dla każdej pary $(i, j) \in \mathcal{R}$. Odpowiednie problemy posiadają oznaczenie $1|r_j, prec|C_{\max}$, $1|prec|L_{\max}$ oraz $1|q_j, prec|C_{\max}$. Właściwe algorytmy rozwiązywania, o złożoności $O(n^2)$, można otrzymać tworząc uporządkowanie *topologiczne* odpowiednich wielkości r_j, d_j lub q_j. (Przez uporządkowanie topologiczne liczb r_j, $j \in \mathcal{J}$, z relacją $\mathcal{R}$ rozumiemy permutację π taką, że dla każdej pary $(\pi(i), \pi(j))$, $i < j$, dla której nie istnieje droga w grafie $(\mathcal{J}, \mathcal{R})$ zachodzi $r_{\pi(i)} \leqslant r_{\pi(j)}$.) W praktyce korzysta się jednak z innego podejścia. Relacja $\mathcal{R}$ w sposób naturalny implikuje zależności pomiędzy wartościami r_j, d_j, q_j pozwalające dokonać modyfikacji danych wejściowych dla każdej pary $(i, j) \in \mathcal{R}$, w najprostszym przypadku według poniższych zasad

$$r_j := \max\{r_j, r_i + p_i\}, \tag{9.5}$$

$$q_i := \max\{q_i, q_j + p_j\}, \tag{9.6}$$

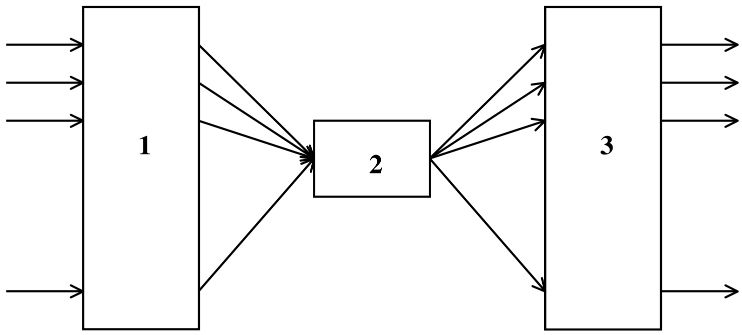
$$d_j := \min\{d_j, d_i - p_i\}. \tag{9.7}$$

bez utraty rozwiązania optymalnego. Modyfikacja ma charakter progresywny, bowiem dokonane zmiany wartości danych mają wpływ na wynik zmian następnych. Zatem jest wykonywana zgodnie z uporządkowaniem liniowym $\mathcal{R}$) od początku dla (9.5) oraz od końca dla (9.6) i (9.7), można ją wykonać w czasie rzędu $O(\max\{n, |\mathcal{R}|\})$. W konsekwencji, każdy z algorytmów opisanych w rozdz. 2.2 zachowuje swą ważność, to znaczy dostarcza rozwiązanie dopuszczalne w sensie $\mathcal{R}$ i optymalne, dla odpowiednich przypadków uwzględniających zadania zależne.

9.4 Czasy przygotowania i dostarczenia

Problem został otrzymany z problemu podstawowego (Rozdz. 9.1) poprzez wprowadzenie niezerowych czasów przygotowawczych r_j implikujących ograniczenie $r_j \leqslant S_j$ oraz czasów dostarczenia zadania $q_j \geqslant 0$. Dostarczenie rozpoczyna się bezpośrednio po zakończeniu wykonywania zadania, przy czym wszystkie zadania mogą byś równocześnie dostarczane. Celem jest wyznaczenie harmonogramu pracy maszyny, który minimalizuje czas w jakim wszystkie zadania zostaną dostarczone. Problem ten jest oznaczany przez $1|r_j, q_j|C_{\max}$.

Problem $1|r_j, q_j|C_{\max}$ jest równoważny problemowi z czasami przygotowawczymi, żądanymi terminami zakończenia oraz kryterium minimalizacji maksymalnego spóźnienia $L_{\max}$ oznaczonego $1|r_j|L_{\max}$ używając notacji z pracy [139]. Istotnie, niech $D = \min_{1 \leqslant j \leqslant n} d_j$. Stąd mamy $q_j = D - d_j \geqslant 0$



Rysunek 9.1: Struktura systemu wytwarzania dla problemu $1|r_j, q_j|C_{\max}$.

oraz

$$L_{\max} = \max_{1 \leqslant j \leqslant n} (C_j - d_j) = \max_{1 \leqslant j \leqslant n} (C_j + q_j) - D = C_{\max} - D \tag{9.8}$$

