

Algorytmy metaheurystyczne 3

Paweł Cegieła, Wojciech Sęk

26 maja 2022

1 Rodzaje algorytmu

Algorytm genetyczny można podzielić ze względu na dwie znaczące modyfikacje:

- algorytm może działać na jednej populacji lub na wielu populacjach żyjących na osobnych wyspach
- algorytm może działać całkowicie sekwencyjnie lub wykonywać część obliczeń równolegle

2 Parametry algorytmów

Dla algorytmu sekwencyjnego z jedną populacją podajemy następujące parametry:

- *matrix*: macierz przechowująca odległości między miastami z problemu TSP
- *gen_rand*: wartość logiczna mówiąca, czy pierwsze pokolenie ma być całkowicie losowe czy przybliżone innymi metaheurystykami
- *gen_size*: licznosc populacji na każdym etapie algorytmu
- *elite_num*: licznosc elity, czyli najlepszych rozwiązań, które przeżywają i przechodzą do następnego pokolenia
- *cross_op*: wartość enumeratywna mówiąca o tym, jaki typ krzyżowania zostanie użyty w algorytmie. Możliwe wartości:
 - 0 - HALF CROSSOVER: dziecko w każdym kroku generowania przyjmuje kolejne od jednego z rodziców miasto z prawdopodobieństwem 1/2, bierzemy najwcześniejsze miasto z rodzica, które nie występuje w dziecku
 - 1 - ORDER CROSSOVER: pół dziecka to podciąg rodzica, drugie pół to podciąg drugiego rodzica składający się z nieużytych wcześniej miast i zaczynający od miasta występującego na tym miejscu w pierwszym rodzicu
 - 2 - CYCLE CROSSOVER: bierzemy miasto z pierwszego rodzica, patrzymy na miasto pod danym indeksem w drugim rodzicu, idziemy do niego w pierwszym rodzicu, powtarzamy aż zamknijemy cykl. Resztę uzupełniamy drugim rodzicem.
 - 3 - PARTIALLY MAPPED CROSSOVER: na danym podciągu indeksów tworzymy mapę między rodzicami, następnie używamy pociągu z jednego rodzica i zmapowanych wartości z drugiego.
- *swap_change*: wartość logiczna decydująca o tym, czy w trakcie mutacji poruszamy się w sąsiedztwie *swap* czy *reverse*
- *size_of_tournament*: rozmiar turnieju przy losowaniu rodziców do tworzenia nowego pokolenia. Wartość ustawiona na 0 sprawia, że zamiast turnieju korzystamy z zasady ruletki, gdzie funkcja dopasowania jest odwrotnością wagi danej permutacji.
- *mut_chance*: prawdopodobieństwo, że dany osobnik zmutuje.
- *max_time*: ograniczenie czasowe, po którym algorytm kończy działanie. Wynik zwracany przez algorytm to najlepszy wynik znaleziony przed przekroczeniem limitu czasowego.

Dla algorytmu wyspowego podajemy także:

- *isles_num*: liczba wysp
- *migration_freq*: liczba iteracji, które dzielą od siebie moment wymiany genetycznej między wyspami

W algorytmach urównoleglonych podaje się również parametr *num_of_threads*, który mówi o liczbie wątków.

3 Teoretyczna złożoność

Rozważmy złożoności poszczególnych etapów w algorytmie sekwencyjnym bez wyp. Niech n to liczba miast, a k to liczność pokolenia:

- **Wyznaczanie populacji początkowej:** w języku Rust losowanie permutacji ma złożoność $O(1)$, zatem losowa generacja osobników ma złożoność $O(k)$. Dla przybliżeń przez inne metaheurystyki co najwyżej $\frac{k}{4}$ jest przybliżone, w naszym przypadku dajemy jeden raz $2 - OPT$ ze złożonością $O(n^3)$ i pozostałe to *nearest – neighbours* od losowego miasta ze złożonością $O(n^2)$.
- **Ewaluacja:** Obliczanie wartości danej permutacji ma złożoność $O(n)$, robimy to k razy, zatem cały krok ma $O(kn)$.
- **Selekcja:** Wybieramy k rodziców. Przy ruletce wybór rodzica ma złożoność $O(1)$, przy turnieju $O(l)$, gdzie l to rozmiar turnieju. Zatem $O(k)$ dla ruletki i $O(kl)$ dla turnieju.
- **Krzyżowanie:** Każde z krzyżowań odbywa się w czasie liniowym względem długości permutacji, czyli $O(n)$, przy tworzeniu całej permutacji mamy $O(kn)$.
- **Mutacja:** mutacja typu *swap* ma złożoność $\Theta(1)$, a typu *reverse* $\Theta(n)$. Mutacji wykonujemy $\Theta(pn)$, gdzie p to prawdopodobieństwo mutacji. Zatem ta faza ma złożoność $\Theta(pn)$ lub $\Theta(pn^2)$. W worst case możemy wylosować wszystkie, więc odpowiednio złożoności są $O(n)$ i $O(n^2)$.

