

# Algorytmy metaheurystyczne

Problem komiwojażera euklidesowego. Local Search.

Karol Janic

16 listopada 2023

# Spis treści

<b>1</b>	<b>Cel zadania</b>	<b>2</b>
<b>2</b>	<b>Algorytm Local Search</b>	<b>2</b>
2.1	Otoczenie invert . . . . .	2
2.1.1	Moc otoczenia . . . . .	2
2.1.2	Własność ruchu invert . . . . .	2
2.1.3	Problem nierówności trójkąta przy zaokrąglaniu . . . . .	3
2.2	Wybór rozwiązań początkowego . . . . .	3
2.3	Przegląd sąsiedztwa . . . . .	3
2.4	Generowanie najlepszego kandydata . . . . .	3
<b>3</b>	<b>Wyniki</b>	<b>3</b>
<b>4</b>	<b>Wnioski</b>	<b>4</b>
<b>5</b>	<b>Wizualizacje rozwiązań</b>	<b>5</b>
5.1	xqf131 . . . . .	5
5.2	xqg237 . . . . .	7
5.3	pma343 . . . . .	9
5.4	pka379 . . . . .	11
5.5	bcl380 . . . . .	13
5.6	pbl395 . . . . .	15
5.7	pbk411 . . . . .	17
5.8	pbn423 . . . . .	19
5.9	pbm436 . . . . .	21
5.10	xql662 . . . . .	23
5.11	xit1083 . . . . .	25
5.12	icw1483 . . . . .	27
5.13	djc1785 . . . . .	29
5.14	dcb2086 . . . . .	31
5.15	pds2566 . . . . .	33

# 1 Cel zadania

Celem zadania jest sprawdzenie skuteczności heurystyki Local Search na przykładzie euklidesowego problemu komiwojażera oraz zbadanie wpływu wyboru rozwiązania początkowego i metody generowania otoczenia na jakość rozwiązania.

## 2 Algorytm Local Search

### 2.1 Otoczenie invert

Otoczeniem rozwiązania reprezentowanego przez permutację  $\pi$  jest zbiór rozwiązań uzyskanych przy użyciu pojedynczego ruchu  $invert(\pi, i, j)$ , który zamienia kolejność wierzchołków od  $i$ -tego do  $j$ -tego.

#### 2.1.1 Moc otoczenia

Można łatwo zauważać, że:

- $invert(\pi, i, i) = \pi$
- $invert(\pi, i, j) = invert(\pi, j, i)$

Zatem ruchy powodujące powstawanie różnych rozwiązań z rozwiązania  $\pi$  można opisać zbiorem (wierzchołki numerujemy liczbami od 1 do  $n$ ):

$$INV = \{invert(\pi, i, j) : 1 \leq i < j \leq n\}$$

Aby obliczyć moc zbioru  $INV$  ustalamy  $i$  kolejno na  $1, 2, \dots, (n - 1)$  oraz dobieramy odpowiednie  $j$ , czyli  $i < j \leq n$ . Wtedy:

$$|INV| = (n - 1) + (n - 2) + \dots + 1 = \frac{(n - 1)n}{2} = \frac{n^2 - n}{2}$$

Interpretując permutację  $\pi$  jako ciąg wierzchołków do odwiedzenia można zauważać, że  $invert(\pi, 1, n)$  nie wyprowadza nowego rozwiązania, ponieważ nie ma znaczenia czy odwiedzanie rozpoczniemy od pierwszego czy ostatniego wierzchołka, więc całe otoczenie ma rozmiar  $\frac{n^2 - n}{2} - 1$ .

#### 2.1.2 Własność ruchu invert

Cechą ruchu typu invert jest "rozplątywanie" pętli w cyklu. Pętle takie są nieoptymalne. Można to pokazać korzystając z nierówności trójkata, która na przestrzeni euklidesowej zachodzi:

$$d(A, D) \leq d(A, P) + d(D, P) \quad d(B, C) \leq d(B, P) + d(P, C)$$

$$\begin{aligned} d(A, C) + d(C, D) + d(D, B) &= \\ &= d(A, P) + d(P, C) + d(C, D) + d(D, P) + d(P, B) \geq \\ &\geq d(A, D) + d(D, C) + d(C, B) \end{aligned}$$



### 2.1.3 Problem nierówności trójkąta przy zaokrąglaniu

W realizacji modelu odległości pomiędzy wierzchołkami wyrażane są liczbami całkowitymi. Konwersja dokładnej odległości następuje poprzez zaokrąglenie jej do najbliższej liczby całkowitej. W takim modelu nierówność trójkąta nie zawsze zachodzi. Weźmy dla przykładu powyżej punkty  $A(0, 0.5)$ ,  $B(0.5, 0.5)$ ,  $C(0.5, 0)$ ,  $D(0, 0)$ . Wówczas punktem przecięcia jest  $P(0.25, 0.25)$ . Rzeczywiste odległości prezentują się następująco:

$$d(A, D) = d(B, C) = 0.5$$

$$d(A, P) = d(D, P) = d(B, P) = d(C, P) \approx 0.35$$

$$d(A, C) = d(B, D) \approx 0.7$$

Odległości w modelu prezentują się natomiast tak jak zapisano poniżej:

$$d'(A, D) = d'(B, C) = 1$$

$$d'(A, P) = d'(D, P) = d'(B, P) = d'(C, P) = 0$$

$$d'(A, C) = d'(B, D) = 1$$

Wtedy nierówności w  $\triangle APC$  oraz  $\triangle BPC$  nie zachodzą a obie skonstruowane drogi mają długość 3.

## 2.2 Wybór rozwiązania początkowego

1. rozwiązanie zbudowane na podstawie MST
2. rozwiązanie wygenerowane w sposób losowy

## 2.3 Przegląd sąsiedztwa

1. przeglądanie całego sąsiedztwa (rozmiar:  $O(n^2)$ )
2. przeglądanie  $n$  losowo wybranych sąsiadów (rozmiar:  $O(n)$ )

## 2.4 Generowanie najlepszego kandydata

Gdy rozwiązaniem początkowym jest rozwiązanie generowane na podstawie MST to  $\sqrt{n}$  razy generujemy takie rozwiązanie zaczynając od losowego wierzchołka drzewa a następnie używamy go jako startowego w algorytmie Local Search.

Gdy rozwiązaniem początkowym jest rozwiązanie losowe to  $n$  razy powtarzamy losowanie rozwiązania oraz poprawianie go algorytmem Local Search. W przypadku  $n > 1000$  procedurę powtarzamy tylko 100 razy.

W każdym przypadku jako wynik wybieramy rozwiązanie o najmniejszej wadze.

## 3 Wyniki

Oznaczenia metod generujących kandydatów:

- LS1 - algorytm Local Search startujący z rozwiązania wygenerowane z losowego wierzchołka MST i przeglądający całe otoczenie w każdej iteracji
- LS2 - algorytm Local Search startujący z losowego rozwiązania i przeglądający całe otoczenie w każdej iteracji
- LS3 - algorytm Local Search startujący z losowego rozwiązania i przeglądający  $n$  losowych sąsiadów w każdej iteracji

Zaimplementowane metody zostały porównane na przykładach z <https://www.math.uwaterloo.ca/tsp/vlsi/index.html>. Wyniki prezentują się następująco:

Przykład	Suma wag rozwiązań optymalnego	Suma wag kandydata opartego o MST	Suma wag najlepszego rozwiązania LS1	Suma wag najlepszego rozwiązania LS2	Suma wag najlepszego rozwiązania LS3
xqf131	564	718	596	578	880
xqg237	1019	1445	1067	1062	1676
pma343	1368	1883	1436	1424	2312
pka379	1332	1855	1398	1399	2271
bcl380	1621	2319	1711	1728	3095
pbl395	1281	1871	1351	1359	2390
pbk411	1343	1935	1421	1426	2521
pbn423	1365	1918	1453	1454	2473
pbm436	1443	2119	1532	1535	2685
xql662	2513	3691	2661	2707	4926
xit1083	3558	5190	3803	3919	7256
icw1483	4416	6754	4757	4839	9293
djc1785	6115	8908	6477	6697	12617
dcb2086	6600	9777	7107	7313	14780
pds2566	7643	11427	8182	8506	17506

Przykład	Rozwiązania LS1			Rozwiązania LS2			Rozwiązania LS3		
	liczba popraw	suma wag		liczba popraw	suma wag		liczba popraw	suma wag	
		średnia	minimum		średnia	minimum		średnia	minimum
xqf131	28.8	601.9	596	133.3	612.4	578	111.9	1044.4	880
xqg237	37.0	1083.7	1067	260.5	1115.8	1062	225.2	2090.4	1676
pma343	75.8	1450.3	1436	403.5	1484.7	1424	366.3	2835.0	2312
pka379	78.9	1416.3	1398	448.7	1445.9	1399	408.9	2769.7	2271
bcl380	62.7	1749.3	1711	446.7	1817.5	1728	374.2	3827.0	3095
pbl395	80.0	1371.5	1351	458.8	1429.0	1359	387.4	2952.4	2390
pbk411	81.6	1437.9	1412	483.7	1488.5	1426	406.6	3162.3	2512
pbn423	72.3	1473.9	1453	497.0	1521.6	1454	419.8	3200.2	2473
pbm436	97.85	1562.5	1543	512.8	1612.0	1535	433.4	3363.1	2685
xql662	122.6	2690.4	2661	811.4	2813.3	2707	697.2	6035.1	4926
xit1083	191.6	3847.2	3803	1383.7	4021.2	3919	1212.0	9052.2	7256
icw1483	278.8	4809.8	4757	1929.4	4990.5	4839	1695.5	11436.9	9293
djc1785	382.6	6533.8	6477	2348.9	6872.1	6697	2065.4	15529.9	12617
dcb2086	390.9	7154.2	7107	2796.6	7457.8	7313	2457.2	17776.7	7107
pds2566	457.3	8271.4	8182	3485.9	8701.8	8506	3052.0	21375.1	17506

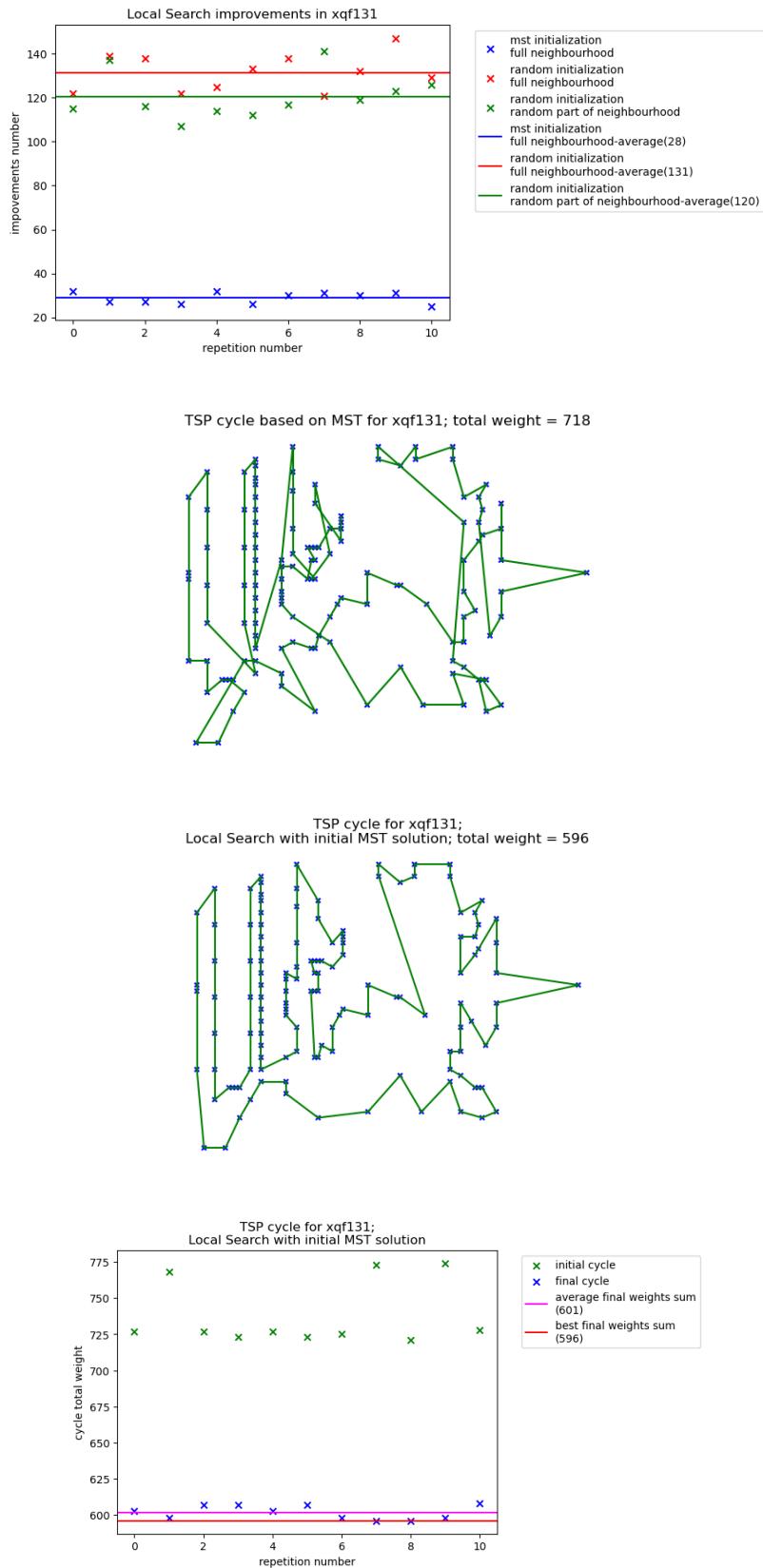
Tabela 1: Podsumowanie działania Local Search dla problemu komiwojażera.