Problem $1|r_j, q_j|C_{\max}$ jest równoważny problemowi szeregowania zadań na trzech maszynach, z których maszyna 2 jest maszyną o ograniczonej jednostkowej przepustowości z czasami wykonywania zadań p_j, zaś maszyny 1 i 3 są maszynami o nieograniczonej przepustowości z czasami wykonywania r_j oraz q_j odpowiednio. Przyjmujemy, że maszyny o nieograniczonej przepustowości mogą wykonywać dowolnie dużo zadań jednocześnie, zatem zadania na tych maszynach nie muszą być szeregowane (kolejkowane) [224], patrz także Rys. 9.1. Logiczne pojęcie maszyny o nieograniczonej przepustowości może modelować m.in.: (a) fizyczne urządzenia o nieograniczonych możliwościach wykonawczych (np. piec grzewczy), (b) fizyczne urządzenia o ograniczonych możliwościach wykonawczych jednakże nie wymagające szeregowania (np. dzięki znikomo krótkim czasom wykonywaniu zadań na tym urządzeniu, brakowi kolejki, niewystępowaniu konfliktów w dostępie zadań do urządzenia), (c) proces wymagający upływu czasu lecz nie angażujący urządzenia fizycznego (np. dojrzewanie, stygnięcie), (d) zagregowany zbiór maszyn o nieograniczonej przepustowości.

Chociaż wszystkie powyższe sformułowania problemu są wzajemnie równoważne, model $1|r_j, q_j|C_{\max}$ jest używany najczęściej ze względu na *autosymetrię*. Oznacza ona, w tym przypadku, że problem otrzymany przez zamianę miejscami wartości r_j z q_j posiada dokładnie taką samą optymalną wartość funkcji celu, osiąganą dla permutacji odwrotnej do tej w problemie przed zamianą.

Problemowi $1|r_j, q_j|C_{\max}$ poświęcono znaczną uwagę w ostatnim dziesięcioleciu ze względu na liczne zastosowania m.in. w szeregowaniu maszyn krytycznych [224], w algorytmach przybliżonych dla problemu gniazdowego [5]), jako dolne ograniczenie dla problemów przepływowych i gniazdowych [43,52].

W pracy [225] pokazano, że problem jest silnie NP-trudny, jednakże dla pewnych przypadków szczególnych istnieją algorytmy wielomianowe.

Oznaczmy przez $C_{\max}(\pi)$ czas, w którym wszystkie zadania zostaną dostarczone dla rozwiązania reprezentowanego permutacją π. Terminy rozpoczęcia S_j i zakończenia C_j wykonywania zadań można znaleźć z zależności rekurencyjnej

$$C_{\pi(j)} = \max\{r_{\pi(j)}, C_{\pi(j-1)}\} + p_{\pi(j)}, \quad j = 1, \ldots, n, \tag{9.9}$$

gdzie $\pi(0) = 0$ oraz $C_0 = 0$. Dodatkowo $S_j = C_j - p_j$, $j = 1, \ldots, n$. Wartość funkcji kryterialnej wynosi

$$C_{\max}(\pi) = \max_{1 \leqslant j \leqslant n} (C_{\pi(j)} + q_{\pi(j)}). \tag{9.10}$$

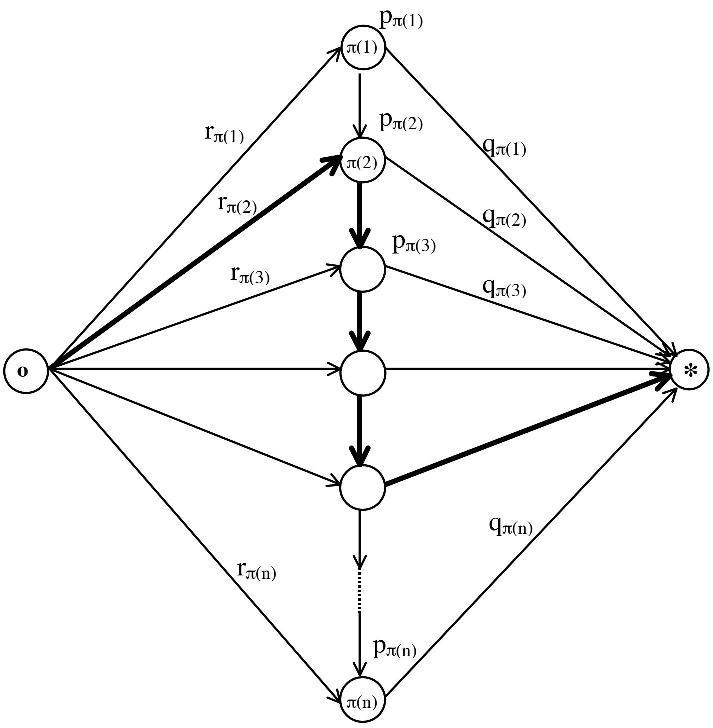
Przy wyprowadzaniu pewnych własności problemu będziemy się także odwoływać do nierekurencyjnej wersji wzoru (9.9)