Ponadto w algorytmie wyspowym regularnie w niektórych iteracjach wymieniana jest informacja genetyczna ze złożonością $O(k^2)$ (każda wyspa wymienia się z każdą jakimś osobnikiem).

Podsumowując, w najgorszym przypadku:

- **Generacja początkowej populacji:**

$$O(n^3 + kn^2 + k) = O((n + k)n^2)$$

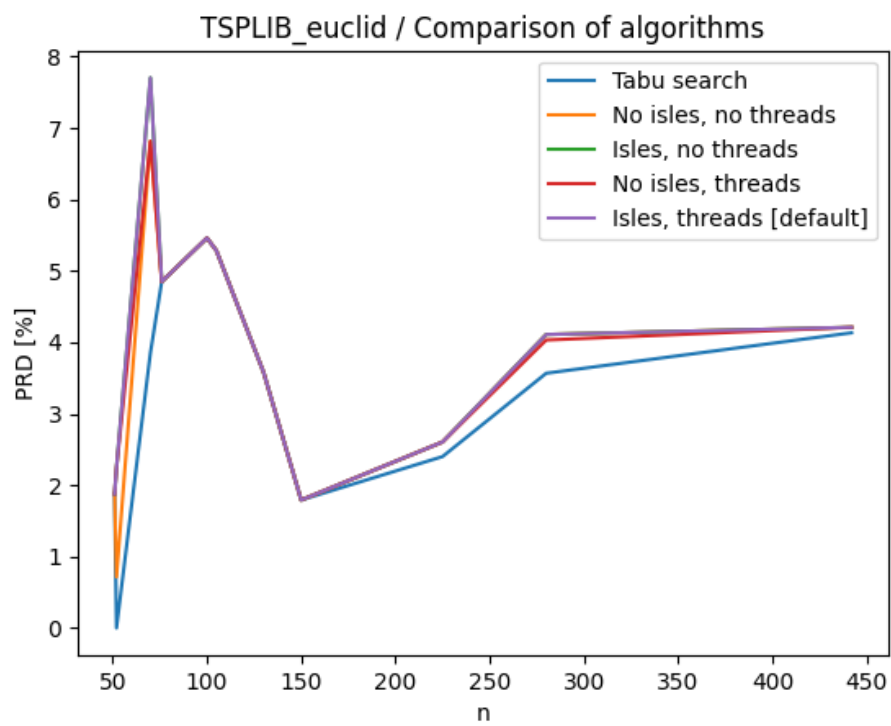
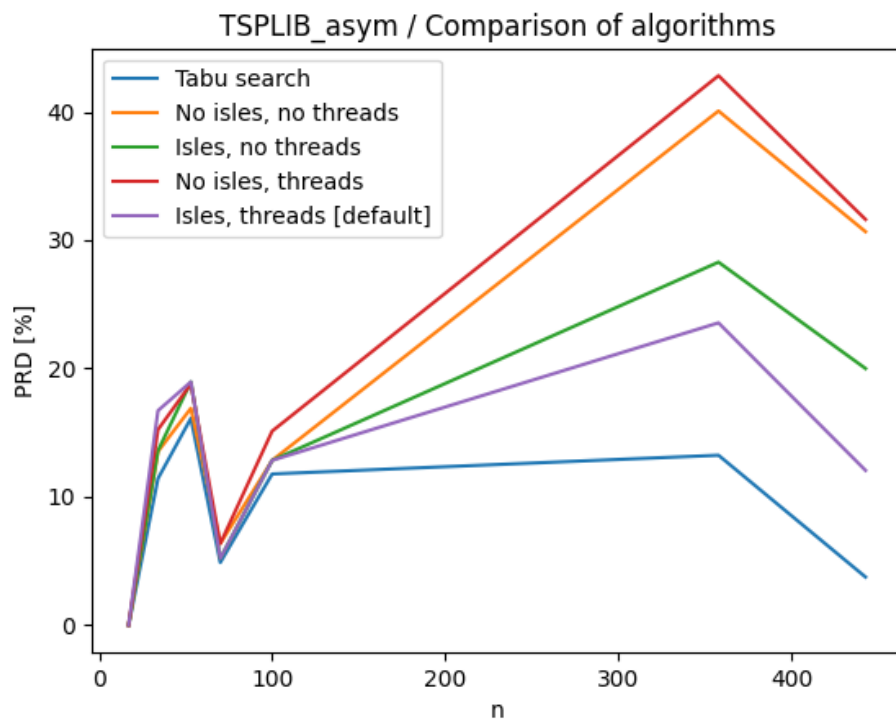
- **Właściwa iteracja:**

