

Wojciech BOŻEJKO¹
Jarosław PEMPERA¹
Czesław SMUTNICKI²
Mariusz UCHROŃSKI¹
Mieczysław WODECKI³

¹Katedra Automatyki, Mechatroniki i Systemów Sterowania, Politechnika Wrocławska

²Katedra Informatyki Technicznej, Politechnika Wrocławska

³Katedra Telekomunikacji i Teleinformatyki, Politechnika Wrocławska

DOKŁADNY ALGORYTM STEROWANY METAHEURYSTYKĄ KWANTOWEGO WYŻARZANIA ROZWIĄZYWANIA JEDNOMASZYNOWEGO PROBLEMU SZEREGOWANIA ZADAŃ

Streszczenie. W pracy przedstawiamy nowe podejście do rozwiązywania *NP*-trudnych problemów optymalizacji dyskretnej dostosowane do najnowszej architektury procesorów kwantowych QPU firmy D-Wave realizujących sprzętowo kwantowe wyżarzanie. Proponujemy hybrydowy algorytm optymalny oparty na metodzie podziału i ograniczeń, w którym zarówno dolne, jak i górne oszacowanie wartości funkcji celu jest wyznaczane przez metaheurystykę realizowaną na QPU. Mimo tego, że wyniki generowane przez maszynę kwantową mają charakter probabilistyczny, to jednak zaproponowana w pracy hybrydowa metoda konstrukcji algorytmu, wykorzystująca CPU i QPU, daje gwarancję optymalności uzyskanego rozwiązania. Jako studium przypadku rozpatrujemy *NP*-trudny jednomaszynowy problem szeregowania zadań z minimalizacją ważonej liczby zadań spóźnionych.

OPTIMAL QUANTUM ANNEALING METAHEURISTICS-DRIVEN ALGORITHM FOR A SINGLE MACHINE SCHEDULING PROBLEM

Summary. In the paper, we propose a new approach to solving *NP*-hard optimization problems using the latest architecture of D-Wave quantum QPU processors that implement quantum annealing in hardware. We propose the optimal hybrid branch and bound algorithm, in which both the lower and upper bounds of the cost function value are determined by the metaheuristics implemented on the QPU. Despite the fact that the results generated by the quantum machine are probabilistic, the hybrid method of algorithm designing proposed in the paper, using both the CPU and QPU, guarantees the optimality of the obtained solution. As a case study, we consider the *NP*-hard single machine scheduling problem with minimizing the weighted number of tardy jobs.

1. Wprowadzenie

Koncepcja obliczeń i komputerów kwantowych pojawiła się w latach 80. XX wieku. Aktualnie dostępne są komputery reprezentujące jedno z dwóch podejść do zagadnienia obliczeń kwantowych. Pierwsze, oferowane przez takie firmy jak Google, Honeywell, IBM i Intel, to komputery kwantowe z modelami bramek kwantowych (np. Hadamarda i Toffoli). W przeciwieństwie do wielu klasycznych bramek logicznych, bramki logiki kwantowej są odwracalne. Programowanie w tym modelu obliczeń kwantowych jest nadal dużym wyzwaniem ze względu na małą skalę rozwiązywalnych problemów oraz brak podejścia wysokopoziomowego, adekwatnego do języków wysokiego poziomu w programowaniu klasycznych komputerów krzemowych.

Drugie podejście, wyżarzanie kwantowe (*quantum annealing*, [1]), poprzez wykorzystanie efektów znanych jako fluktuacje kwantowe oraz tunelowanie kwantowe, wyznacza możliwe najlepsze rozwiązanie problemu optymalizacji. Firma D-Wave Systems udostępnia komputery kwantowe, proponując podejście do obliczeń ograniczone co prawda tylko do korzystania z kwantowego wyżarzania, ale za to doskonale wpisujące się w potrzeby dyscypliny badań operacyjnych. W tym przypadku, zamiast wyrażać algorytm rozwiązywania badanego problemu w postaci bramek kwantowych, użytkownik przedstawia go jako problem binarny programowania kwadratowego.