## 4 Wnioski

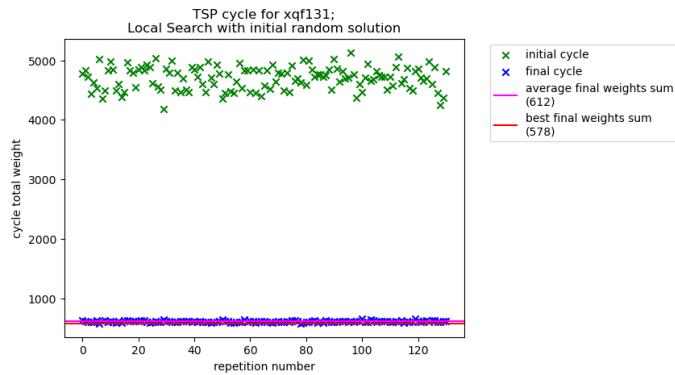
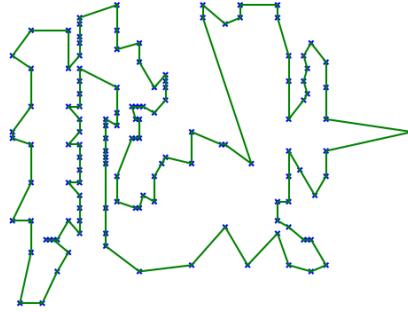
- Rozwiązanie początkowego tworzone na podstawie MST skracza czas działania algorytmu w porównaniu do losowego rozwiązania początkowego
- Rozwiązanie początkowego tworzone na podstawie MST prowadzi do uzyskania lepszej średniej wyników w porównaniu do losowego rozwiązania początkowego
- Losowe rozwiązanie początkowe dla mniejszych danych prowadzi do znalezienia lepszego minimum w porównaniu do rozwiązanie początkowego tworzone na podstawie MST
- Algorytm Local Search przeglądający całe otoczenie invert "rozplątuje" skrzyżowania. Nie jest to prawda, gdy rozpatrywana jest wyłącznie losowa część otoczenia
- Przeglądanie części otoczenia skracą czas działa algorytmu ale prowadzi do gorszych rezultatów

## 5 Wizualizacje rozwiązań

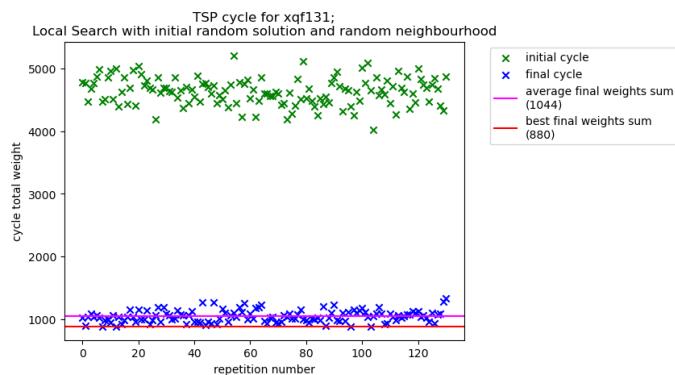
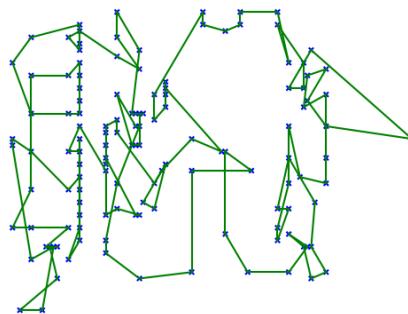
### 5.1 xqf131



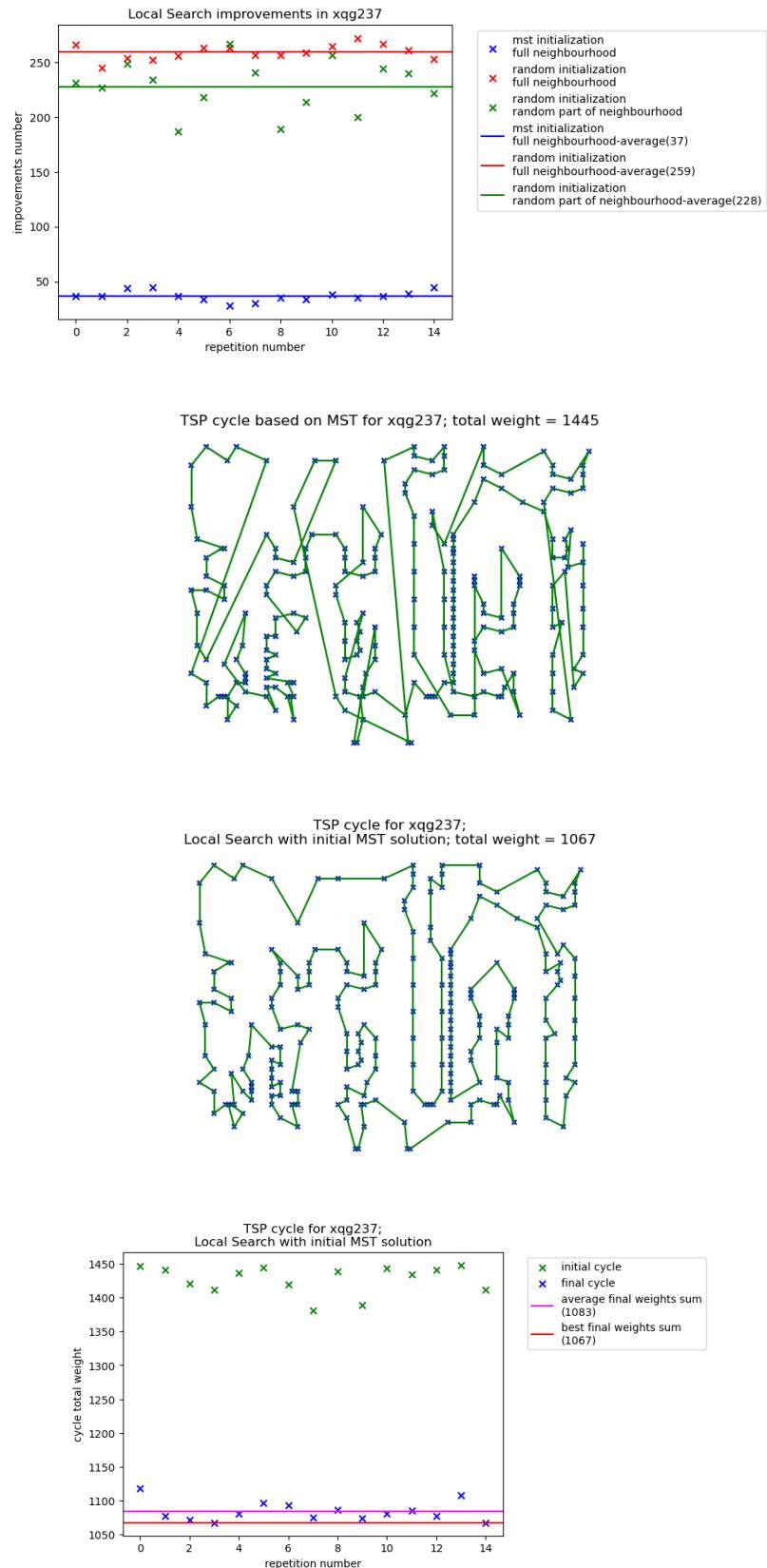
TSP cycle for xqf131;  
Local Search with initial random solution; total weight = 578



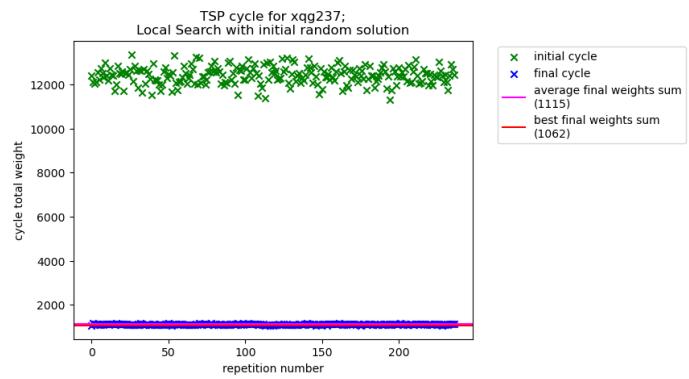
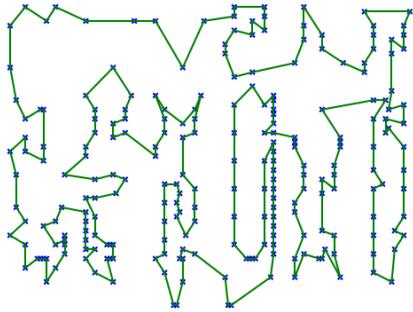
TSP cycle for xqf131;  
Local Search with initial random solution and random neighbourhood; total weight = 880



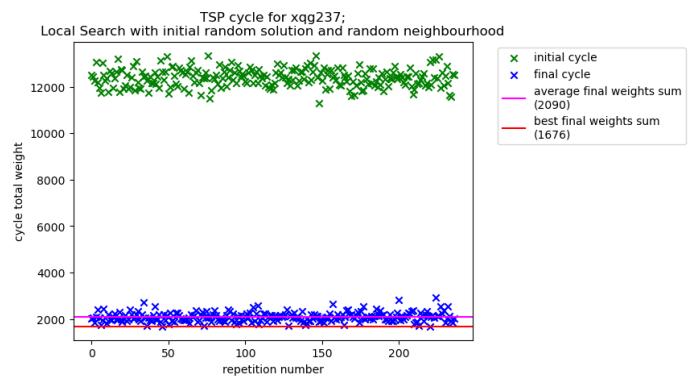
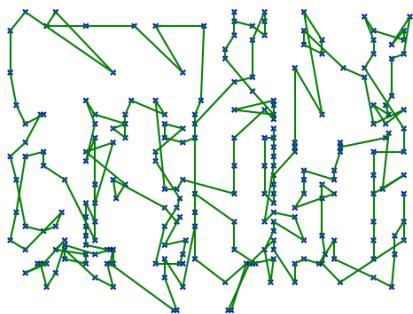
## 5.2 xqg237



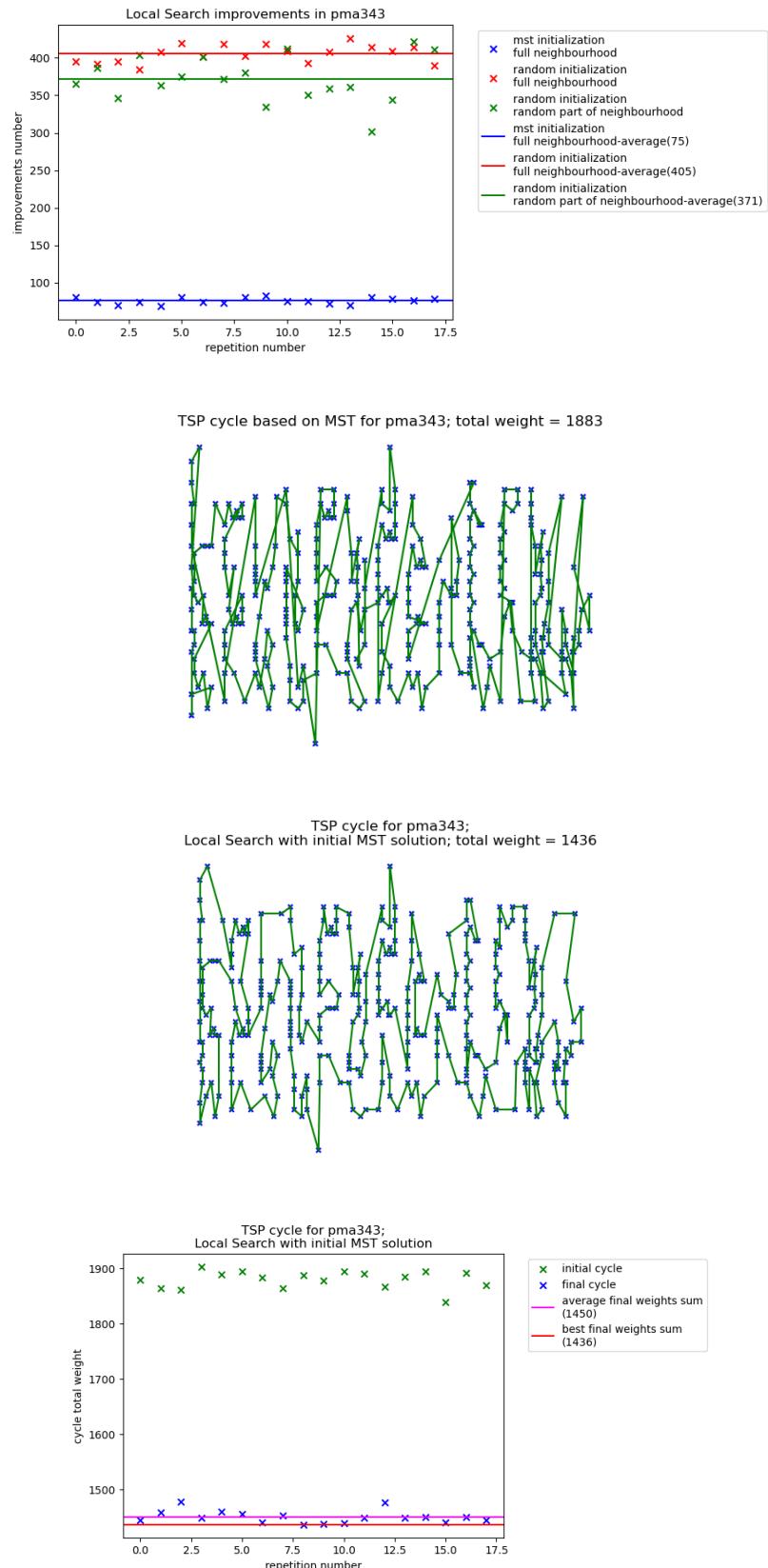
TSP cycle for xqg237;  
Local Search with initial random solution; total weight = 1062