$$O(kn) + O(kl) + O(kn) + O(n^2) + O(k^2) = O(k(n + l + k) + n^2)$$

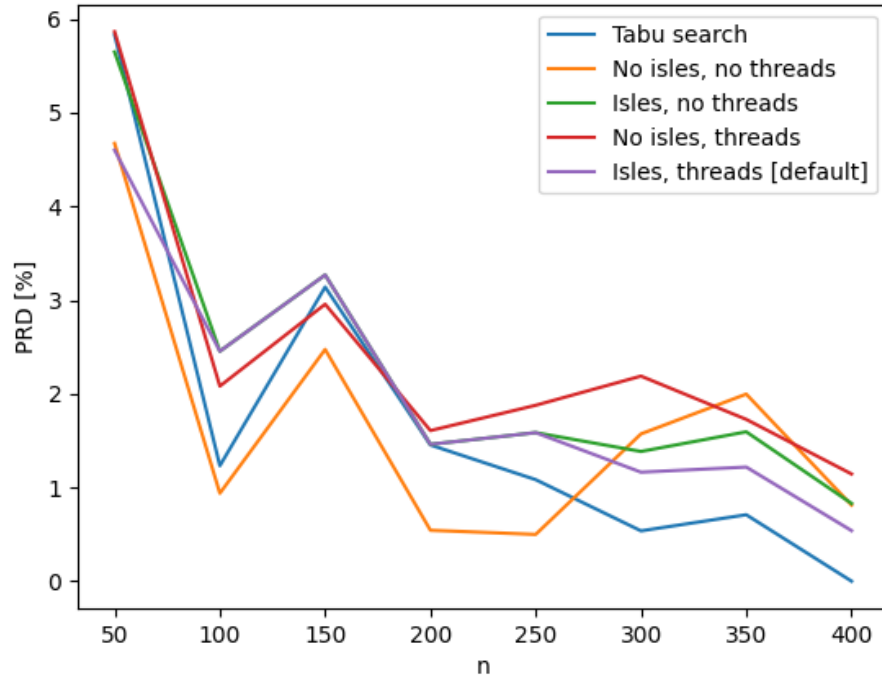
4 Wykonane eksperymenty

Strojenie algorytmu zostało wykonane na podstawie danych z TSPLIB. Wybrane zostały parametry, dla których algorytm najlepiej zachowywał się podczas testowania. Jeśli jeden z dwóch parametrów miał np. lepsze wyniki od drugiego dla jednych danych (np. dla mniejszych wielkości macierzy), a gorsze dla innych, wybierany był jeden z tych parametrów po przemyśleniu. Po strojeniu zostały przeprowadzone eksperymenty porównujące algorytm w wersji oraz z parametrami wybranymi jako domyślne: z innymi wersjami algorytmu oraz z tabu search, a także z różnymi wartościami parametrów. Nie w każdym eksperymencie czy nie dla każdego typu macierzy domyślny algorytm okazał się najlepszy, natomiast strojenie pomogło nam znaleźć taki algorytm oraz parametry, które średnio zachowują się najlepiej.