W pracy jest rozpatrywany jeden z najprostszych w sformułowaniu (choć *NP*-trudny) problem szeregowania zadań na jednej maszynie. Dany jest zbiór zadań $\mathcal{J} = \{1, 2, \dots, n\}$ które należy, bez przerywania, wykonać na jednej maszynie. W dowolnym momencie maszyna może wykonywać co najwyżej jedno zadanie. Z każdym zadaniem $i \in \mathcal{J}$ są związane: czas wykonywania p_i , linia krytyczna d_i , oraz waga funkcji kary w_i . Dla ustalonej kolejności wykonywania zadań na maszynie, niech C_i będzie momentem zakończenia wykonywania zadania $i \in \mathcal{J}$. Wówczas, spóźnienie $U_i = 1$, gdy $C_i > d_i$ lub 0 w przeciwnym przypadku. Wielkość $w_i U_i$ jest kosztem spóźnienia (wykonania zadania po terminie). Jednomaszynowy problem minimalizacji sumy kosztów zadań spóźnionych (ang. *Weighted Number of Tardy Jobs*, w skrócie WNTJ) polega na wyznaczeniu takiej kolejności wykonywania zadań, która minimalizuje sumę kosztów spóźnień, tj. $\sum_{i=1}^n w_i U_i$. W literaturze jest on oznaczany przez $1 || \sum w_i U_i$ i pomimo prostoty sformułowania należy do klasy problemów *NP*-trudnych (Karp [3]). Jest to jeden z wielu badanych od lat jednomaszynowych problemów szeregowania zadań. W literaturze rozpatrywanych jest wiele jego wariantów, głównie o wielomianowej złożoności obliczeniowej.

Dla problemu $1 | p_i = 1 | \sum w_i U_i$ (wszystkie czasy wykonywania zadań są jednakowe) Monma w pracy [8] przedstawił algorytm o złożoności $O(n)$. Podobnie, dla problemu $1 | w_i = c | \sum U_i$ z jednakowymi współczynnikami funkcji kosztów, istnieje algorytm Moore'a [9] o złożoności $O(n \ln n)$. Lawler [5] zaadaptował algorytm Moore'a do rozwiązywania problemu $1 | p_i < p_j \Rightarrow w_j \geq w_i | \sum w_i U_i$. Osobną grupę stanowią zagadnienia z najwcześniejszymi możliwymi terminami rozpoczęcia r_i . Kise i in. [4] udowodnili, że już problem minimalizacji liczby zadań spóźnionych ($1 | r_i | \sum U_i$, tj. bez wag funkcji kosztów) jest silnie *NP*-trudny. Przedstawili oni także algorytm wielomianowy o złożoności $O(n^2)$ dla pewnego szczególnego przypadku, tj. problemu $1 | r_i < r_j \Rightarrow d_i \leq d_j | \sum U_i$. Jeżeli na zbiorze zadań określona jest relacja częściowego porządku, to problem WNTJ jest silnie *NP*-trudny nawet, gdy czasy wykonywania zadań

są jednostkowe (Garey i Johnson [2]). Z kolei Lenstra i Rinnoy Kan [7] udowodnili, że jeżeli relacja częściowego porządku jest sumą niezależnych łańcuchów, to problem jest także silnie NP -trudny. W literaturze opublikowano zaledwie kilka algorytmów optymalnych rozwiązywania problemu WNTJ. Są one oparte na metodzie programowania dynamicznego (Lawler i Moore [9] – złożoność $O(n \min\{\sum p_i, \max\{d_i\}\})$) oraz Sahni [11] – złożoność $O(n \min\{\sum p_i, \sum w_i, \max\{d_i\}\})$, a także na metodzie podziału i ograniczeń (Villarreal i Bulfin [12], Potts i Van Wassenhove [10]).

W pracy przedstawiamy hybrydowy algorytm rozwiązywania problemu WNTJ którego konstrukcja jest oparta na metodzie podziału i ograniczeń (*Branch and Bound*, B&B). Proponujemy nowe podejście, polegające na wyznaczeniu górnego i dolnego ograniczenia funkcji celu na komputerze kwantowym. Ponadto, jako kryterium wyboru kolejnego wierzchołka w drzewie rozwiązań, stosujemy górne ograniczenie. Do wyznaczenia górnego ograniczenia na komputerze D-Wave formułujemy zadanie binarnego programowania kwadratowego z ograniczeniami, które jest naturalnym sposobem formułowania zadań obliczeniowych dla tej maszyny. Natomiast, przy wyznaczeniu dolnego ograniczenia, stosujemy metodę relaksacji Lagrange’a. Przybliżenie wartości minimalnej funkcji Lagrange’a wyznaczamy na komputerze kwantowym.