$$C_{\pi(j)} = \max_{1 \leqslant u \leqslant v \leqslant j} (r_{\pi(u)} + \sum_{s=u}^{v} p_{\pi(s)}) \tag{9.11}$$

oraz odpowiedniej nierekurencyjnej wersji wzoru (9.10)

$$C_{\max}(\pi) = \max_{1 \leqslant u \leqslant v \leqslant n} (r_{\pi(u)} + \sum_{s=u}^{v} p_{\pi(s)} + q_{\pi(v)}) \tag{9.12}$$

Wzory (9.9)–(9.12) posiadają wygodną interpretację grafową. Dla permutacji π konstruujemy graf skierowany $G(\pi) = (\mathcal{J} \cup \{o, *\}, E \cup E(\pi))$, gdzie $\mathcal{J}$ jest zbiorem węzłów powiększonym o dwa węzły sztuczne o (początek) oraz $*$ (koniec), patrz rys. 9.2. Węzły $j \in \mathcal{J}$ posiadają wagę p_j zaś węzły sztuczne – wagę zero. Zbiór krawędzi $E = \bigcup_{j \in \mathcal{J}} \{(o, j), (j, *)\}$ posiada wagi r_j dla krawędzi (o, j) oraz q_j dla krawędzi $(j, *)$ odpowiednio. Zbiór $E(\pi) = \bigcup_{j=1}^{n-1} \{(\pi(j), \pi(j+1))\}$ posiada krawędzie z zerowym obciążeniem. Wartość C_j jest długością najdłuższej drogi prowadzącej do węzła j (wraz z



Rysunek 9.2: Graf $G(\pi)$ dla problemu $1|r_j, q_j|C_{\max}$.

obciążeniem tego węzła) zaś $C_{\max}(\pi)$ jest długością najdłuższej drogi w tym grafie.

Permutację π^*, która minimalizuje $\min_\pi C_{\max}(\pi)$ nazywamy optymalną. Niech $C^* = C_{\max}(\pi^*)$. Każdą parę liczb całkowitych (u, v), $1 \leqslant u \leqslant v \leqslant n$, występujących w prawej stronie wzoru (9.12) będziemy nazywać drogą w π. Para ta całkowicie określa przebieg drogi w odpowiednim grafie $G(\pi)$. Drogę (a, b), dla której

$$a = \min\{u : C_{\max}(\pi) = r_{\pi(u)} + \sum_{s=u}^{b} p_{\pi(s)} + q_{\pi(b)})\} \tag{9.13}$$

będziemy nazywać drogą krytyczną w π. Zadanie $c = \pi(k')$ takie, że $a \leqslant k' < b$, $q_{\pi(k')} < q_{\pi(b)}$ oraz $q_{\pi(k)} \geqslant q_{\pi(b)}$, $k = k' + 1, ..., b$ będzie nazywane zadaniem *interferencyjnym.* Zbiór krytyczny jest definiowany jako $K = \{\pi(k'+1), \ldots, \pi(b)\}$ jeśli k' istnieje oraz $K = \{\pi(a), \ldots, \pi(b)\}$ w przeciwnym przypadku.

Problem $1|r_j, q_j, prec|C_{\max}$ zawierający zadania zależne poprzez relację $\mathcal{R}$, może być rozwiązywany metodami podobnymi jak $1|r_j, q_j|C_{\max}$. Jednak często, dla prostoty rozważań, relację $\mathcal{R}$ eliminuje się z problemu wykonując odpowiednią modyfikację danych wejściowych, analogicznie do opisanej w Rozdz. 9.3. Postępowanie takie nie powoduje utraty rozwiązania optymalnego jednak niesie ze sobą pewne niebezpieczeństwa spowodowane "nieszczelnością" systemu modyfikacji. Pierwszym z nich jest wprowadzenie do zbioru rozwiązań permutacji niedopuszczalnych z punktu widzenia relacji (dla problemu $1|r_j, q_j|C_{\max}$ wszystkie permutacje są dopuszczalne). Drugim - jako konsekwencja pierwszego wymienionego, jedynie w przypadku istnienia wielu permutacji optymalnych – pewna optymalna lecz niedopuszczalna permutacja może zostać zwrócona w wyniku działania algorytmu. O ile pierwszy z wymienionych problemów jest tylko pewną niedogodnością (co najwyżej czas pracy algorytmu zostanie wydłużony), to drugi stanowi już poważny mankament. W niektórych algorytmach (dokładnych i przybliżonych) sposób generacji rozwiązania pozwala "automatycznie" zapobiec drugiemu problemowi już po spełnieniu warunku (9.5) lub (9.6). Dla przeciwdziałania pierwszemu mankamentowi, to jest w celu eliminacji maksymalnie dużej liczby permutacji niedopuszczalnych, wprowadza się bardziej zaawansowane sposoby modyfikacji takie jak na przykład