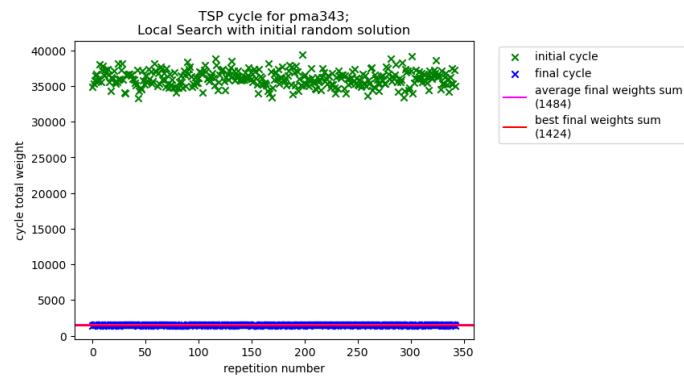
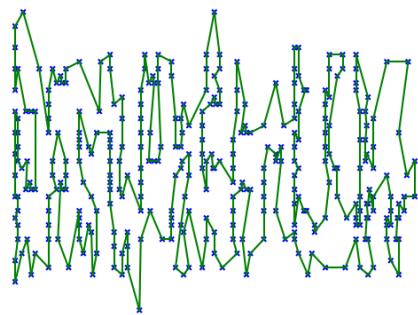
TSP cycle for xqg237;  
Local Search with initial random solution and random neighbourhood; total weight = 1676



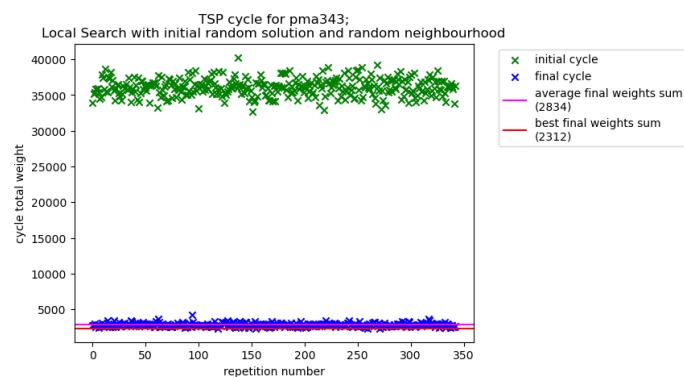
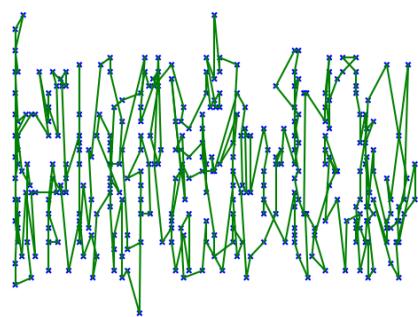
### 5.3 pma343



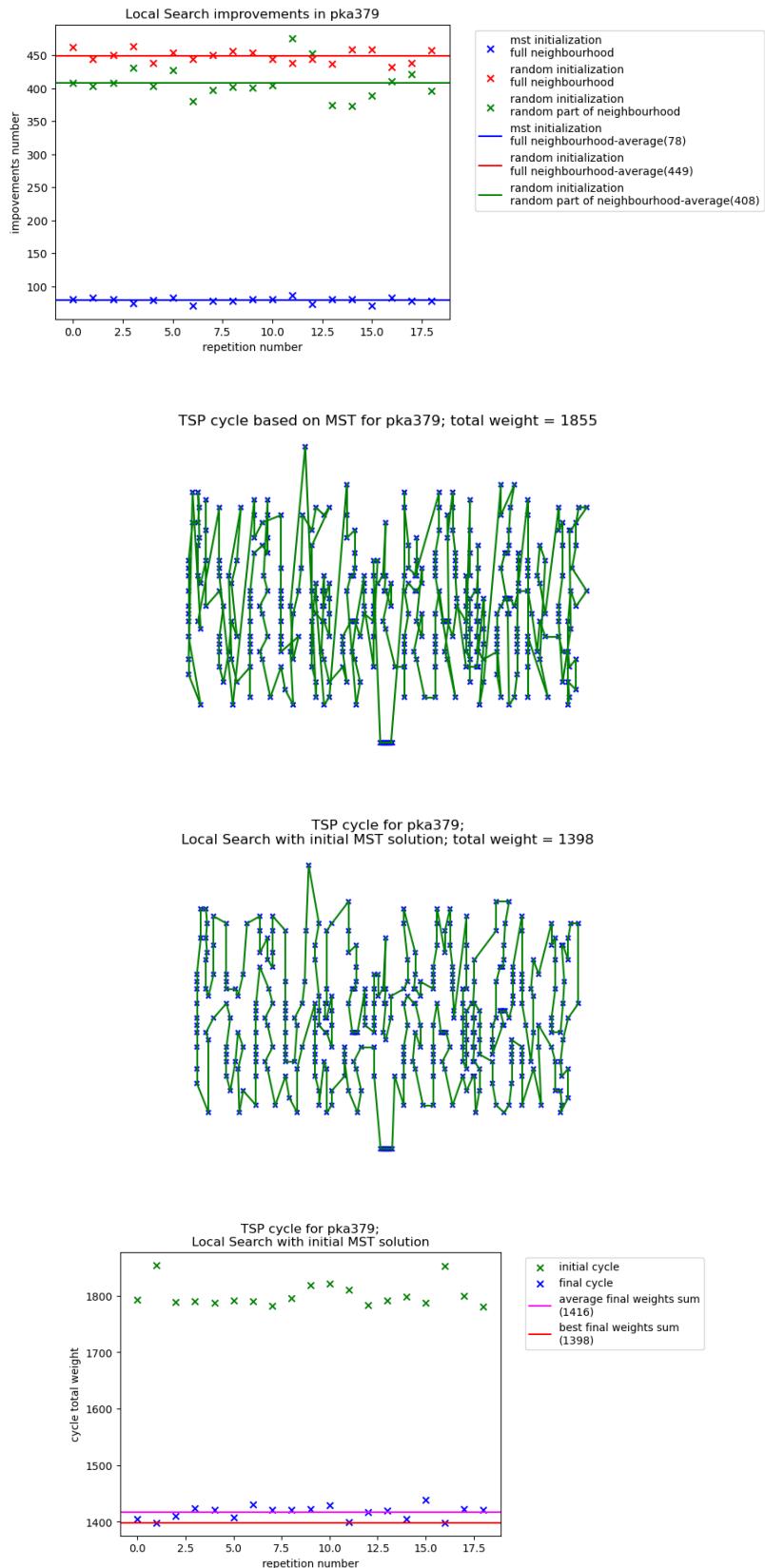
TSP cycle for pma343;  
Local Search with initial random solution; total weight = 1424



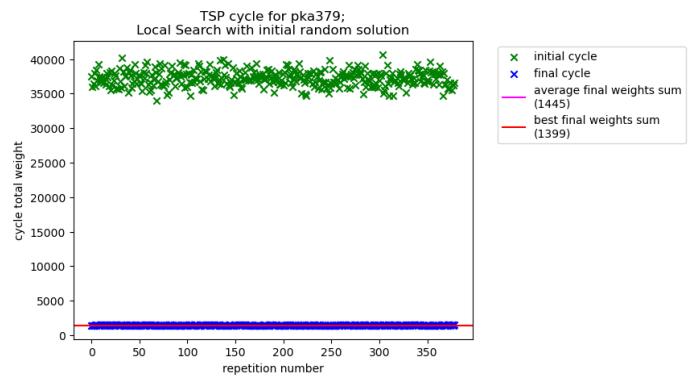
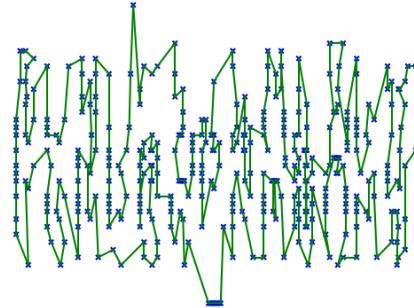
TSP cycle for pma343;  
Local Search with initial random solution and random neighbourhood; total weight = 2312



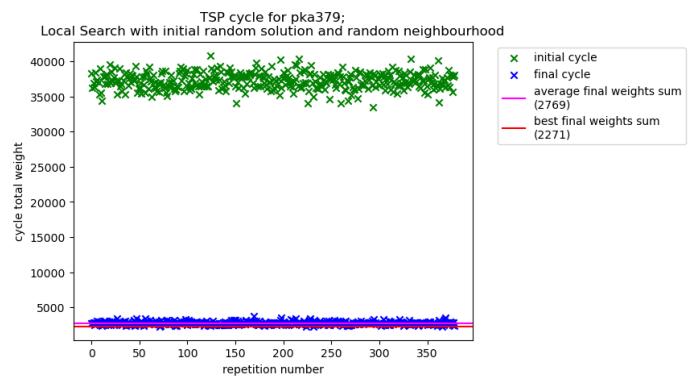
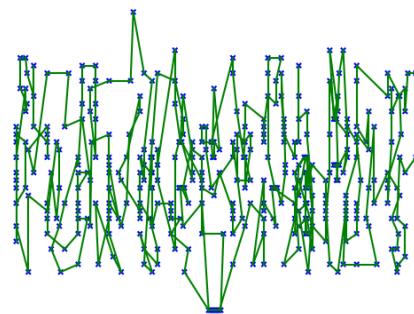
## 5.4 pka379



