowo po wygenerowaniu π^S dla wszystkich analizowanych problemów potomnych.

6.9.4 Dodatkowe testy eliminacyjne

Dla rozpatrywanej permutacji π^S tworzony jest zbiór

$$L = \{i \in N \setminus K : p_i > UB - h(K)\}. \tag{6.73}$$

Jeśli $i \in L$, to i musi być wykonywane przed lub za K. Odpowiednio budowane są dwa dodatkowe testy eliminacyjne:

1. Jeśli $i \in L$ oraz

$$r_i + p_i + p(K) + q_{\pi^S(b)} \geq UB \tag{6.74}$$

to i musi być wykonywane za wszystkimi zadaniami z K, co pozwala na wykonanie modyfikacji

$$r_i := \max\{r_i, r(K) + p(K)\} \tag{6.75}$$

2. Jeśli $i \in L$ oraz

$$r(K) + p_i + p(K) + q_i \geq UB \tag{6.76}$$

to i musi być wykonywane przed wszystkimi zadaniami z K, co pozwala na wykonanie modyfikacji

$$q_i := \max\{q_i, q(K) + p(K)\} \tag{6.77}$$

6.9.5 Dolne ograniczenie

Dla każdego węzła używane są dwa dolne ograniczenia: (a) podstawowe $LB = C^*_{\max,pmtn}$, gdzie $C^*_{\max,pmtn}$ jest optymalną wartością funkcji celu problemu zrelaksowanego $1|r_j, q_j, pmtn|C_{\max}$, z danymi r_j, p_j, q_j, $j \in N$, skojarzonymi z tym węzłem, oraz (b) pomocnicze (predyktywne dla następników) $LB(K,c) = \max\{h(K), h(K \cup \{c\}), LB\}$, gdzie wartości $h(K)$, $h(K \cup \{c\})$ są obliczane dla bezpośrednich następników tego węzła, zaś K oraz c są odpowiednio zbiorem krytycznym i zadaniem interferencyjnym stanowiącym podstawę przeprowadzenia podziału.

Węzeł dla którego $LB \geq UB$ jest zamykany (eliminowany). Węzeł potomny, dla którego $LB(K,c) \geq UB$ nie jest generowany (jest zamykany a priori).

6.9.6 Strategia przeglądania

Realizowana jest strategia "w głąb" rozgałęziania węzła niezamkniętego o minimalnym pomocniczym dolnym ograniczeniu. Z wybranego węzła drzewa rozwiązań generowane są co najwyżej dwa węzły potomne, chwilowo oba niezamknięte (odłożone), mające ustalone wstępnie pomocnicze dolne ograniczenia $LB(K,c)$. Po zamknięciu (sprawdzeniu lub wyeliminowaniu) wybranego węzła, kontynuowany jest proces wyboru. Poszukiwania kończą się, gdy wszystkie odłożone węzły zostaną przeanalizowane.

6.9.7 Przykładowy przebieg algorytmu

Rozważmy przebieg algorytmu C dla przykładu liczbowego z rozdz. 6.5. Inicjujemy wartości $\pi^* = ()$, $UB = \infty$.

Iteracja 1. W tej iteracji poszukujemy rozwiązania problemu wyjściowego, startując z danych oryginalnych zawiertych w rozdz. 6.5.

Stosujac algorytm S wyznaczamy $\pi^S = (6,1,2,3,4,5,7)$. Stąd otrzymujemy także $C = (15,21,28,32,35,6,37)$, $C_{\max}(\pi^S) = 53$, $(a,b) = (2,5)$, $c = 1$, $\pi^S(b) = 4$, $K = \{2,3,4\}$. Aktualizacja górnego ograniczenia daje $UB = 53$ dla $\pi^* = (6,1,2,3,4,5,7)$. Zgodnie z definicją mamy $r(K) = \min\{13,11,20\} = 11$, $p(K) = C_{\pi^S(b)} - C_c = C_4 - C_1 = 32 - 15 = 17$, $q(K) = q_{\pi^S(b)} = q_4 = 21$. Kolejno, stosując algorytm S* z rozdz. 6.5 otrzymujemy $C^*_{\max,pmtn} = 49$. Stąd mamy $LB = 49$.

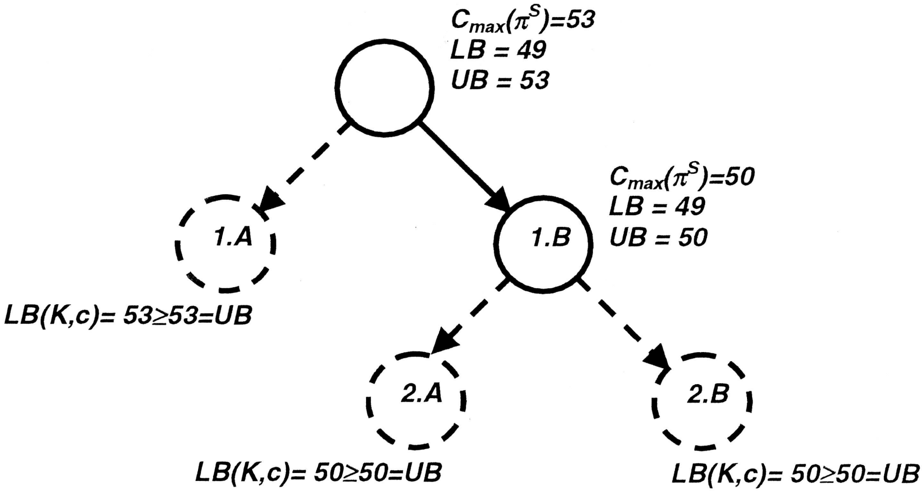
Węzeł nie może zostać zamknięty i będzie podlegał podziałowi generując dwa problemy potomne oznaczone jako 1.A i 1.B, zgodnie z regułą podziału. Problem 1.A otrzymamy przez wprowadzenie dodatkowego ograniczenia $1 \to \{2,3,4\}$, przeniesionego na modyfikację $q_1 = \max\{q_1, p(K) + q_4\} = \max\{7, 17 + 21\} = 38$. Wartość pomocniczego dolnego ograniczenia dla problemu 1.A wynosi $LB(K,c) = \max\{LB, h(K), h(K \cup \{c\})\} = \max\{49, 49, 53\} = 53$. Węzeł odpowiadający problemowi 1.A nie będzie dodany do drzewa bowiem $LB(K,c) \geq UB$. Problem 1.B otrzymamy przez wprowadzenie dodatkowego ograniczenia $\{2,3,4\} \to 1$, przeniesionego na modyfikację $r_1 = \max\{r_1, r(K) + p(K)\} = \max\{11, 11 + 17\} = 28$. Wartość pomocniczego dolnego ograniczenia dla problemu 1.B wynosi $LB(K,c) = \max\{LB, h(K), h(K \cup \{c\})\} = \max\{49, 49, 40\} = 49$.

