Problem dystrybucji towarów z najpóźniejszymi terminami dostaw

Piotr Rzepecki, Krzystof Zielonka

16 grudnia 2012

Opis problemu (orginalny)

Mamy dany ważony graf pełny miast z magazynem (punktem startowym). Wagi na krawędziach interpretujemy jako czasy potrzebne na przemieszczenie sie między miastami. Dodatkowo każdy wierzchołek ma przypisany najpóźniejszy czas dostawy za przekroczenie którego otrzymujemy karę. Ciężarówka ma nieskończoną pojemność wiec raz załadowana może jeździć dowolnie długo. Znamy również maksymalny czas podróży ciężarówki po jakim żadne zadanie nie jest dopuszczalne. Należny znaleść ciag miast, taki że każde miasto jest odwiedzone i suma kar jest minimalna.

Instancja problemu

Miastom przyporządkowujmey kolejno numery $2, \cdots, n$. Dla uproszczenia będziemy zakładać, że magazyn zawsze ma numer 1. Instancją problemu jest trójka:

$$P = \langle T, K, D \rangle \tag{1}$$

Macierz czasu przejazdów

$$T = [t_{ij}]_{n \times n} \tag{2}$$

Kwadratowa maciarz gdzie element t_{ij} to czas potrzebny na przejazd z miasta i do j.

K i D

Wektory K i D przyporządkowują wierzchołkom odpowiednio najpóźniejsze czasy dostawy oraz karę za ich przekroczenie.

Rozwiązania dopuszczalne

Rozwiązaniem dopuszczalnym (spełniającym warunki zadania) jest permutacja liczb $1, \cdots, n$.

Uzasadnienie

- W rozwiąznu musza znaleść sie wszystkie miasta. (definicja problemu)
- W rozwiązaniu miasta nie mogą sie powtarzać. (nierówność trójkąta + założenie o nieskończonej ładowności ciężarówki)
- W rozwiązaniu magazyn musi być pierwszym wierzchołkiem.

Funkcja celu

Jeśli przez t_i oznaczymy czas dotarcia do wierzchołka v_i to rozwiązanie możemy ocenić za pomocą funkcji celu której interpretacja to suma kar jakie ponieśliśmy:

$$f = \sum_{i=2}^{n} m_i \tag{3}$$

$$m_i = \text{if } t_i > d_i \text{ then } k_i \text{ else } 0 \tag{4}$$

Warianty problemu:

- ograniczenia na **limit pojemności** pojazdu wymuszający odwiedzanie magazynu,
- **okna dostawy** (najpóźniejszy jak i najwcześniejszy czas dostawy),
- minimalizacja **liczby błędów** zamiast minimalizacji funkcji kary.

Ocena

W drugim przypadku oceną rozwiązania jest krotka (liczba błędów, długość przejazdu) i wyniki sortujemy leksykograficznie, tj. w pierwszej kolejności minimalizujemy liczbę przekroczonych terminów, a dopiero potem długość trasy.

Dane testowe i benchmarki:

- VRPTW Benchmark Problems (M. Solomon)
  `http://web.cba.neu.edu/~msolomon/problems.htm`
  (dane do wariantu z limitem ładowności)
- TSPTW Benchmark Instances (różni autorzy)
  `http://iridia.ulb.ac.be/~manuel/tsptw-instances`
  (bez limitu ładowności)

Różnice i optymalne wyniki

Dane testowe z prefixem R to rozmieszczenie losowe, C to rozmieszczenie pogrupowane (ang. clustered), RC to dane mieszane.
Do większości przykładów znane są tylko najlepsze znalezione wyniki (niekoniecznie optymalne).

Ocena wyników:

- porównanie z najlepszym opublikowanym rozwiązaniem benchmarka,
- porównanie z algorytmem losowym przetwarzający taką samą liczbę rozwiązań.

Schemat algorytmu SGA

```
def SGA(F, N, M, θ_C, θ_M):
    P = RANDOM-POPULATION(N)
    POPULATION-EVALUATION(P, F)
    while not TERMINATION-CONDITION(P):
        PS = PARENT-SELECTION(P)
        PC = CROSSOVER(PS, θ_C)
        P = MUTATION(PC, θ_M)
        POPULATION-EVALUATION(P, F)
```

Operator krzyżowania

Operator PMX

- Każdy z rodziców jest dzielony na trzy segmenty,
- środkowe segmenty zostają wymienione ze sobą,
- pozostałe segmenty przepisujemy naprawiając własność permutacji.

Operatory mutacji

Wykorzystaliśmy dwa operatory, w każdej mutacji wybierany jest jeden z nich z jednakowym prawdopodobieństwem.