2. Metoda rozwiązania

Rozwiązania problemu WNTJ mogą być reprezentowane przez permutacje elementów zbioru zadań \mathcal{J} . Niech Φ będzie zbiorem wszystkich takich permutacji. Koszt (waga) permutacji $\pi \in \Phi$, $F(\pi) = \sum_{i=1}^n w_{\pi(i)} U_{\pi(i)}$, gdzie czas zakończenia zadania $\pi(i) \in \mathcal{J}$, $C_{\pi(i)} = \sum_{j=1}^i p_{\pi(j)}$, dla $i = 1, 2, \dots, n$. Rozpatrywany problem minimalizacji sumy kosztów zadań spóźnionych sprowadza się więc do wyznaczenia permutacji optymalnej $\pi^* \in \Phi$ czyli takiej, dla której $F(\pi^*) = \min\{F(\beta) : \beta \in \Phi\}$.

Proces przeglądania elementów zbioru rozwiązań Φ będzie przedstawiony za pomocą skierowanego *drzewa rozwiązań* \mathcal{H} . Korzeniem drzewa (jedynym wierzchołkiem na poziomie zero) jest dowolna permutacja (startowa) $\pi \in \Phi$, w której wszystkie zadania są wolne. Z korzenia można wygenerować n nowych wierzchołków (permutacji) na poziomie pierwszym. Każdy z nich powstaje przez ustalenie na pozycji n -tej jednego zadania. W każdej z tych permutacji jest ustalone dokładnie jedno zadanie. Podobnie, z dowolnej permutacji pierwszego poziomu można wygenerować, przez ustalenie zadania wolnego na wolnej $n-1$ -szej pozycji, $n-1$ nowych permutacji tworzących poziom 2. drzewa. Pełne drzewo ma więc n poziomów. Na ostatnim n -tym poziomie drzewa, jest $n!$ permutacji (wierzchołków) i w każdej z nich wszystkie zadania są ustalone.

Jeżeli permutacja β została wygenerowana z π przez ustalenie pewnego zadania wolnego na wolnej pozycji, to para (π, β) tworzy w drzewie \mathcal{H} łuk. Mówimy wówczas, że β została wygenerowana bezpośrednio z π . Przez $\mathcal{H}(\pi)$ oznaczmy poddrzewo w \mathcal{H} , którego korzeniem jest wierzchołek π .

Dowolny wierzchołek π z h -tego poziomu ($h = 0, 1, 2, \dots, n$) w drzewie rozwiązań \mathcal{H} jest scharakteryzowany przez *ciągi zadań wolnych*

$$\pi^{\mathcal{B}} = (\pi(1), \pi(2), \dots, \pi(n-h)), \quad (1)$$

oraz *zadań ustalonych*

$$\pi^{\mathcal{E}} = (\pi(n-h+1), \pi(n-h+2), \dots, \pi(n)). \quad (2)$$

Tak więc w permutacji π można wyróżnić dwie subpermutacje:

$$\pi = \underbrace{(\pi(1), \pi(2), \dots, \pi(n-h))}_{\pi^B}, \underbrace{(\pi(n-h+1), \pi(n-h+2), \dots, \pi(n))}_{\pi^E}.$$

Przez

$$\mathcal{Y}(\pi) = \{\pi^B(1), \pi^B(2), \dots, \pi^B(n-h)\} \quad (3)$$

oznaczamy *zbiór zadań wolnych* w permutacji (wierzchołku drzewa) π .