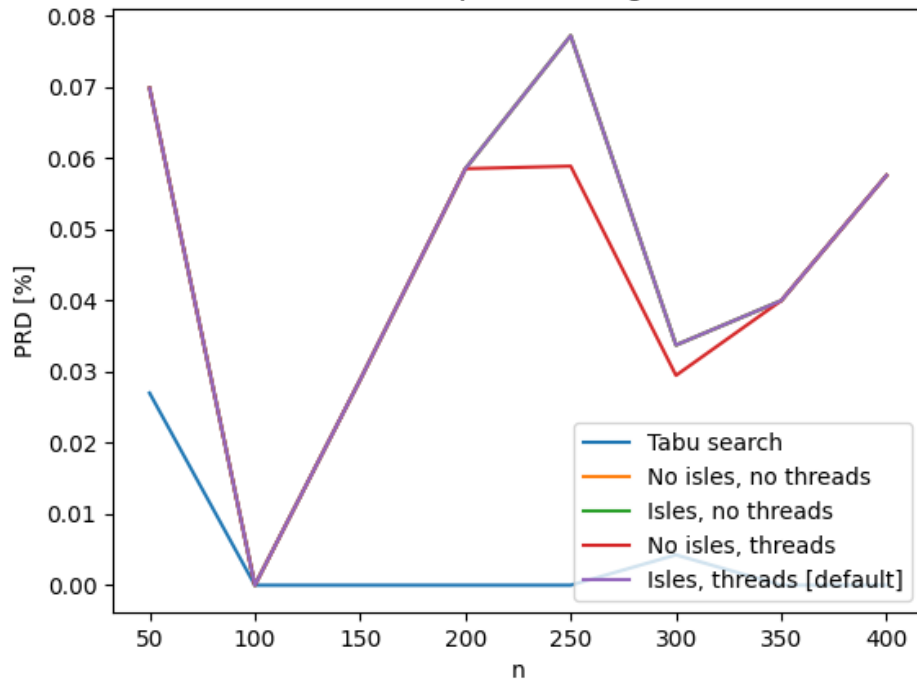
4.1 Porównanie wersji algorytmu genetycznego i tabu search