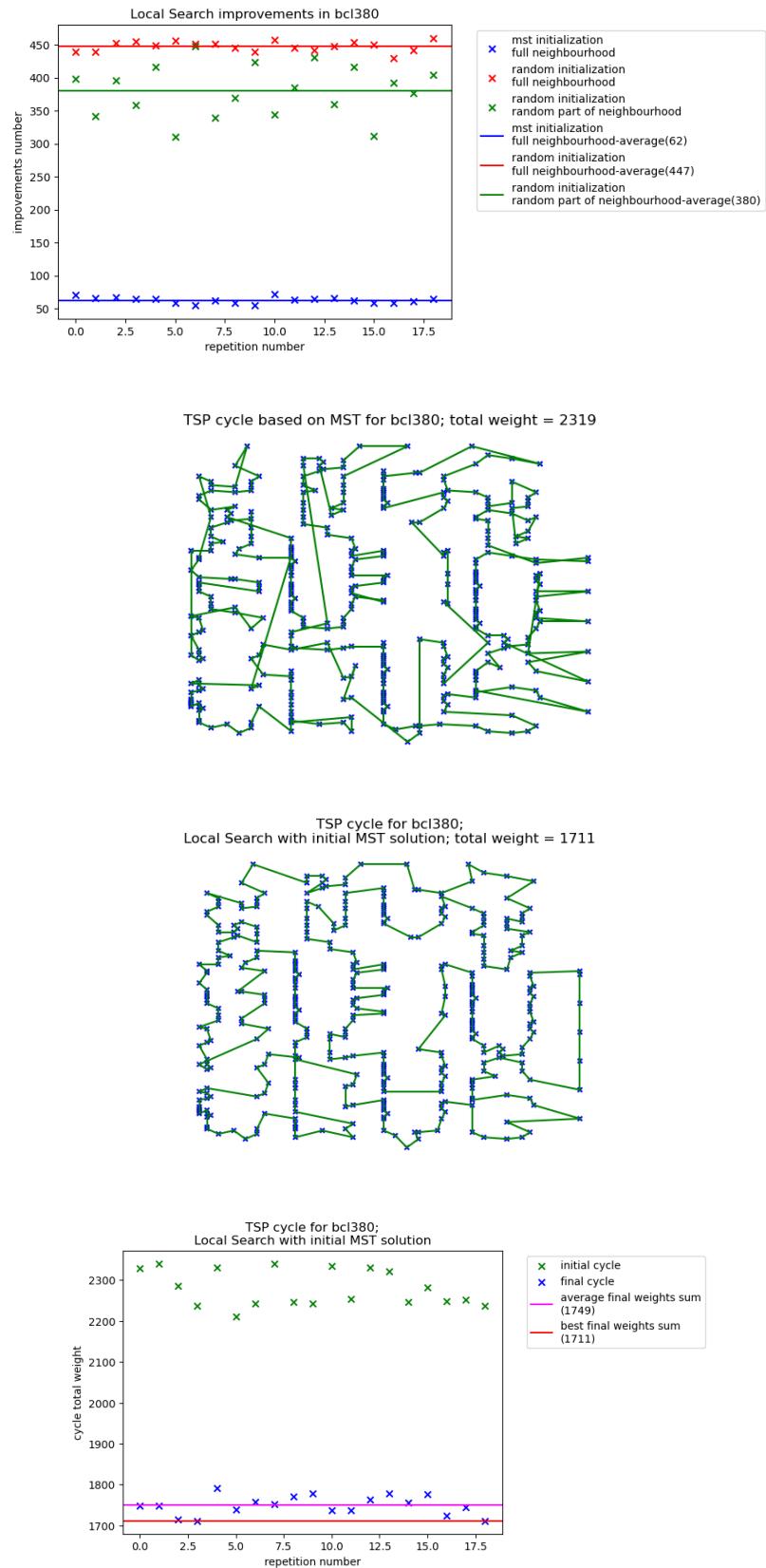
TSP cycle for pka379;  
Local Search with initial random solution; total weight = 1399



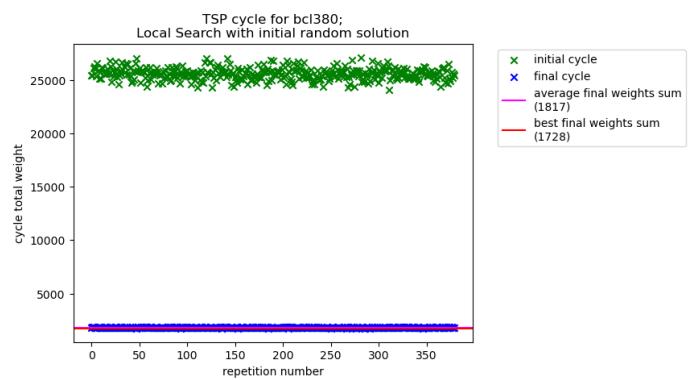
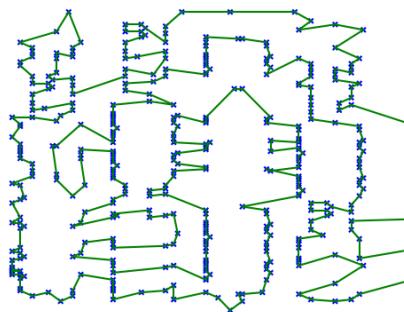
TSP cycle for pka379;  
Local Search with initial random solution and random neighbourhood; total weight = 2271



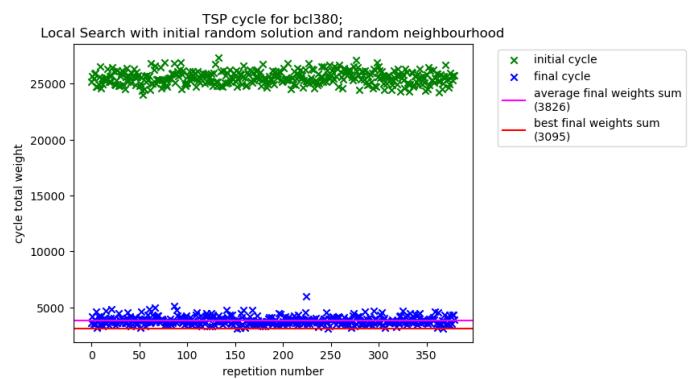
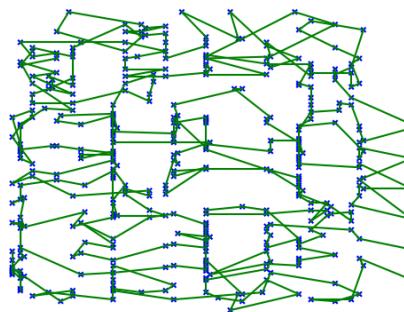
## 5.5 bcl380



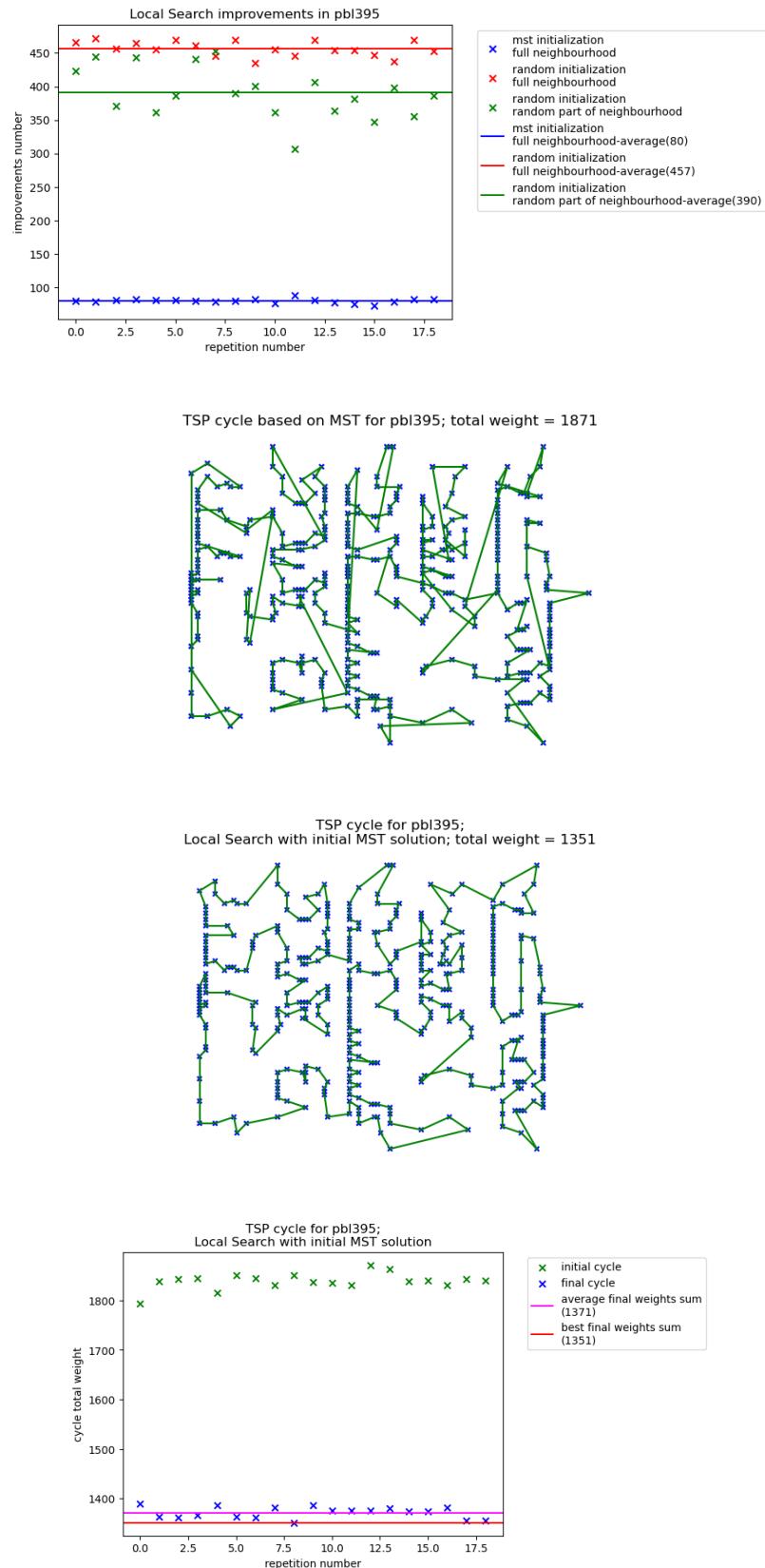
TSP cycle for bcl380;  
Local Search with initial random solution; total weight = 1728



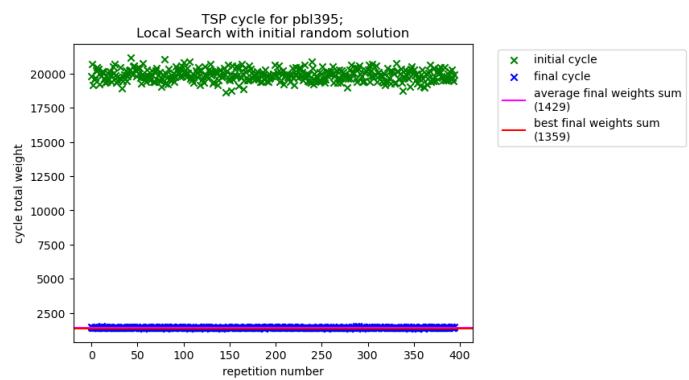
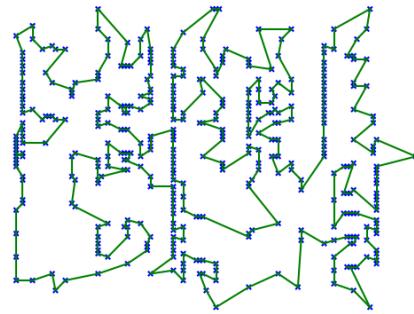
TSP cycle for bcl380;  
Local Search with initial random solution and random neighbourhood; total weight = 3095



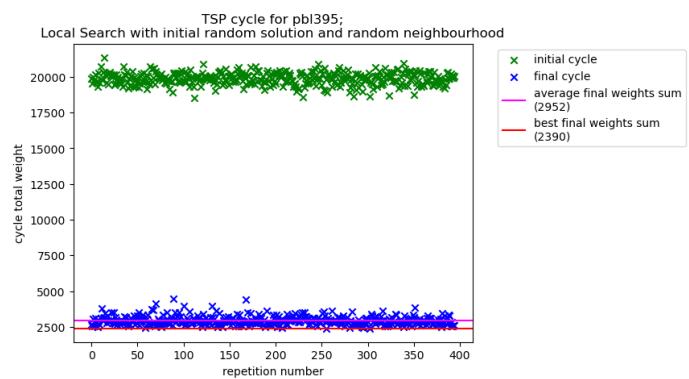
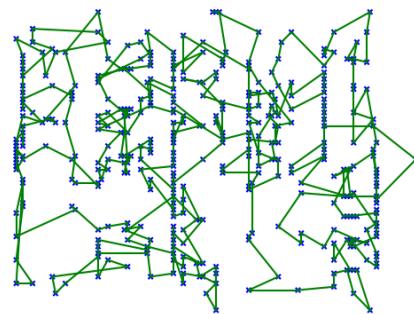
## 5.6 pbl395



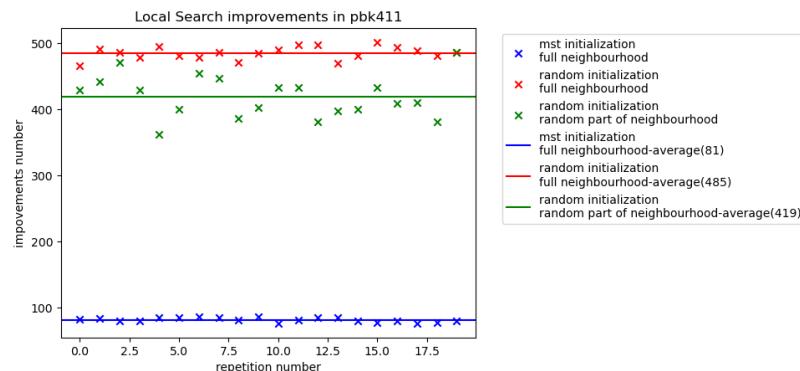
TSP cycle for pbl395;  
Local Search with initial random solution; total weight = 1359



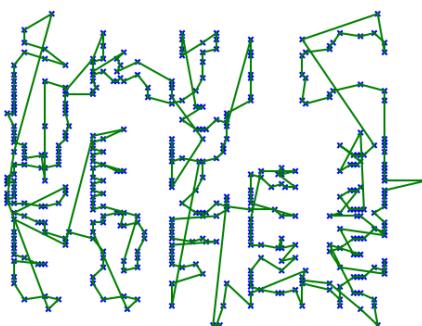
TSP cycle for pbl395;  
Local Search with initial random solution and random neighbourhood; total weight = 2390