Niech h będzie pewnym poziomem drzewa \mathcal{H} . Dla uproszczenia zapisu, w dalszej części tego rozdziału, przyjmujemy $t = n - h$ (jest to ostatnia wolna pozycja w π). Generowanie z π nowej permutacji β (wierzchołka na $(h+1)$ -szym poziomie drzewa) polega na ustaleniu, na pozycji t , w β jednego zadania wolnego ze zbioru $\mathcal{Y}(\pi)$. Wstawia się je na pozycję t wykonując odpowiedni ruch typu *wstaw*. Jeżeli β generujemy przez ustalenie zadania wolnego $\pi(k)$ na pozycji t , to do drzewa \mathcal{H} dodajemy wówczas łuk (π, β) . W każdym następniku β (permutacji z $\mathcal{H}(\beta)$) zadanie $\pi(k)$ jest ustalone na pozycji t . Łatwo udowodnić, że każda permutacja ze zbioru Φ jest pewnym wierzchołkiem pełnego drzewa \mathcal{H} oraz że każdy wierzchołek drzewa należy do zbioru Φ . Rozwiązanie problemu WNTJ sprowadza się więc do wyznaczenia wierzchołka w drzewie \mathcal{H} (permutacji) o minimalnej wartości funkcji kosztu. W każdym wierzchołku π drzewa \mathcal{H} zadania wolne ze zbioru $\mathcal{Y}(\pi)$ są *kandydatami* do ustalenia na ostatniej pozycji wolnej. Wobec tego z π można wygenerować bezpośrednio $|\mathcal{Y}(\pi)|$ wierzchołków.

Kolejność wybierania kandydatów do ustalenia ma istotny wpływ na ewentualne odrzucenie gałęzi, w algorytmie opartym na schemacie B&B. Z reguły dąży się do wyboru takich kandydatów aby algorytm wyznaczył jak najszybciej jak najlepsze rozwiązanie. W praktyce najczęściej wybierany jest węzeł, dla którego dolne oszacowanie wartości funkcji celu jest najmniejsze.

Zasadniczymi elementami przedstawionej metody rozwiązania problemu WNTJ są procedury wyznaczania wartości dolnego i górnego ograniczenia funkcji celu. Mają one bowiem bezpośredni wpływ na liczbę generowanych wierzchołków drzewa rozwiązań, a więc i czas obliczeń. Do ich wyznaczania wykorzystujemy sampler komputera kwantowego firmy D-Wave realizujący sprzętowo kwantowe wyżarzanie. Wymaga to sformułowania rozpatrywanego problemu w postaci kwadratowego zadania optymalizacji dyskretnej z ograniczeniami (CQM, *Constrained Quadratic Model*).

2.1. Dolne ograniczenie na komputerze D-Wave

Zadania formułowane do rozwiązania przez komputer kwantowy D-Wave muszą mieć postać programowania kwadratowego z ograniczeniami (CQM). W przypadku rozpatrywanego problemu jest to zadanie programowania liniowego całkowitoliczbowego z ograniczeniami – możliwe jest wtedy zastosowanie do jego rozwiązywania solvera *LeapHybridCQMSampler* realizującego sprzętowo kwantowe wyżarzanie. Dla ułatwienia notacji założymy, że w wierzchołku $\mathcal{H}(\pi)$ zadania wolne $\pi(1), \pi(2), \dots, \pi(t)$ są ponumerowane kolejnymi liczbami od 1 do t w taki sposób, że $d_i \leq d_{i+1}$, $i = 1, 2, \dots, t-1$. Wówczas zagadnienie $1|| \sum w_i U_i$ można sformułować równoważnie (jak w pracy Lawler'a i Moore'a [6]) jako problem maksymalizacji ważonej liczby zadań wykonywanych na czas spośród t zadań wolnych ze zbioru $\mathcal{Y}(\pi)$ w wierzchołku π drzewa \mathcal{H} :

$$\max \sum_{i=1}^t w_i x_i \quad (4)$$

przy ograniczeniach:

$$\begin{aligned}
 p_1x_1 &\leq d_1, \\
 p_1x_1 + p_2x_2 &\leq d_2, \\
 p_1x_1 + p_2x_2 + p_3x_3 &\leq d_3, \\
 &\vdots \\
 p_1x_1 + p_2x_2 + \dots + p_t x_t &\leq d_t, \\
 x_i &\in \{0, 1\}, \quad i = 1, 2, \dots, t.
 \end{aligned} \tag{5}$$

Ze względu na wymagania komputera kwantowego (minimalizacja funkcji), zadanie (4 – 5) będzie rozpatrywana jako zadanie minimalizacji z kryterium:

$$- \min \left(- \sum_{i=1}^t w_i x_i \right). \tag{6}$$

Tak sformułowany problem optymalizacyjny ma rozwiązanie tylko wtedy, gdy wszystkie linie krytyczne są nieujemne, tj. $d_i \geq 0$, $i = 1, 2, \dots, t$ (bowiem dla ujemnego d_i niespełnione jest jedno z ograniczeń (5), ponieważ p_i są dodatnie a x_i nieujemne). Jednak dla kryterium $\sum w_i U_i$, bez straty ogólności, zadania z ujemnymi liniami krytycznymi można przenieść na koniec harmonogramu. Tak więc, w dalszej części pracy, będziemy rozpatrywani problem z nieujemnymi liniami krytycznymi.