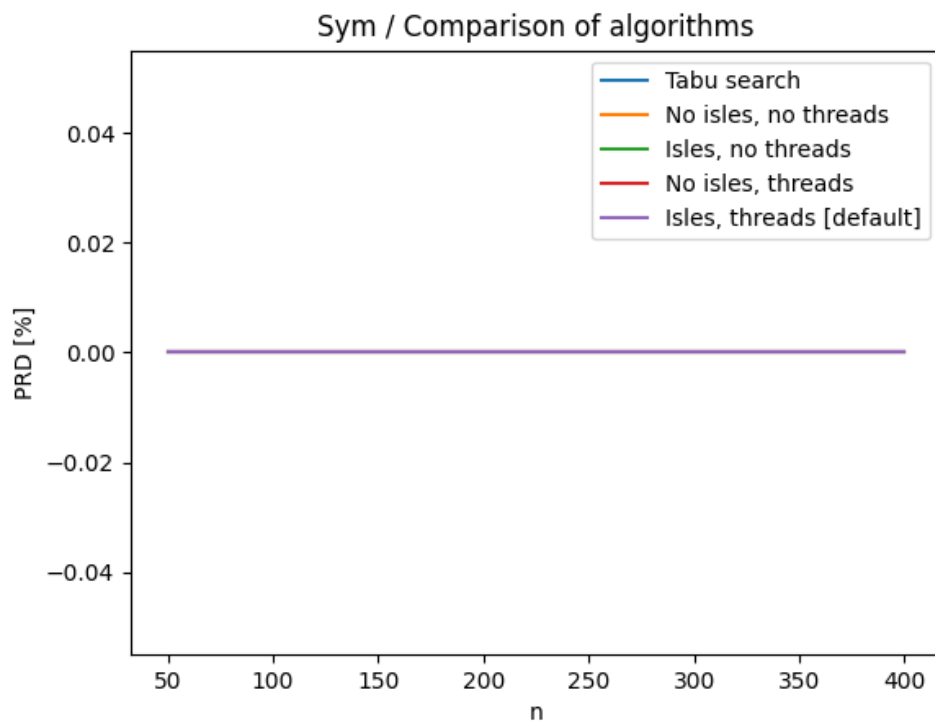


Asym / Comparison of algorithms

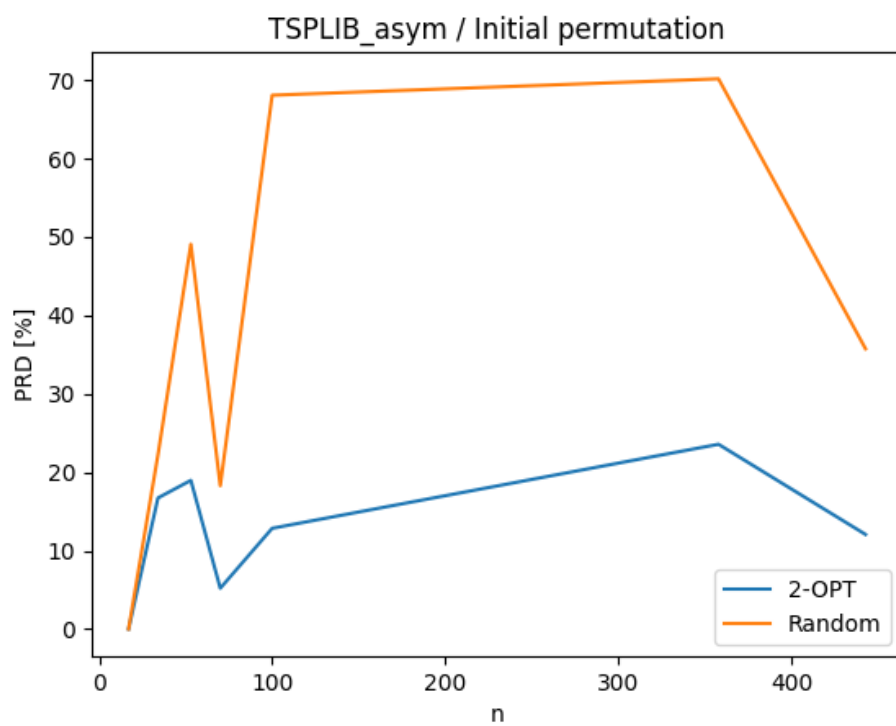


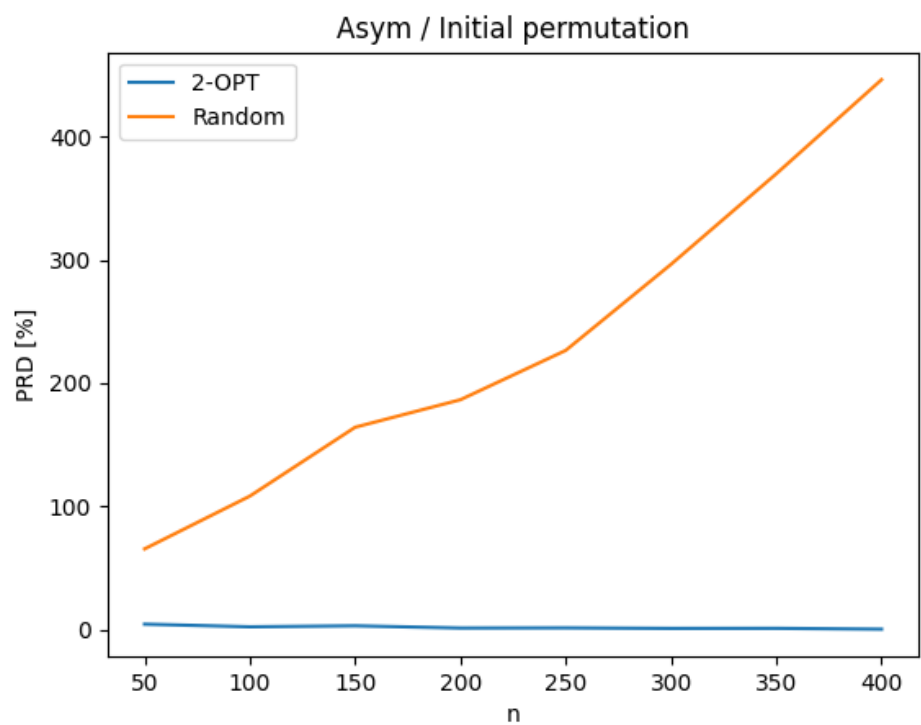
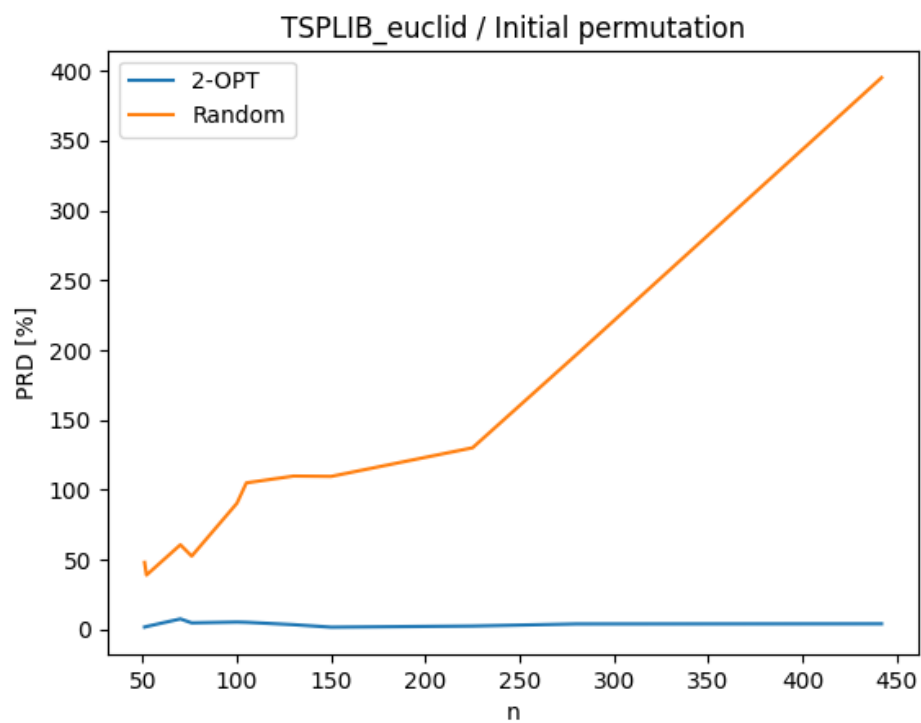
Euclid / Comparison of algorithms

