Problem optymalnego transportu dźwigarów w systemie Just in Time

Izabela Biczysko, Krzysztof Piecuch

Instytut Informatyki Uniwersytetu Wrocławskiego

Wrocław, 5 grudnia 2012

Agenda

Motywacja

- Firma zajmująca się przebudową sieci transportowej chce zoptymalizować proces transportu dźwigarów na plac budowy
- Proces montażu dźwigarów jest ściśle powiązany z innymi pracami budowlanymi
- Przekroczenie terminu zamontowania dźwigaru wiąże się z dodatkowymi kosztami
- Nie ma możliwości składowanie dźwigarów na placu budowy
- Bezpośrednio z dźwigarów pojazdy są umieszczane na podporach
- Transport dźwigarów odbywa się w systemie Just in Time (JIT)

Zadania

Znane są okna czasowe tzn. najwcześniejsze i najpóźniejsze terminy dostawy poszczególnych dźwigarów. Należy znaleźć takie terminy dostawy poszczególnych dźwigarów na plac budowy aby zminimalizować koszty związane z przekroczeniem okien czasowych.

Ograniczenia

1. dźwigary należy dostarczyć zgodnie z kolejnością montażu (porządkiem technologicznym)
2. jednocześnie pojazd może przewozić tylko jeden dźwigar
3. po załadunku dźwigar może być zdjęty jedynie bezpośrednio przed montażem

Model matematyczny

Zbiór dźwigarów

$\mathbf{B} = B_1, B_2, B_3, ..., B_m$

Dla dowolnego dźwigara $B_i \in \mathbf{B}$ wprowadzamy oznaczenia

- z_i - czas załadunku
- t_i - czas transportu na plac budowy
- r_i - czas rozładunku na placu budowy
- p_i - czas powrotu
- e_i - żądany najwcześniejszy termin przywozu
- d_i - żądany najpóźniejszy termin przywozu
- v_i - współczynnik funkcji kary za zbyt wczesne przybycie
- w_i - współczynnik funkcji kary za zbyt późne przybycie

Model matematyczny

Założenia

Niech:

- kolejność transportu dźwigarów to $(B_1, B_2, ..., B_m)$
- załadunek pierwszego dźwigara rozpocznie się w chwili 0
- S_i oznacza termin dostarczenia dźwigara B_i na plac budowy

Model matematyczny

Terminy $S_1, S_2, ..., S_n$ muszą spełniać następujące ograniczenia

- załadunek pierwszego dźwigara B_i rozpoczyna się w chwili 0

$$S_1 \geqslant z_1 + t_1 \tag{1}$$

- dźwigary należy dostarczyć zgodnie z kolejnością montażu

$$S_{i+1} \geqslant S_i, i = 1, 2, ..., m - 1 \tag{2}$$

Model matematyczny

Terminy $S_1, S_2, ..., S_n$ muszą spełniać następujące ograniczenia

- jednocześnie pojazd może przewozić tylko jeden dźwigar

$$\begin{aligned} \forall B_i, B_j \in \mathbf{B}, S_i - r_i - t_i - z_i \geqslant S_j - r_j - t_j - z_j \\ \vee S_j - r_j - t_j - z_j \geqslant S_i - r_i - t_i - z_i \end{aligned} \tag{3}$$

- po załadunku dźwigar może być zdjęty jedynie bezpośrednio przed montażem

$$S_{i+1} \geqslant S_i + r_i + p_i + z_{i+1} + t_{i+1}, i = 1, 2, ..., m-1 \tag{4}$$

Zadanie

Problem transportu dźwigarów sprowadza się więc do wyznaczenia terminów dostaw $S_1, S_2, ..., S_n$ spełniających ograniczenia (1) - (4), które optymalizują pewne przyjęte kryterium optymalizacyjne