Do obliczenia wartości dolnego ograniczenia funkcji $\sum w_i U_i$ w węźle drzewa \mathcal{H} , wyznaczamy górne ograniczenie wartości funkcji (4), oznaczane przez $UB_{on\ time}$. W tym celu, dla problemu (4 – 5), stosujemy metodę relaksacji Lagrange’a. Funkcja Lagrange’a z mnożnikami $u = (u_1, u_2, \dots, u_t)$ przyjmuje dla $x = (x_1, x_2, \dots, x_t)$ postać:

$$\begin{aligned}
 L(x, u) &= \sum_{i=1}^t w_i x_i + u_1 \underbrace{(p_1 x_1 - d_1)}_{\leq 0 \text{ dla dop.}} + u_2 \underbrace{(p_1 x_1 + p_2 x_2 - d_2)}_{\leq 0 \text{ dla } x \text{ dopuszczalnych}} + \\
 &\quad \dots + u_t \underbrace{(p_1 x_1 + p_2 x_2 + \dots + p_t x_t - d_t)}_{\leq 0 \text{ dla } x \text{ dopuszczalnych}} = \\
 &= x_1 (w_1 + u_1 p_1 + u_2 p_1 + \dots + u_t p_1) + x_2 (w_2 + u_2 p_2 + u_3 p_2 + \dots + u_t p_2) + \\
 &\quad \dots + x_t (w_t + u_t p_t) - \sum_{i=1}^t u_i d_i = \sum_{i=1}^t x_i \left(w_i + p_i \sum_{j=i}^t u_j \right) - \sum_{i=1}^t u_i d_i.
 \end{aligned}$$

Niech

$$L_i(x_i, u) = x_i \left(w_i + p_i \sum_{j=i}^t u_j \right),$$

wówczas

$$L(x, u) = \sum_{i=1}^t L_i(x_i, u) - \underbrace{\sum_{i=1}^t u_i d_i}_{\text{niezależne od } x}$$

przy czym, maksymalizację $L(x, u)$ względem poszczególnych zmiennych x_i , dla ustalonych wartości u_i , można wykonywać niezależnie.

Zauważmy, że dla dowolnych $u_i \leq 0$, $i = 1, 2, \dots, t$ oraz optymalnego x^* będącego rozwiązaniem problemu z funkcją celu (4) zachodzi nierówność

$$\begin{aligned} \sum_{i=1}^t w_i x_i^* &= \max_x \sum_{i=1}^t w_i x_i \leq \min_u \max_x L(x, u) = \\ &= \min_u \left(\sum_{i=1}^t \max_x L_i(x_i, u) \right) - \sum_{i=1}^t u_i d_i = UB_{on\ time}. \end{aligned} \quad (7)$$

Z kolei, aby obliczyć górne ograniczenie funkcji Lagrange'a $UB_{on\ time}$ na komputerze D-Wave, wyznaczamy wartości wektora u korzystając z kwantowego wyżarzania, rozwiązując następujące zadanie CQM:

$$UB_{on\ time} = \min_u \left(\sum_{i=1}^t L_i(x_i, u) \right) - \sum_{i=1}^t u_i d_i, \quad (8)$$

przy ograniczeniach

$$L_i(x_i, u) \geq L_i(0, u),$$

oraz

$$L_i(x_i, u) \geq L_i(1, u),$$

dla każdego $i = 1, 2, \dots, t$ (ograniczeń jest w sumie $2t$).

Obliczenie wartości $UB_{on\ time}$ zgodnie z nierównością (7) wymaga wyznaczenia optymalnych wartości zmiennych x_i , $i = 1, 2, \dots, t$. Dla każdego $L_i(x_i, u)$ optymalizację względem x_i można wykonać niezależnie, sprawdzając dwie wartości zmiennych x_i , tj. 0 lub 1. Optymalizacja względem u może być przybliżona. Ostatecznie, wartość dolnego ograniczenia funkcji celu w wierzchołku drzewa \mathcal{H} wynosi $LB = \sum_{i=1}^t w_i - UB_{on\ time}$.