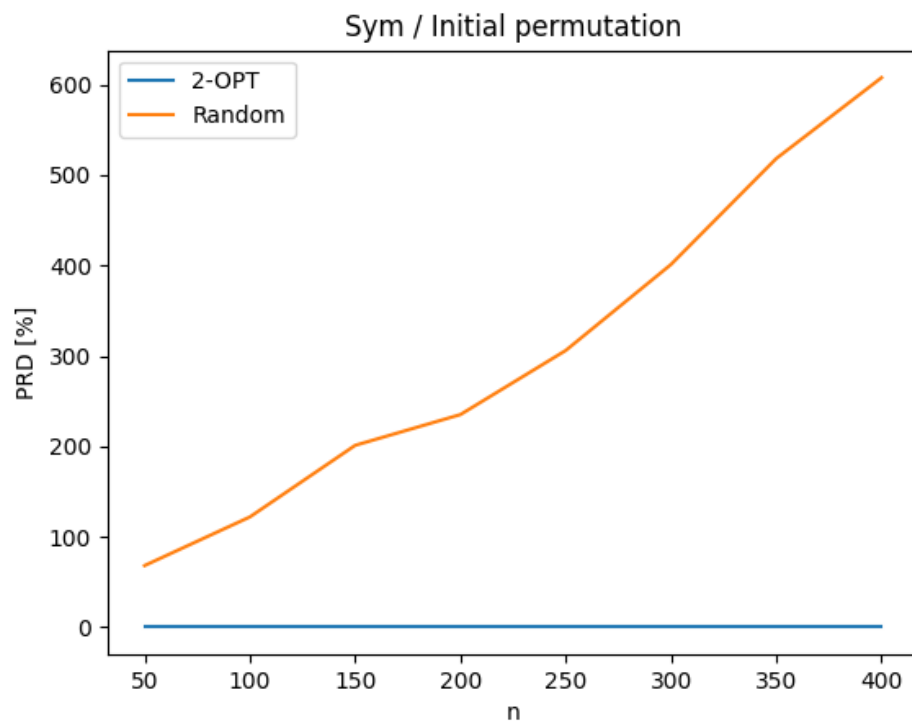
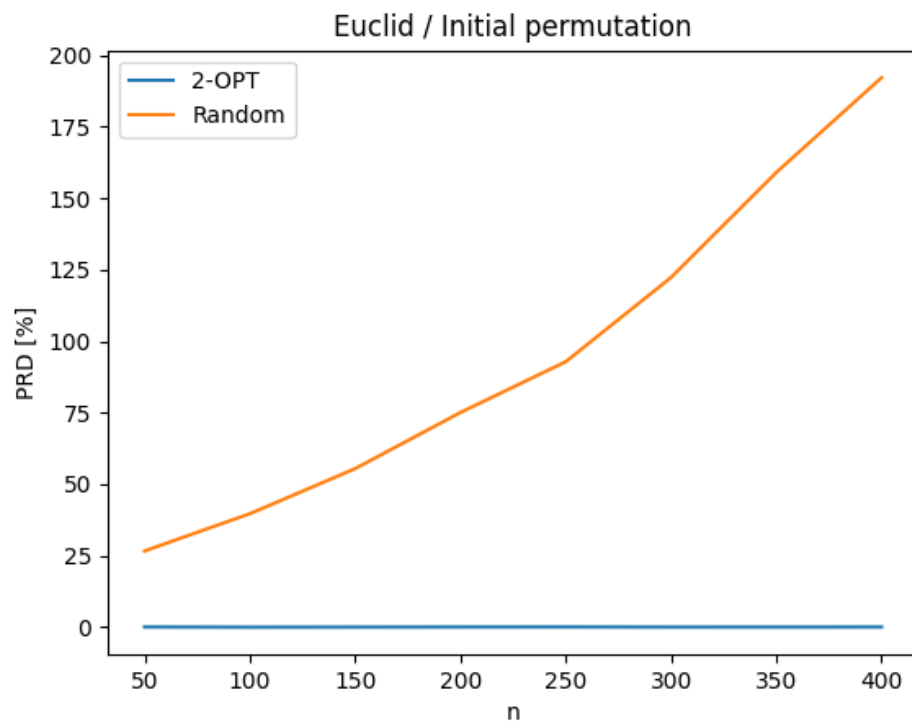