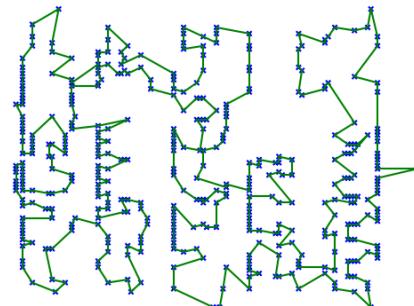
## 5.7 pbk411



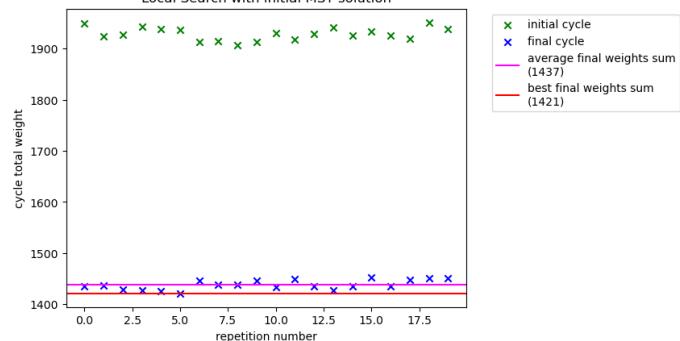
TSP cycle based on MST for pbk411; total weight = 1935



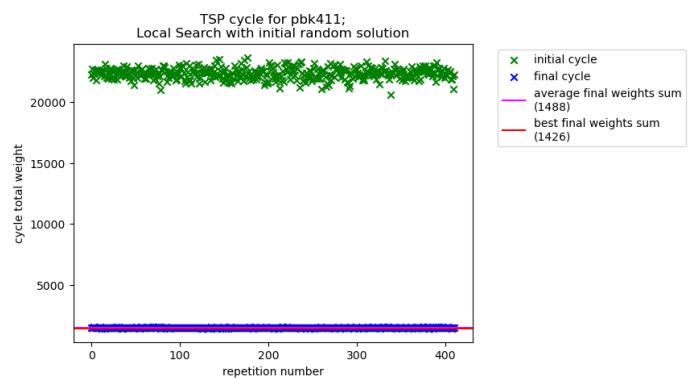
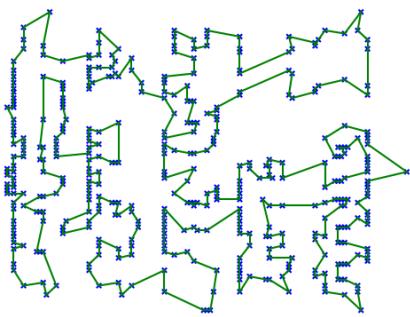
TSP cycle for pbk411;  
Local Search with initial MST solution; total weight = 1421



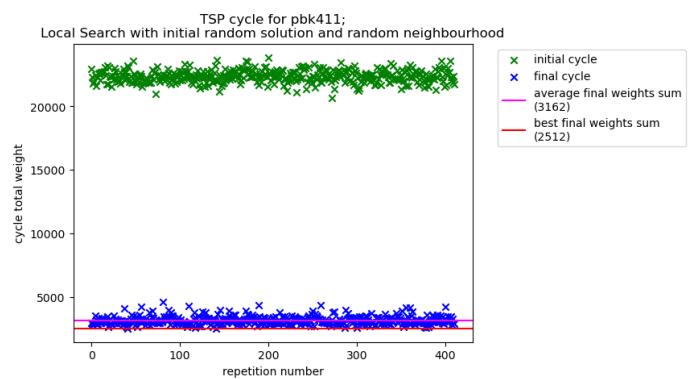
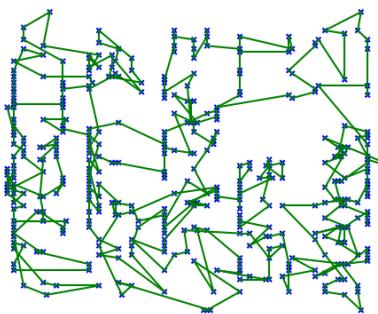
TSP cycle for pbk411;  
Local Search with initial MST solution



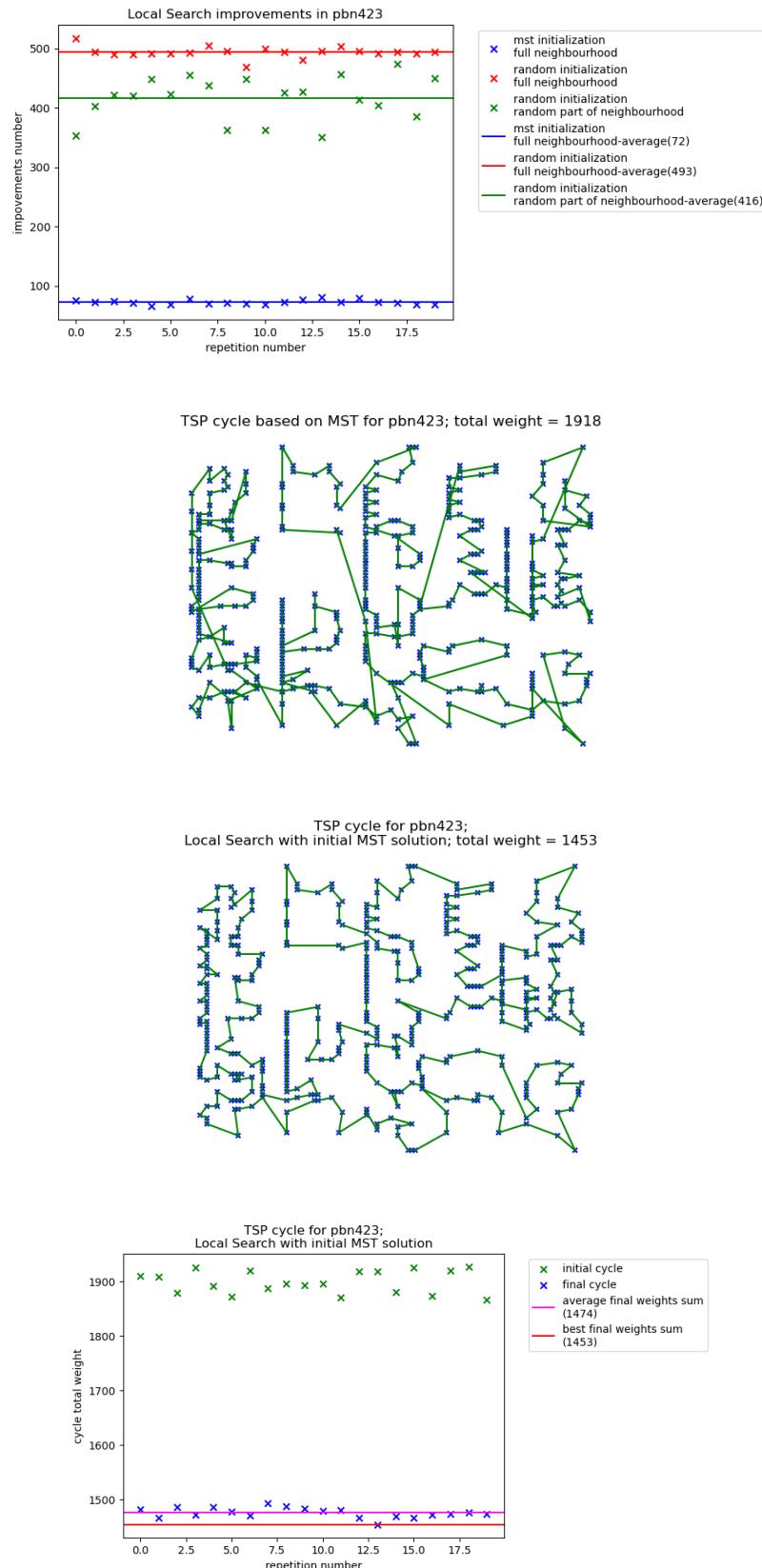
TSP cycle for pbk411;  
Local Search with initial random solution; total weight = 1426



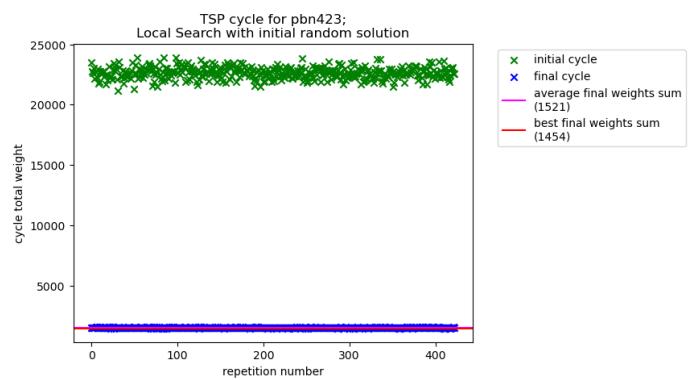
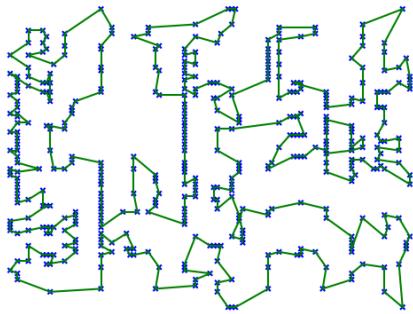
TSP cycle for pbk411;  
Local Search with initial random solution and random neighbourhood; total weight = 2512



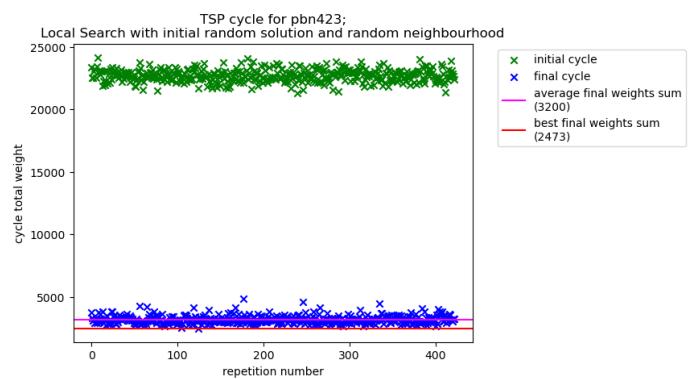
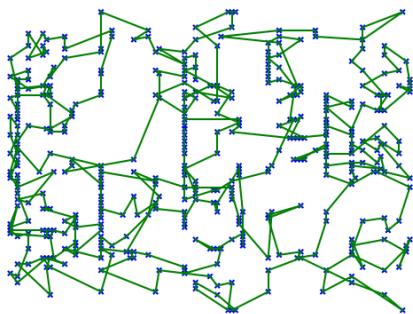
## 5.8 pbn423



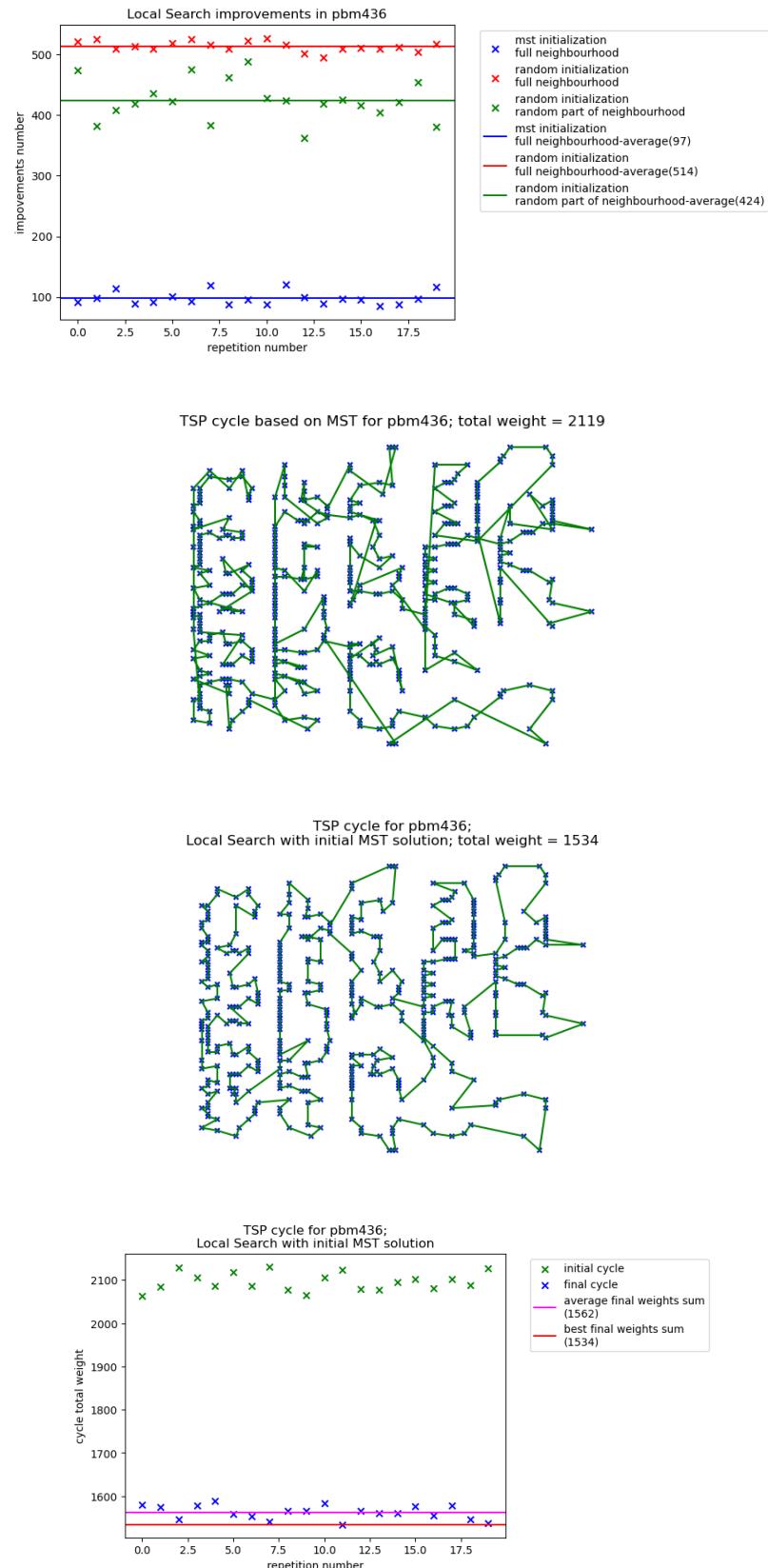
TSP cycle for pbn423;  
Local Search with initial random solution; total weight = 1454