Niech

$$QuantumLagrangeLB(\pi) = F(\pi^{\mathcal{E}}) + QALagrange(\pi^{\mathcal{B}}). \quad (9)$$

Wartość

$$F(\pi^{\mathcal{E}}) = \sum_{j=t+1}^n w_{\pi(j)} U_{\pi(j)}(C_{\pi(j)}) \quad (10)$$

jest sumą kosztów zadań ustalonych, gdzie $U_{\pi(j)}(C_{\pi(j)}) = 1$ jeśli $C_{\pi(j)} > d_{\pi(j)}$ oraz $U_{\pi(j)}(C_{\pi(j)}) = 0$ w przeciwnym przypadku, a $C_{\pi(j)} = \sum_{i=1}^j p_{\pi(i)}$. Z kolei $QALagrange(\pi^{\mathcal{B}})$ jest równe $\sum_{i=1}^t w_i - L(x^*, u)$ dla optymalnych wartości zmiennych binarnych x^* oraz u otrzymanego w wyniku zastosowania kwantowego wyżarzania.

2.2. Górne ograniczenie na komputerze D-Wave

Obecnie przedstawimy metodę obliczania górnego ograniczenia optymalnej wartości funkcji celu w poddrzewie $\mathcal{H}(\pi)$. Niech

$$QuantumAnnealingUB(\pi) = F(\pi^{\mathcal{E}}) + QA(\pi^{\mathcal{B}}), \quad (11)$$

gdzie $QA(\pi^{\mathcal{B}})$ jest pewnym rozwiązaniem przybliżonym NP-trudnego problemu WNTJ dla zbioru zadań wolnych $\mathcal{Y}(\pi) = \{\pi^{\mathcal{E}}(1), \pi^{\mathcal{E}}(2), \dots, \pi^{\mathcal{E}}(t)\}$ w wierzchołku π drzewa \mathcal{H} . Wyznaczamy je stosując (sprzętowa) metaheurystykę kwantowego wyżarzania, dla modelu obliczeń opisanego w poprzednim rozdziale (czyli transformacji do problemu maksymalizacji $\sum w_i x_i$).

3. Algorytm dokładny sterowany kwantowym wyżarzaniem

W tym rozdziale przedstawiamy pseudokod hybrydowego algorytmu dokładnego rozwiązywania problemu WNTJ opartego na metodzie B&B. Kluczowe elementy – dolne i górne ograniczenia – są liczone za pomocą kwantowego wyżarzania na komputerze D-Wave. Za rozwiązanie startowe przyjęto permutację identycznościową π_0 . Pseudokod przedstawia Algorytm 1.

Algorithm 1: QAB&B

Input : π_0 – rozwiązanie startowe;
Output: π^* – rozwiązanie optymalne;

- 1 *Heap* : kolejka priorytetowa rozwiązań częściowych sortowana rosnąco względem ich górnych ograniczeń *LocalUB*;
- 2 Put(*Heap*, (π_0 , n , 0, $F(\pi_0)$));
- 3 $\pi^* \leftarrow \pi_0$
- 4 **while** *Heap* \neq **do**
- 5 Get(*Heap*, (π , t , LB_π , UB_π));
- 6 **if** $LB_\pi < F(\pi^*)$ **then**
- 7 Wyznacz zbiór K_π – zadań, które można ustalić na t -tej pozycji;
- 8 **for** $j \in K_\pi$ **do**
- 9 Swap(π , $\pi^{-1}(j)$, t);
- 10 $LocalLB \leftarrow QuantumLagrangeLB(\pi)$;
- 11 **if** $LocalLB < F(\pi^*)$ **then**
- 12 $\pi_{LocalUB} \leftarrow \arg(QuantumAnnealingUB(\pi))$;
- 13 **if** $\min\{F(\pi_{LocalUB}), F(\pi)\} < F(\pi^*)$ **then**
- 14 $\pi^* \leftarrow \arg \min\{F(\pi_{LocalUB}), F(\pi)\}$;
- 15 Put(*Heap*, (π , $t - 1$, $LocalLB$, $F(\pi_{LocalUB})$));
- 16 Swap(π , t , $\pi^{-1}(j)$);