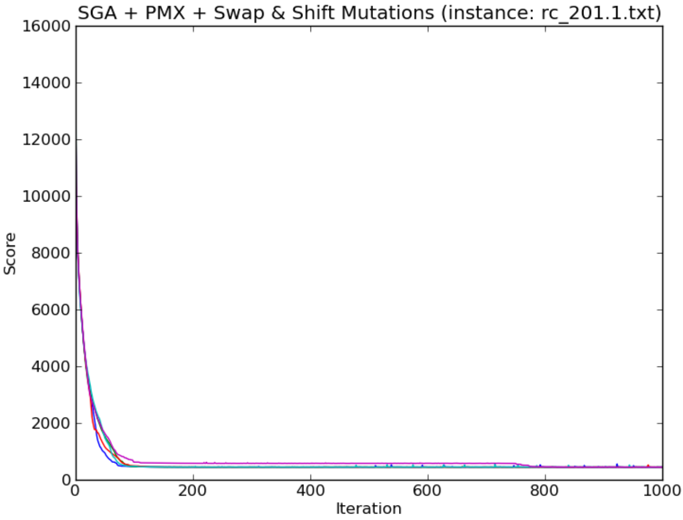
- swap mutation
  1 2 **3** 4 5 6 **7** 8 9 10
  1 2 **7** 4 5 6 **3** 8 9 10
- shift mutation
  1 2 **3 4 5 6 7** 8 9 10
  1 2 **4 5 6 7 3** 8 9 10

Replacement

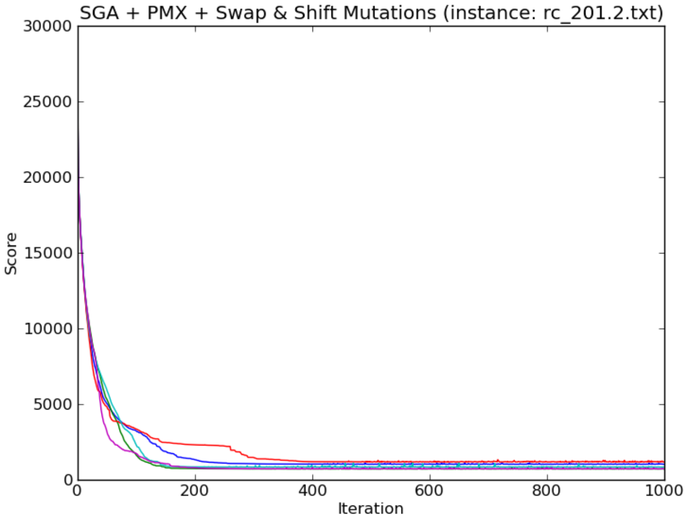
Wybraliśmy wariant $(\mu + \lambda)$, tj. w każdej kolejnej iteracji populacja wybierana jest z najlepszych osobników zarówno z rodziców jak i z dzieci.

Tablica: Wyniki dla zestawu danych SolomonPotvinBengio

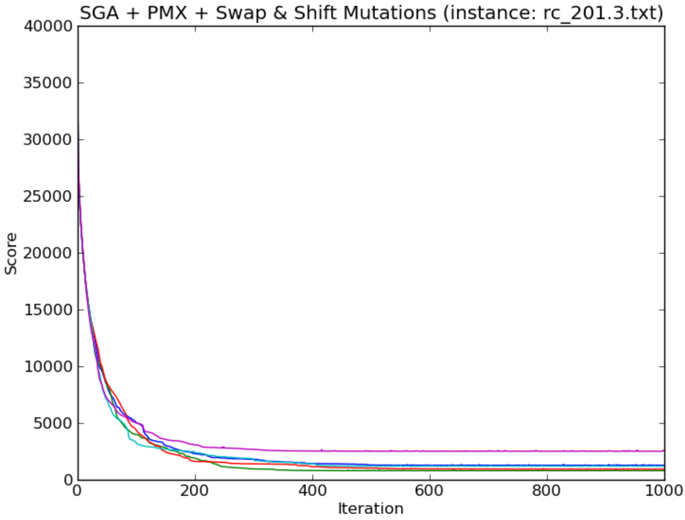
zestaw	najlepszy opublikowany	najlepszy znaleziony
$rc\_205.1(n = 14)$	$(0, 343.21)$	$(0, 343.21)$
$rc\_203.4(n = 15)$	$(0, 314.29)$	$(0, 314.29)$
$rc\_203.1(n = 19)$	$(0, 453.48)$	$(0, 479.83)$
$rc\_201.1(n = 20)$	$(0, 444.54)$	$(0, 444.54)$
$rc\_201.2(n = 26)$	$(0, 711.54)$	$(1, 737.38)$
$rc\_201.3(n = 32)$	$(0, 790.61)$	$(2, 809.72)$
$rc\_204.1(n = 46)$	$(0, 878.64)$	$(4, 990.07)$



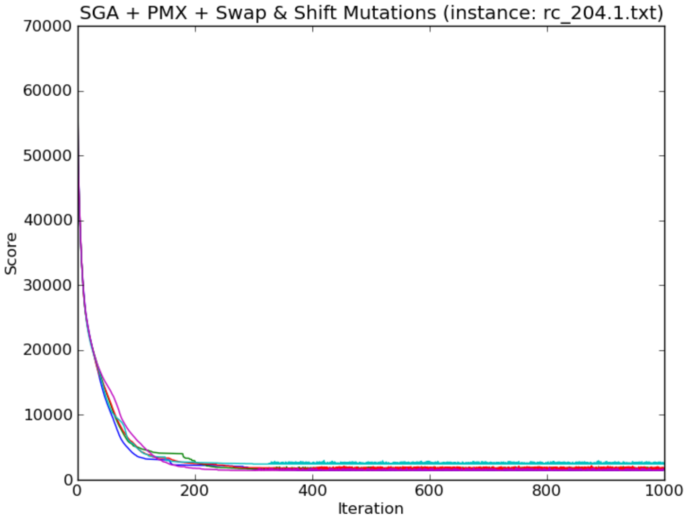
Rysunek: rc_201.1 (n=20)



Rysunek: rc_201.2 (n=26)



Rysunek: rc_201.2 (n=32)



Rysunek: rc_201.2 (n=46)