Transport dokładnie na czas jednym pojazdem

- dźwigary są transportowane na plac przez dokładnie jeden pojazd

- $E_i = max\{0, e_i - S_i\}$
- $T_i = max\{0, S_i - d_i\}$

Kryterium optymalizacyjne

$$F(S) = \sum_{i=1}^{n}(u_i E_i + w_i T_i) \tag{5}$$

Pojęcia i twierdzenia

Theorem

Istnieje optymalne rozwiązanie w którym dla każdego dźwigara i czynności z_i, t_i, r_i, p_i *są wykonywane pod rząd bez okresów bezczynności*

Uwaga

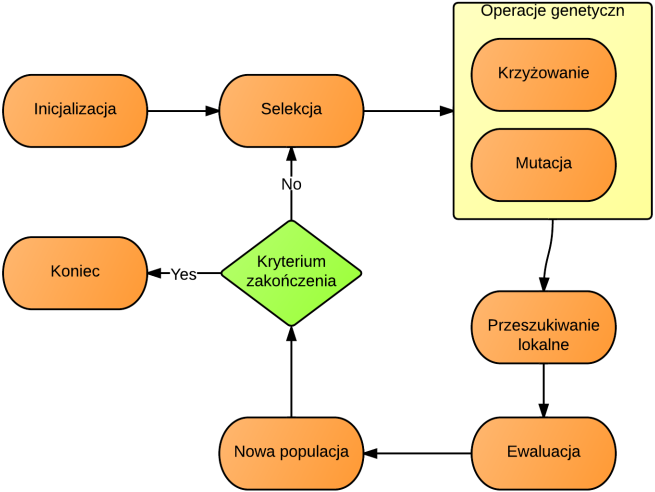
Wobec twierdzenia 1, niech $j_i = z_i, t_i, r_i, p_i$ będzie praca związaną z dostarczeniem i-tego dźwigara.

Pojęcia i twierdzenia

Definition

Podciąg $u, u+1, \ldots v$ ciągu $\mathbf{S} = S_1, S_2, \ldots, S_n$ nazwiemy **blokiem**, jeżeli zadania j_u, j_{u+1}, j_v są wykonywane pad rząd bez czasu bezczynności, natomiast jest czas bezczynności przed zadaniem u i po zadaniu v.

Schemat algorytmu



Inicjalizacja

Opcja 1: losowe permutacje

Opcja 2: dźwigary posortowane według czasu początkowego, czasu końcowego, średniej z tych czasów plus losowe osobniki

Opcja 3: dźwigary posortowane według czasu początkowego, czasu końcowego, średniej z tych czasów i mutacje tych osobników

Selekcja

- osobniki są losowo łączone w pary i dla każdej pary przeprowadzana jest mutacja

Krzyżowanie - Uniform order based crossover

Rodzic 1:	1	2	3	4	5	6	7	8
Rodzic 2:	6	8	1	2	4	3	7	5
Wektor Binarny	1	0	1	1	1	0	0	1

Krzyżowanie - Uniform order based crossover

Wybieramy numery genów zgodnie z wektorem binarnym

Rodzic 1:	1	2	3	4	5	6	7	8
Wektor Binarny	1	0	1	1	1	0	0	1
Dziecko 1:	_	2	_	_	_	6	7	_
Rodzic 2:	6	8	1	2	4	3	7	5
Wektor Binarny	1	0	1	1	1	0	0	1
Dziecko 2:	6	_	1	2	4	_	_	5

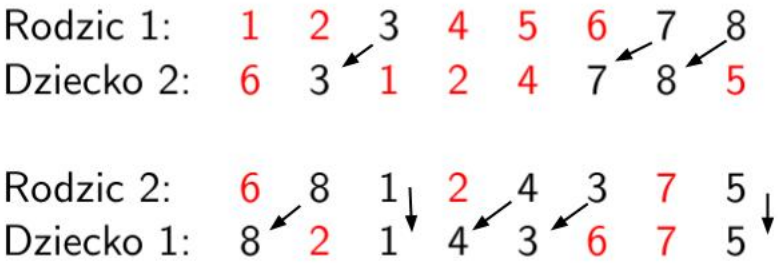
Krzyżowanie - Uniform order based crossover