Stosujac dodatkowe testy eliminacyjne mamy $L = \{i \in \{1,5,6,7\} : p_i > 53 - 49\} = \{1,6\}$. Dla $1 = i \in L$ zachodzi warunek (6.74) bowiem $r_1 + p_1 + p(K) + q_4 = 10 + 5 + 17 + 21 = 53 \geq 53 = UB$, co pozwala na wprowadzenie dodatkowego ograniczenia $K \to i$ poprzez modyfikację (6.75) w formie $r_1 = \max\{r_1, r(K) + p(K)\} = \max\{10, 11 + 17\} = 28$. Dla $1 = i \in L$ warunek (6.76) nie zachodzi. Dla $6 = i \in L$ nie zachodzi ani warunek (6.74) ani (6.76). Ograniczenia uzyskane w wyniku dodatkowych testów eliminacyjnych muszą być uwzględnione przy rozwiązywaniu 1.B.

Do dalszego rozwiązywania wbieramy niezamknięty węzeł o minimalnym pomocniczym dolnym ograniczeniu. W tym przypadku wybieramy 1.B (jest to jedyny niezamknięty węzeł).

Iteracja 2. W tej iteracji poszukujemy rozwiązania problemu 1.B. Startujemy z danych oryginalnych zawierających zmienioną wartość $r_1 = 28$ (w wyniku podziału prowadzącego do 1.B plus dodatkowe testy eliminacyjne towarzyszące podziałowi 1.A-1.B).

Stosujac algorytm S wyznaczamy $\pi^S = (6,3,2,4,1,5,7)$. Stąd otrzymujemy także $C = (33,24,18,28,36,6,38)$, $C_{\max}(\pi^S) = 50$, $(a,b) = (2,3)$, $c = 3$, $\pi^S(b) = 2$, $K = \{2\}$. Aktualizacja górnego ograniczenia daje $UB = 50$ dla $\pi^* = (6,3,2,4,1,5,7)$, $C^* = (33,24,18,28,36,6,38)$. Zgodnie z definicją mamy $r(K) = 13$, $p(K) = C_{\pi^S(b)} - C_c = C_2 - C_3 = 24 - 18 = 6$, $q(K) = q_{\pi^S(b)} = q_2 = 26$. Kolejno, stosując algorytm S* z rozdz. 6.5



Rys. 6.5. Drzewo rozwiązań algorytmu C dla przykładu z rozdz. 6.5

otrzymujemy $C^*_{\max,pmtn} = 49$. Stąd mamy $LB = 49$.

Węzeł nie może zostać zamknięty i będzie podlegał podziałowi generując dwa problemy potomne oznaczone jako 2.A i 2.B, zgodnie z regułą podziału. Problem 2.A otrzymamy przez wprowadzenie dodatkowego ograniczenia $3 \rightarrow \{2\}$, przeniesionego na modyfikację $q_3 = \max\{q_3, p(K) + q_2\} = \max\{24, 6 + 26\} = 32$. Wartość pomocniczego dolnego ograniczenia dla problemu 2.A wynosi $LB(K, c) = \max\{LB, h(K), h(K \cup \{c\})\} = \max\{49, 45, 50\} = 50$. Węzeł odpowiadający problemowi 2.A nie będzie dodany do drzewa bowiem $LB(K, c) \geq UB$. Problem 2.B otrzymamy przez wprowadzenie dodatkowego ograniczenia $\{2\} \rightarrow 3$, przeniesionego na modyfikację $r_3 = \max\{r_3, r(K) + p(K)\} = \max\{11, 13 + 6\} = 19$. Wartość pomocniczego dolnego ograniczenia dla problemu 2.A wynosi $LB(K, c) = \max\{LB, h(K), h(K \cup \{c\})\} = \max\{49, 45, 50\} = 50$. Węzeł odpowiadający problemowi 2.B nie będzie dodany do drzewa bowiem $LB(K, c) \geq UB$.

Wszystkie odłożone problemy zostały rozwiązane. Algorytm skończył działanie. Mamy $\pi^* = (6, 3, 2, 4, 1, 5, 7)$, $C_{\max}(\pi^*) = UB = 50$. Drzewo rozwiązań wygenerowane przez algorytm C pokazano na rys. 6.5; linią przerywaną zaznaczono węzły eliminowane a priori.