Podczas przeglądania drzewa rozwiązań \mathcal{H} zastosowano strategię *best-first* z użyciem kolejki priorytetowej realizowanej za pomocą kopca (zmienna *Heap*). W wierzchołkach są pamiętane czwórki: (*permutacja*, *liczba zadań wolnych*, *dolne ograniczenie*, *wartość funkcji celu*). W linii 9-tej algorytmu zadania-kandydaci są wstawiani na ostatnią wolną pozycję t . Następnie, w linii 10 jest obliczane dolne ograniczenie. Jeżeli jest ono mniejsze od wartości funkcji celu najlepszego rozwiązania, to w wierszu (12), na komputerze kwantowym, jest wyznaczane najlepsze rozwiązanie w poddrzewie $\mathcal{H}(\pi)$. W wierszach 13 i 14 jest ewentualnie modyfikowane najlepsze do tej pory znalezione rozwiązanie. Po przeprowadzeniu tych obliczeń generowany jest nowy wierzchołek (korzeń poddrzewa), który w linii 15 jest dodawany do kolejki priorytetowej *Heap*. Wynikiem działania algorytmu jest permutacja optymalna π^* .

4. Podsumowanie

Wykonywanie obliczeń na komputerze kwantowym stwarza nowe możliwości rozwiązywania NP-trudnych problemów optymalizacji dyskretnej. Pomimo, że algo-

rytm kwantowego wyżarzania nie gwarantuje optymalności wyznaczonego rozwiązania, to jednak może on być z powodzeniem wykorzystany w konstrukcji algorytmu dokładnego. Znacznie poprawia bowiem jego efektywność poprzez (niedeterministyczne) sterowanie procesem przeszukiwania przestrzeni rozwiązań. W pracy przedstawiono koncepcję kwantowego hybrydowego algorytmu podziału i ograniczeń rozwiązywania jednomaszynowego problemu szeregowania zadań z minimalizacją ważonej liczby zadań spóźnionych. Do wyznaczania trajektorii obliczeń wykorzystuje się wyniki obliczeniowe algorytmu kwantowego wyżarzania. Przedstawiony algorytm może być zaadoptowany do rozwiązywania innych problemów NP-trudnych. Eksperymenty wykonywano na maszynie kwantowej firmy D-Wave.

LITERATURA

1. Bożejko, W., Pempera, J., Uchroński, M., Wodecki, M.: Distributed Quantum Annealing on D-Wave for the Single Machine Total Weighted Tardiness Scheduling Problem. In: Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds) Computational Science – ICCS 2022. LNCS 13353, Springer, (2022) p. 171–178.
2. Garey M.R., Johnson D.S.: Scheduling tasks with nonuniform deadlines on two processor. *Journal of ACM*, 23, 1976, p. 461–467.
3. Karp R.M.: Reducibility among combinatorial problems. Eds.: R.E. Miller and J.W. Thatcher, *Complexity of Computer Computation*, Plenum Press, New York, 1972, p. 85–104.
4. Kise H., Ibaraki T., Mine H.: A solvable case of the one-machine scheduling problem with ready times and due times. *Operations Research*, 26, 1978, p. 121–126.
5. Lawler E.L.: A „pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1, 1977, p. 331–342.
6. Lawler E. L., Moore J. M.: A Functional Equation and its Application to Resource Allocation and Sequencing Problems. *Management Science* 16(1), 1969, p. 77–84.
7. Lenstra J.K., Rinnoy Kan A.H.G.: Complexity results for scheduling chains on a single machine. *European Journal of Operational Research* 4, 1980, p. 270–275.
8. Monma C.I.: Linear-time algorithms for scheduling on parallel processor. *Operations Research*, 30, 1982, 116–124.
9. Moore J.M.: An n-job, one machine sequencing algorithm for minimising the number of late jobs. *Management Science*, 15, 1968, p. 102–109.
10. Potts C.N., Van Wassenhove L.N.: A Branch and Bound Algorithm for the Total Weighted Tardiness Problem. *Operations Research*, 33, 1985, p. 177–181.
11. Sahni S.K.: Algorithms for Scheduling Independent Jobs. *Journal Association for Computing Machinery*, 23, 1976, p. 116–127.
12. Villareal F.J., Bulfin R.L.: Scheduling a Single Machine to Minimize the Weighted Number of Tardy Jobs. *IEEE Transaction*, 15, 1983, p. 337–343.