9.9 Algorytmy przeglądu

Metody przeglądu dla problemu $1|r_j, q_j|C_{\max}$ były analizowane między innymi w pracach [25,51,129,238]. Jak dotychczas najbardziej elegancka wersja pochodzi z pracy [51] i jest porównywalna, w sensie efektywności, z nieco bardziej skomplikowanym algorytmem blokowym [129].

9.9.1 Algorytm Carliera

Opisywany algorytm jest schematem B&B, który generuje dla każdego węzła w drzewie rozwiązań kompletne rozwiązanie używając do tego celu algorytmu S. Na bazie tego rozwiązanie określana jest zasada podziału, dolne i górne ograniczenia oraz reguły eliminacji, korzystając z pewnych własności π^S opisanych poniżej.

Podstawowe własności

Niech π^S będzie permutacją wygenerowaną Algorytmem S z drogą krytyczną (a, b) odpowiednio do definicji podanej w Rozdz. 9.4. Zachodzą dwa wyłączające się wzajemnie przypadki: (1) jeśli π^S nie jest optymalna to istnieje zadanie $c \in \mathcal{J}$ oraz zbiór $K \subseteq \mathcal{J}$ 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 π^S jest optymalna to istnieje $I \subseteq \mathcal{J}$ taki, że $C_{\max}(\pi^S) = h(I)$.

Istotnie, w celu dowodu własności, bez straty ogólności możemy założyć, że $\pi^S = \pi$ gdzie $\pi(i) = i$, $i = 1, \ldots, n$. Zatem zgodnie ze wzorem mamy $C\max(\pi) = r_a + p(I) + q_b$, gdzie $I = \{a, a+1, \ldots, b\}$.

Wpierw pokażemy, że żadne zadanie nie jest wykonywane w przedziale $(r_a - 1, r_a)$. Przypuśćmy, że istnieje 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. Dalej pokażemy, że $r_a = r(I)$. 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 $t = r_a = \min_{j \in \mathcal{J} \setminus U} r_j$ oraz $r_a = \min_{j \in I} r_j = r(I)$ ponieważ $I \subseteq \mathcal{J} \setminus U$. Wreszcie, 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 $r_i \leqslant r_j$ to $q_i \geqslant q_j$". Przykładowo, jeśli wszystkie r_j sa 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 $S_c \geqslant r_i$ 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 $q_j > q_c$. Zatem otrzymujemy nierówność prawdziwą dla każdego $j \in K$

$$r_j > S_c = r_a + \sum_{t=a}^{c-1} p_t \tag{9.54}$$

implikującą oczywistą nierówność

$$r(K) = \min_{j\in K} r_j > S_c = r_a + \sum_{t=a}^{c-1} p_t. \tag{9.55}$$

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

$$h(K) = r(K) + p(K) + q(K) > r_a + p(I) + q_b - p_c = C_{\max}(\pi) - p_c \tag{9.56}$$

co kończy dowód własności.

Odpowiednio, zadanie c oraz zbiór K zdefiniowany w Rozdz. 9.4 spełniają warunki podanej własności. W przykładzie opisanym w Tablicy 9.7.1 mamy $c = 1$, $K = \{2,3,4\}$, $h(K) = 49$, patrz także Rys. 9.3, 9.5.

Zasada podziału

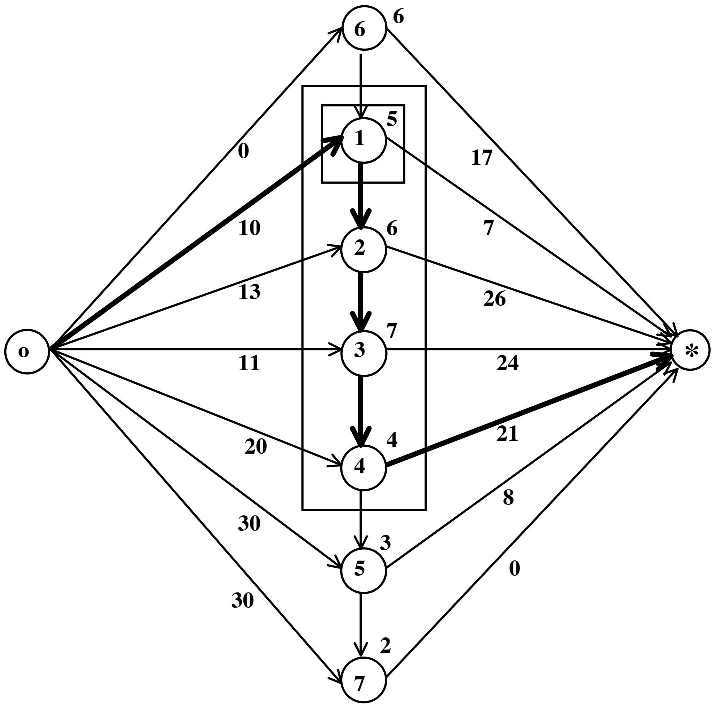
Z każdym węzłem drzewa rozwiązań kojarzymy:

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. 9.4

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.



Rysunek 9.5: Interpretacja zadania c oraz zbioru K na grafie.

Przyjmujemy początkowo $UB = \infty$. Wyznaczamy π^S poprzez zastosowanie Algorytmu S, a w dalszej kolejności 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. 9.4. Jeśli c nie istnieje to otrzymane rozwiązanie jest optymalne dla rozważanego problemu. 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
$$q_c := \max\{q_c, p(K) + q_b\}. \tag{9.57}$$

(B) "zadanie c jest wykonywane za wszystkimi zadaniami ze zbioru K". Warunek ten jest uwzględniany poprzez modyfikację danych problemu według zasady
$$r_c := \max\{r_c, r(K) + p(K)\}. \tag{9.58}$$

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

Górne ograniczenie

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

Eliminacja

Dla rozpatrywanej permutacji π^S tworzony jest zbiór
$$L = \{i \in \mathcal{J} \setminus K : \; p_i > UB - h(K)\}. \tag{9.59}$$

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
$$r_i + p_i + p(K) + q_b \geqslant UB \tag{9.60}$$
to i musi być wykonywane za wszystkimi zadaniami z K co pozwala na wykonanie modyfikacji
$$r_i := \max\{r_i, r(K) + p(K)\} \tag{9.61}$$

2. Jeśli $i \in L$ oraz
$$r(K) + p_i + p(K) + q_i \geqslant UB \tag{9.62}$$
to i musi być wykonywane przed wszystkimi zadaniami z K co pozwala na wykonanie modyfikacji
$$q_i := \max\{q_i, q(K) + p(K)\} \tag{9.63}$$

Dolne ograniczenie

Proponuje się zastosować
$$LB = \max\{C^*_{\max,pmtn}, h(K), h(K \cup \{c\})\} \tag{9.64}$$
gdzie $C^*_{\max,pmtn}$ jest optymalną wartością funkcji celu problemu postaci $1|r_j, q_j, pmtn|C_{\max}$ dla danych r_j, q_j, $j = 1, \ldots, n$ skojarzonych z analizowanym węzłem.

Strategia przeglądania

Realizowana jest strategia "w głąb". Z bieżącego węzła drzewa rozwiązań generowane są co najwyżej dwa węzły potomne w kolejności wynikającej z pomocniczego dolnego ograniczenia. Wybrany węzeł staje się bieżącym, zaś ten drugi staje się chwilowo odłożonym. Po zamknięciu (sprawdzeniu lub wyeliminowaniu) bieżącego węzła kontynuowany jest proces analizy węzłów poczynając od ostatnio odłożonego węzła. Poszukiwania kończą się gdy wszystkie odłożone węzły zostaną przeanalizowane.

9.9.2 Algorytm blokowy

Bezpośrednie wykorzystanie grafowego modelu problemu oraz wzoru (9.12) pozwala na otrzymanie pewnych teoretycznych własności użytecznych, między innymi, przy konstrukcji algorytmów dokładnych opartych na schemacie B&B. Własności te stanowią podstawę *podejścia blokowego*, oryginalnego kierunku badawczego opracowanego pierwotnie w ICT PWr, a następnie stosowanego z powodzeniem w wielu innych ośrodkach zagranicznych. Uniwersalność proponowanego podejścia czyni je przydatnym zarówno w algorytmach dokładnych jak i metodach przybliżonych.

Podstawowe własności

Niech π będzie pewną permutacją z drogą krytyczną (a, b), patrz Rozdz. 9.4. Ciąg zadań $B = (\pi(a), \ldots, \pi(b))$ będziemy nazywać *blokiem* zadań w π.