$$r_j := \max\{r_j, \max_{i \in B_j}(r_i + p_i), r(B_j) + p(B_j)\}, \tag{9.14}$$

$$q_i := \max\{q_i, \max_{i \in A_j}(q_j + p_j), p(A_j) + q(A_j)\}, \tag{9.15}$$

gdzie $B_j = \{i \in \mathcal{J} : (i,j) \in \mathcal{R}\}$ jest zbiorem bezpośrednich poprzedników zadania j, zaś $A_j = \{i \in \mathcal{J} : (j,i) \in \mathcal{R}\}$ jest zbiorem bezpośrednich następników zadania j. Pozostałe wielkości występujące we wzorach (9.14)–(9.15) są określone następująco: $p(I) = \sum_{s \in I} p_s$, $r(I) = \min_{s \in I} r_s$, $q(I) = \min_{s \in I} q_s$. Złożoność obliczeniowa wykonania modyfikacji jest rzędu $O(\max\{n, |\mathcal{R}|\})$ podobnie jak w Rozdz. 9.3 jednak pracochłonność jest praktycznie większa. W różnych konkretnych algorytmach stosowane są także inne specyficzne, mniej i bardziej złożone, sposoby modyfikacji, które jednakże będą przedstawiane łącznie z każdym opisywanym algorytmem.

9.5 Zadania przerywalne

Dopuszczenie przerywania wykonywania zadań dla problemów opisanych w Rozdz. 9.1 nie zmienia nic w zakresie możliwości ich rozwiązania; odpowiednie algorytmy rozwiązują również te przypadki.

W przypadku NP-trudnego problemu z Rozdz. 9.4 sytuacja jest odmienna. Dopuszczenie przerywania wykonywania zadań jest relaksacją prowadzącą do otrzymania problemu $1|r_j, q_j, pmtn|C_{\max}$ posiadającego algorytm wielomianowy. Możliwość rozwiązania w czasie wielomianowym istnieje już dla zagadnienia nieco ogólniejszego od wymienionego, a mianowicie $1|r_j, prec|f_{\max}$ [27] bez znaczącej zmiany złożoności obliczeniowej. Ponieważ oba problemy mają istotne znaczenie dla budowy i działania bardziej złożonych algorytmów, metoda rozwiązania zostanie opisana szczegółowo dla drugiego z wymienionych jako ogólniejszego. Specjalizowany algorytm dla problemu $1|r_j, q_j, pmtn|C_{\max}$ podano w Rozdz. 9.7.1.

Ze względu na przerywanie wykonywania zadań rozwiązanie nie może być reprezentowane jedynie i jednoznacznie permutacją. Stąd przez uszeregowanie $S = (S_1, \ldots, S_n)$ będziemy rozumieć przyporządkowanie dla każdego zadania ciągu przedziałów czasowych $S_j = ((S_j^i, C_j^i) : i = 1, \ldots, l_j)$ w których zadanie to jest wykonywane. Oczywiście dla uzyskania dopuszczalności muszą być spełnione warunki $\sum_{i=1}^{l_j}(C_j^i - S_j^i) = p_j$, $r_j \leqslant S_j^1$, $C_i^{l_i} \leqslant S_j^1$ dla $(i,j) \in R$, oraz maszyna wykonuje co najwyżej jedno zadanie w dowolnej chwili czasowej.

Rozpatrzmy wpierw prostszy problem $1|prec|f_{\max}$ z relacją poprzedzania R. Niech $I \subseteq \mathcal{J}$ oznacza dowolny podzbiór zadań, zaś niech $I' \subseteq I$ będzie zbiorem zadań nie posiadających następników w I. Oznaczmy przez $p(I) = \sum_{j \in I} p_j$, przez $f_{\max}(I; S) = \max_{j \in I} f_j(C_j^{l_j})$ wartość funkcji celu dla zadań ze zbioru I oraz uszeregowania S określonego na tym zbiorze, zaś przez $f^*_{\max}(I) = \min_S f_{\max}(I; S)$ wartość funkcji celu dla uszeregowania optymal-