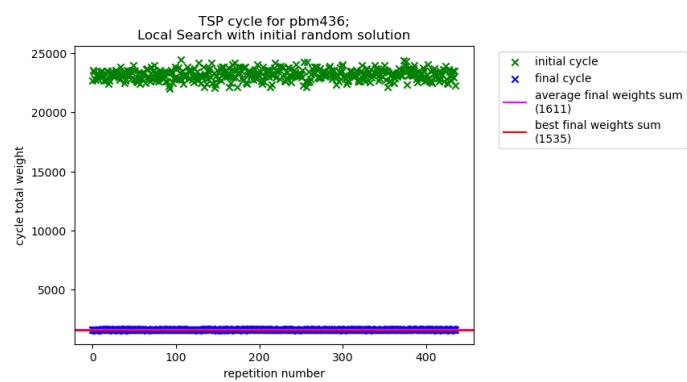
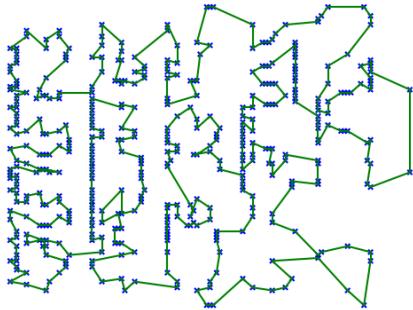
TSP cycle for pbn423;  
Local Search with initial random solution and random neighbourhood; total weight = 2473



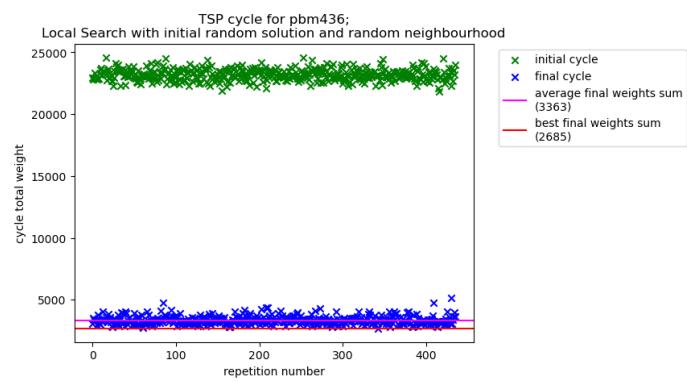
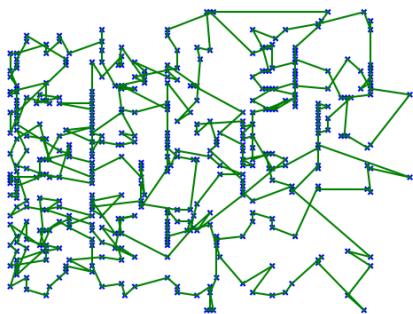
## 5.9 pbm436



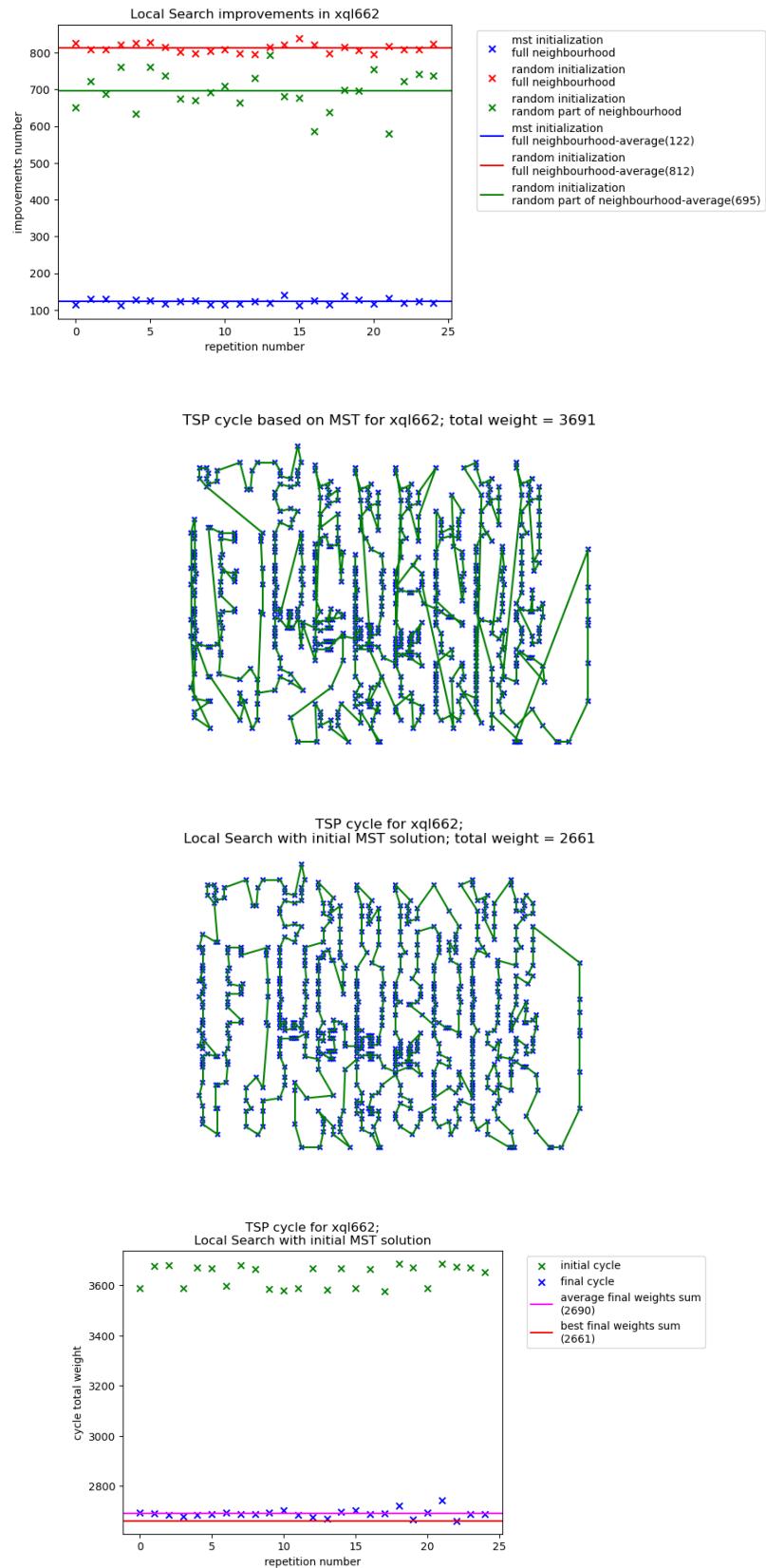
TSP cycle for pbm436;  
Local Search with initial random solution; total weight = 1535



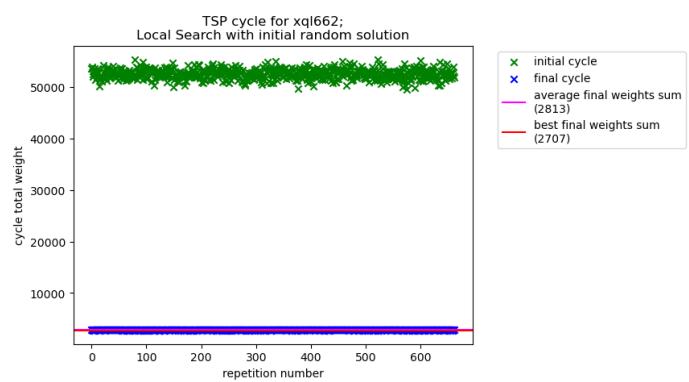
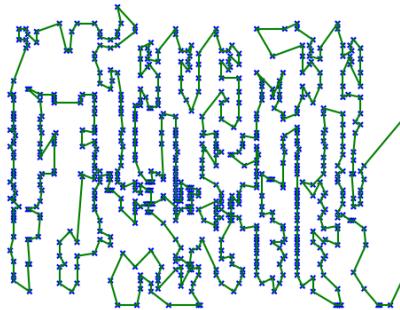
TSP cycle for pbm436;  
Local Search with initial random solution and random neighbourhood; total weight = 2685



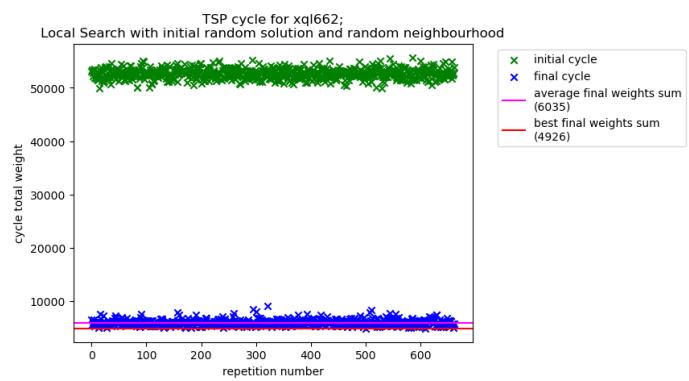
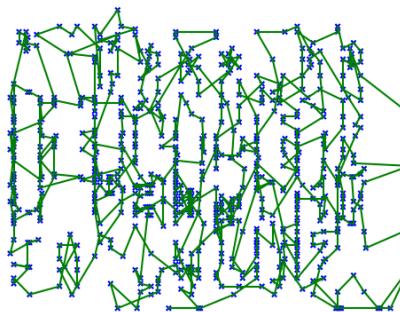
## 5.10 xql662