Z zodzica 1 usuwamy wszystkie geny które są w dziecku 2 a z rodzica 2 wszystkie geny które są w dziecku 1

Rodzic 1:	1	2	3	4	5	6	7	8
Dziecko 2:	6	_	1	2	4	_	_	5
Rodzic 2:	6	8	1	2	4	3	7	5
Dziecko 1:	_	2	_	_	_	6	7	_

Krzyżowanie - Uniform order based crossover

Nieusunięte geny z rodziców przepisujemy do dzieci nie zmieniając ich kolejności



Mutacja

Dla każdego osobnika z niewielkim prawdopodobieństwem θ_1 zamieniamy dwa losowe geny.

Poszukiwanie lokalne

Dla najlepszego osobnika uruchamiamy heurystykę bazującą na algorytmie NEH

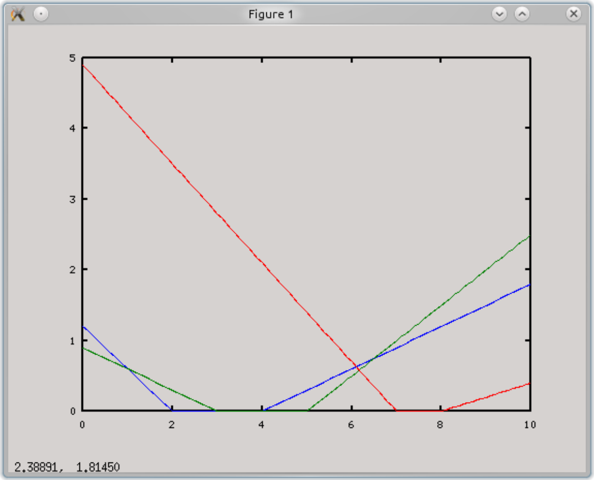
Heurystyka NEH

1. Z danego osobnika weź pierwsze dwa geny i znajdź dla nich kolejność minimalizującą funkcję celu
2. Dla osobnika o długości od 3 do N wykonuj następującą operację:
    - Wstaw k-ty gen w taki sposób pomiędzy k-1 genów aby zminimalizować funkcję celu dla osobnika o k genach

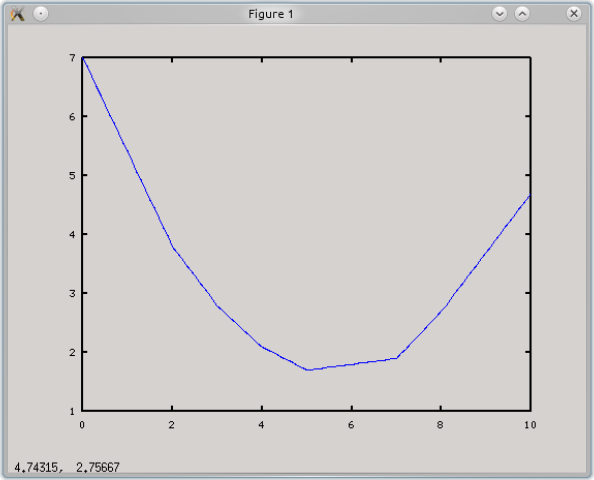
Ewaluacja

- Algorytm przyrostowy
- Próbujemy umieścić i-ty dźwigar w okienku
- Jeśli to nie możliwe, kładziemy go zaraz po dźwigarze $i - 1$.
- Przesuwamy cały blok w lewo, aż nie znajdziemy minimum

Ewaluacja



Ewaluacja



Tworzenie nowej populacji

Z sumy populacji N rodziców i N dzieci stwórz populację składającą się z N najlepszych osobników

Warunek zakończenia algorytmu

Jeżeli przez D populacji nie nastąpi polepszenie funkcji celu zakończ działanie algorytmu