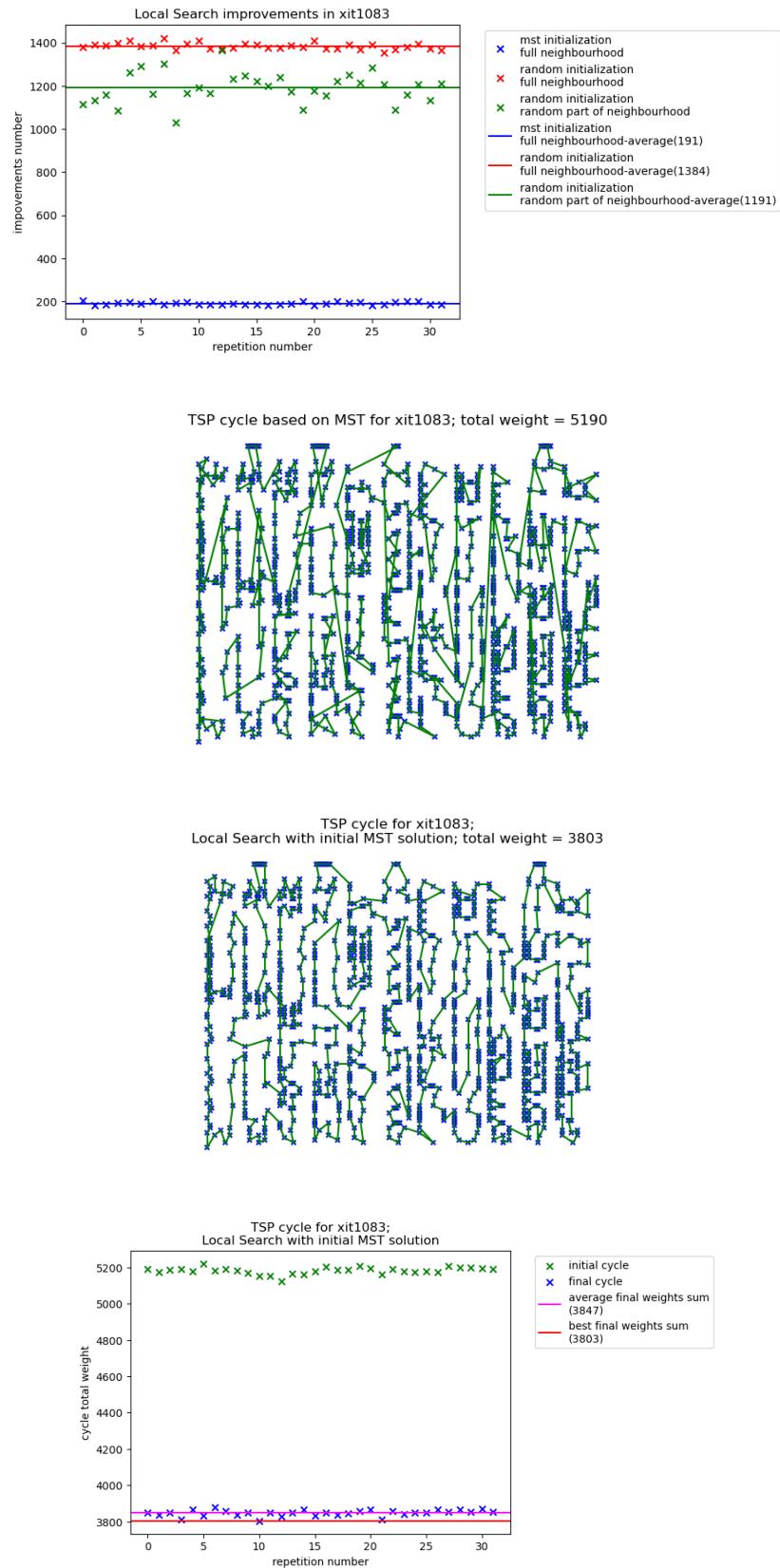
TSP cycle for xql662;  
Local Search with initial random solution; total weight = 2707



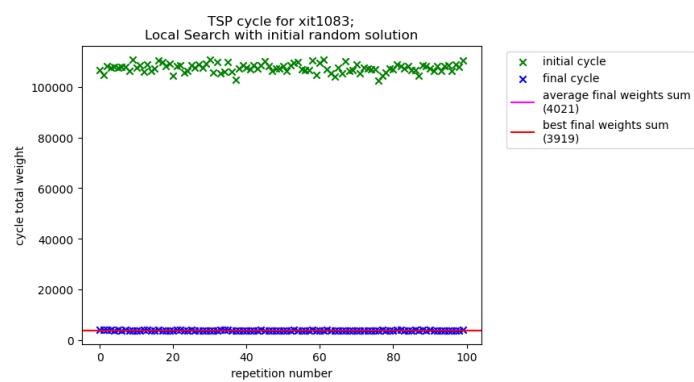
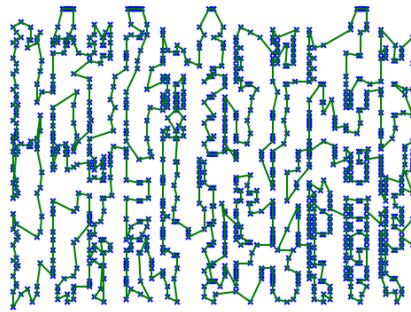
TSP cycle for xql662;  
Local Search with initial random solution and random neighbourhood; total weight = 4926



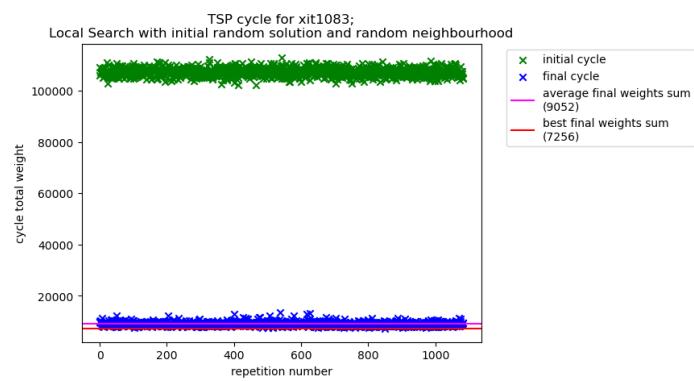
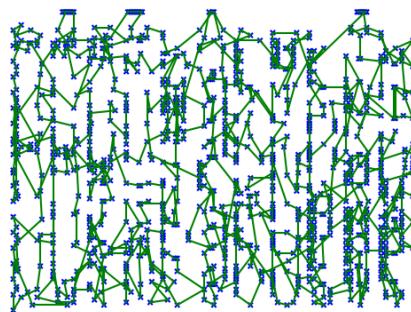
## 5.11 xit1083



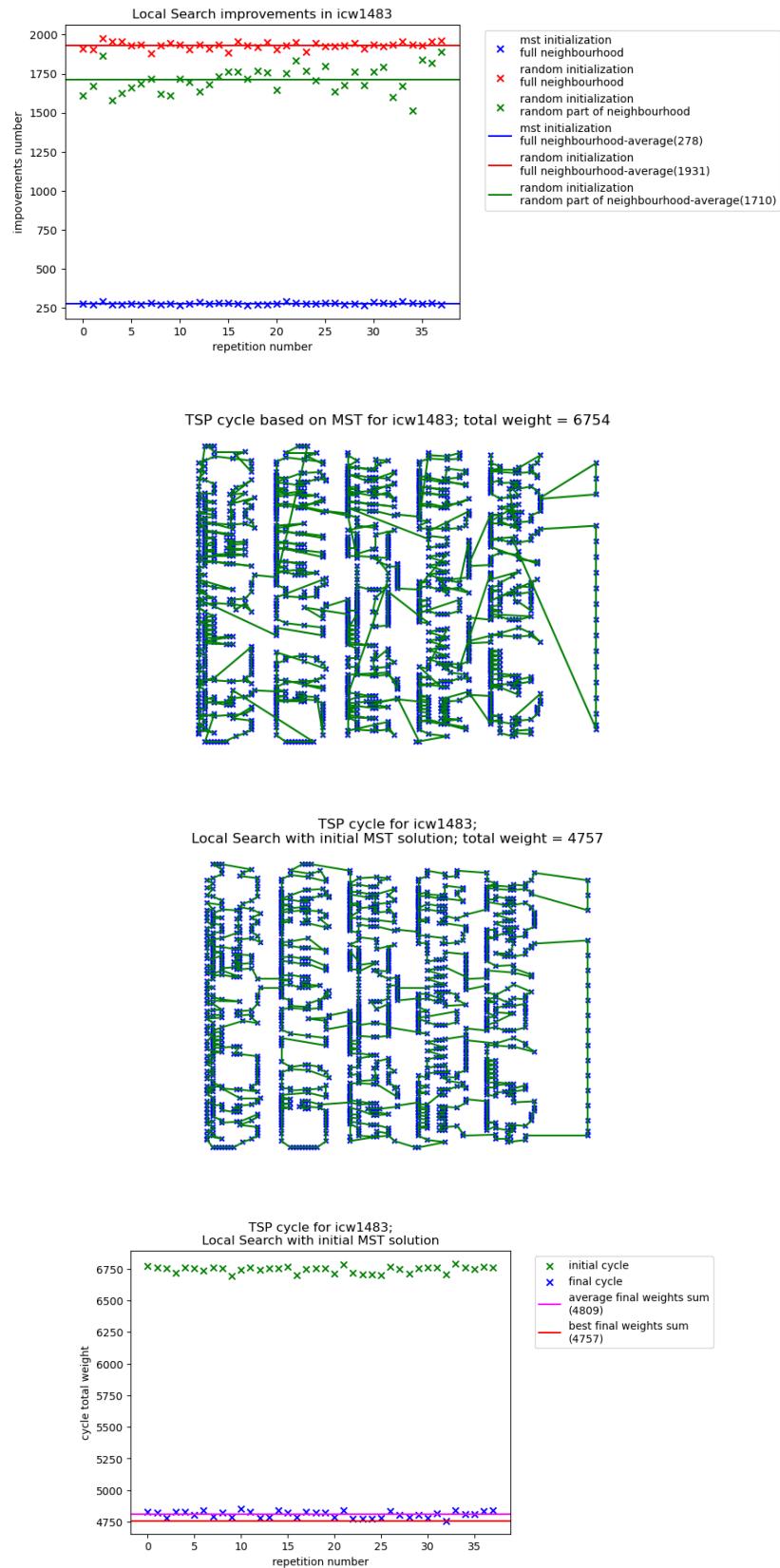
TSP cycle for xit1083;  
Local Search with initial random solution; total weight = 3919



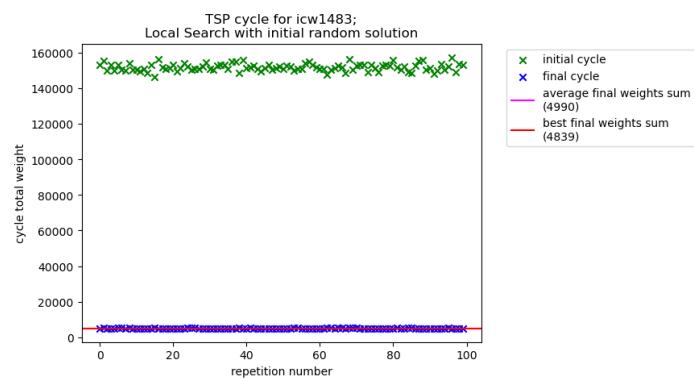
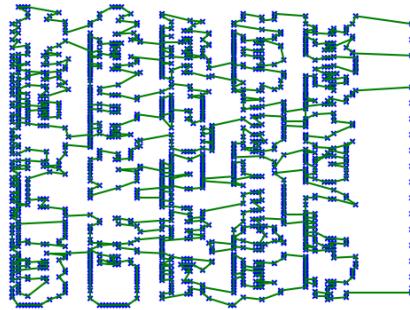
TSP cycle for xit1083;  
Local Search with initial random solution and random neighbourhood; total weight = 7256



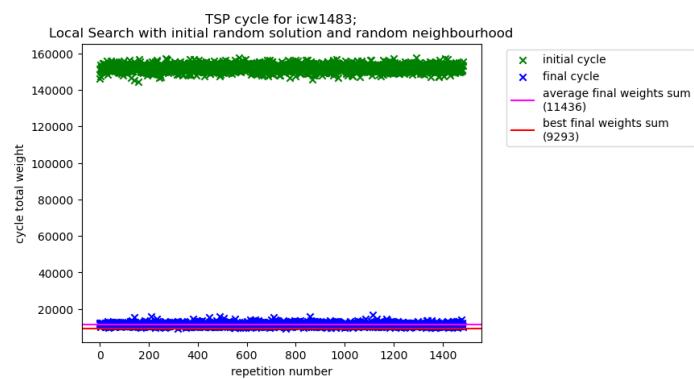
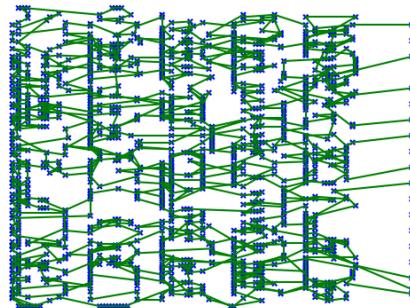
## 5.12 icw1483



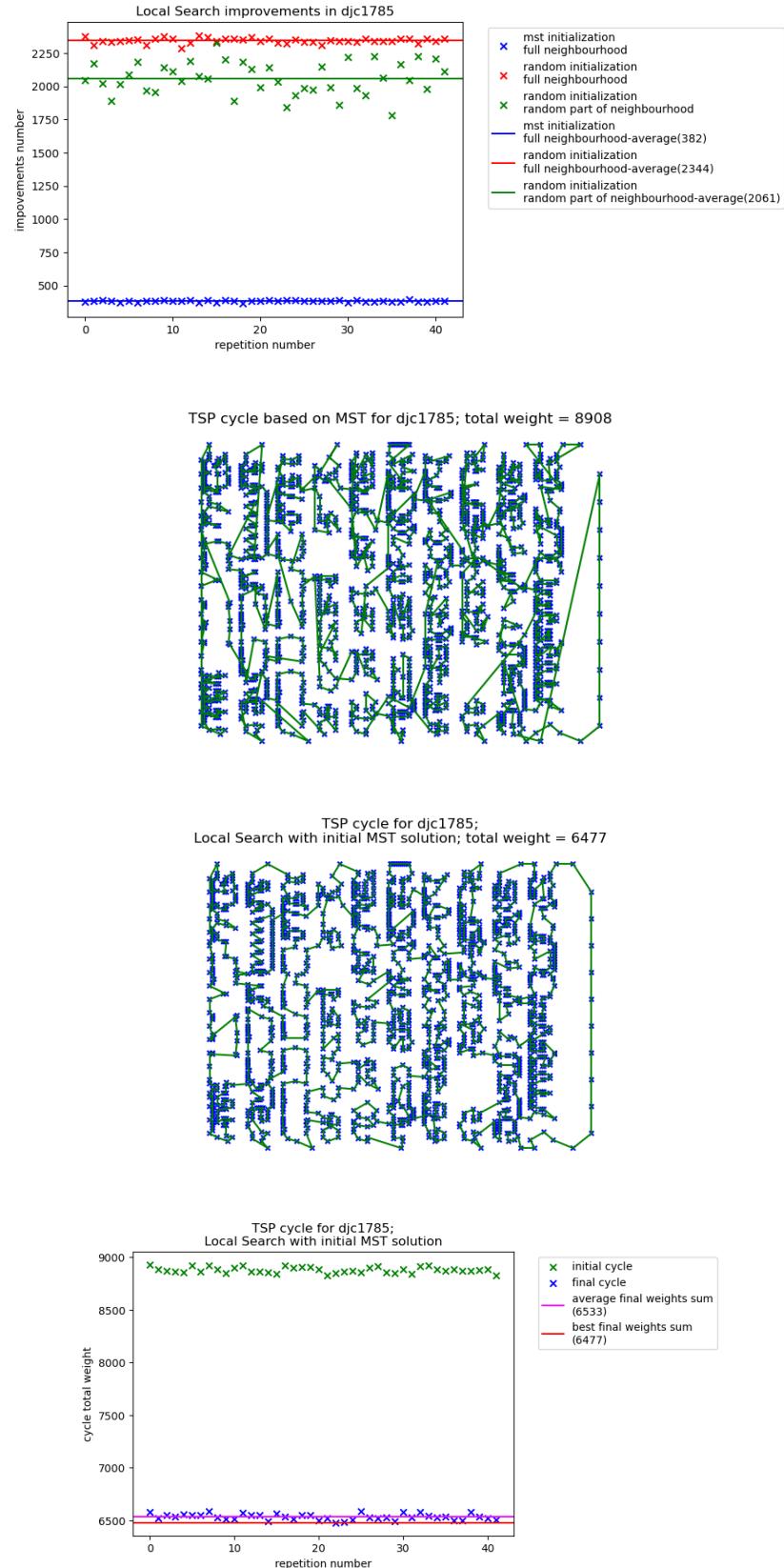
TSP cycle for icw1483;  
Local Search with initial random solution; total weight = 4839



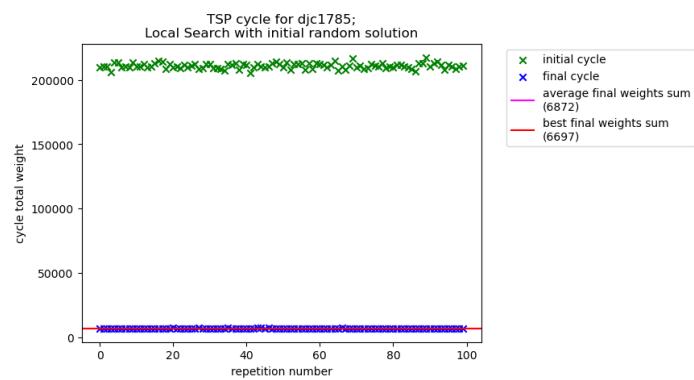
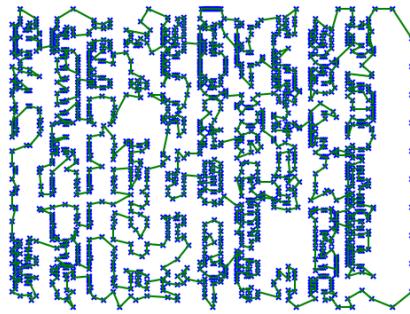
TSP cycle for icw1483;  
Local Search with initial random solution and random neighbourhood; total weight = 9293



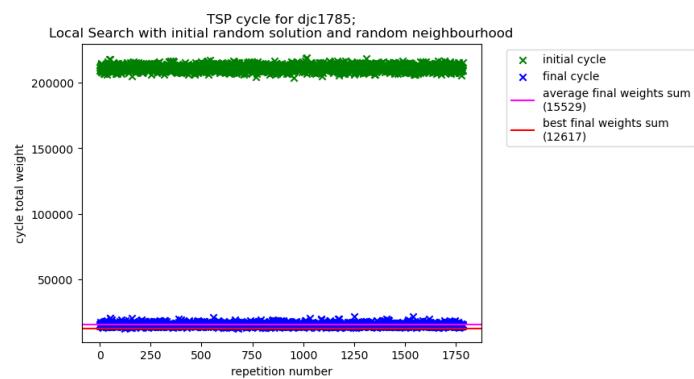
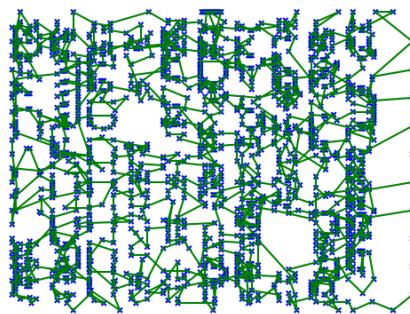
## 5.13 djc1785



TSP cycle for djc1785;  
Local Search with initial random solution; total weight = 6697



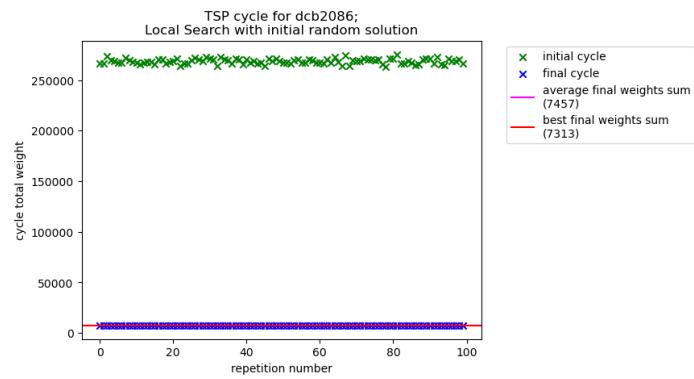
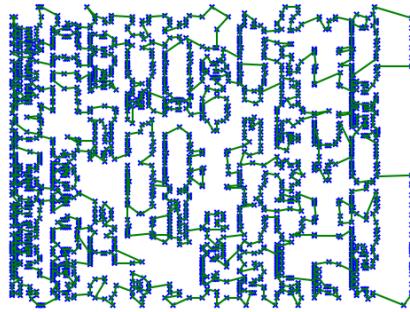
TSP cycle for djc1785;  
Local Search with initial random solution and random neighbourhood; total weight = 12617



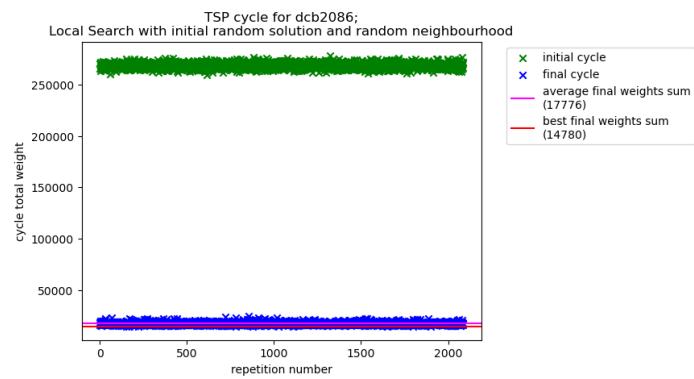
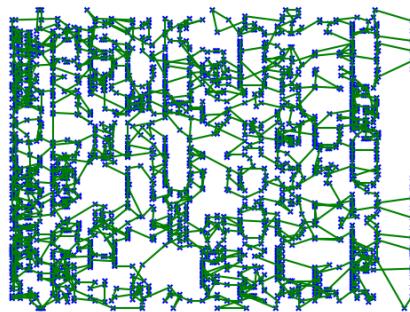
## 5.14 dcb2086



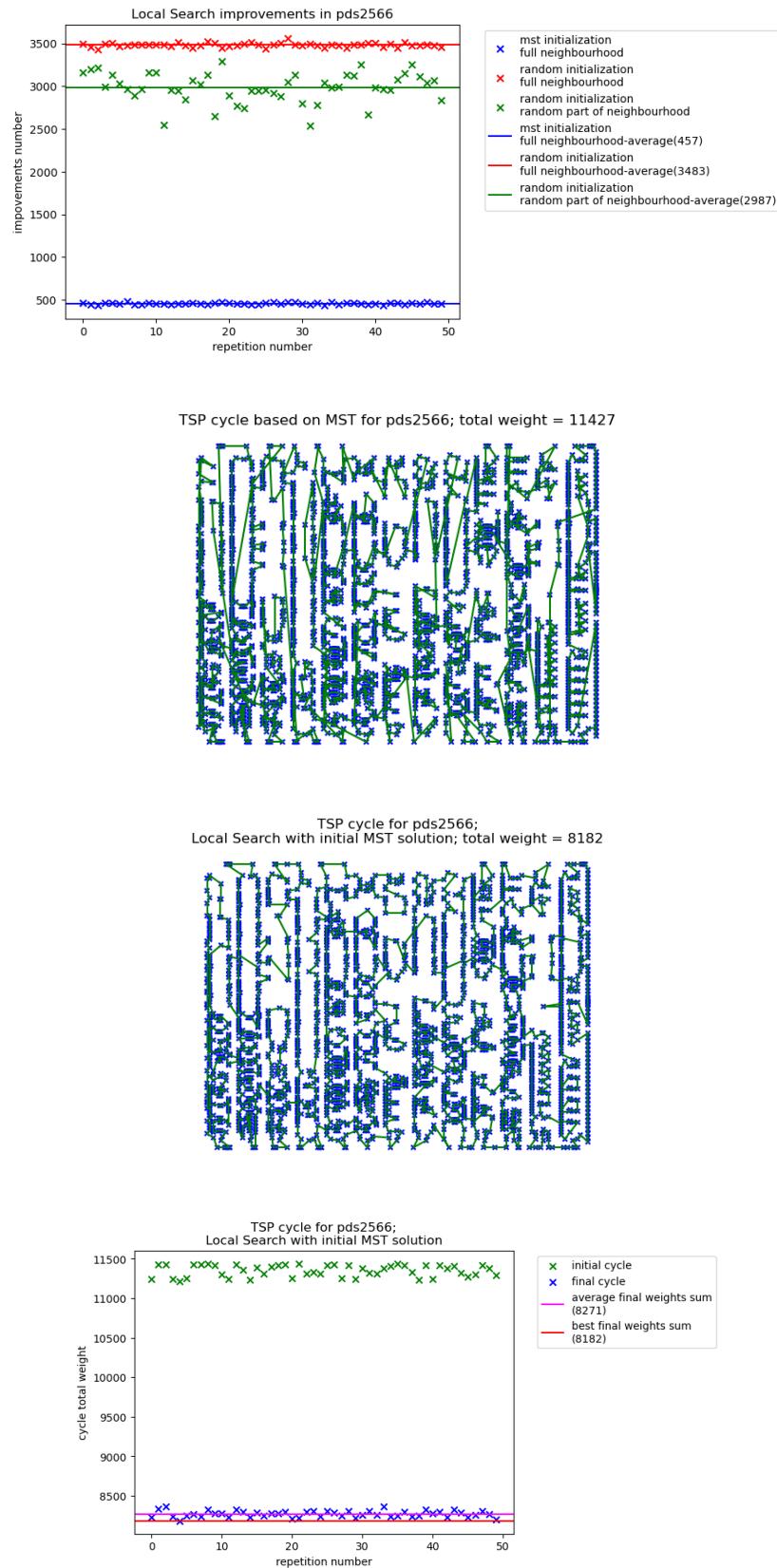
TSP cycle for dcb2086;  
Local Search with initial random solution; total weight = 7313



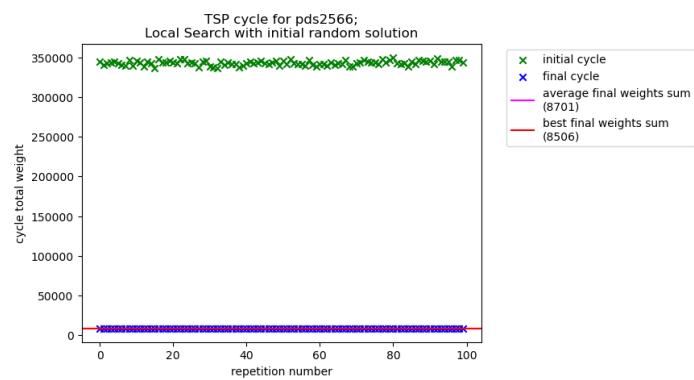
TSP cycle for dcb2086;  
Local Search with initial random solution and random neighbourhood; total weight = 14780



## 5.15 pds2566



TSP cycle for pds2566;  
Local Search with initial random solution; total weight = 8506



TSP cycle for pds2566;  
Local Search with initial random solution and random neighbourhood; total weight = 17506