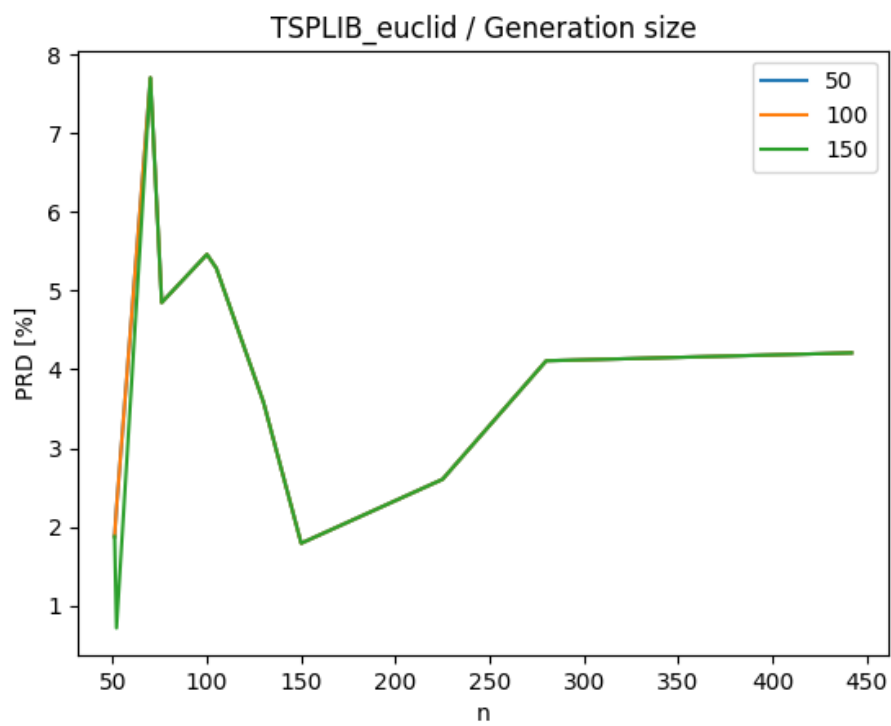
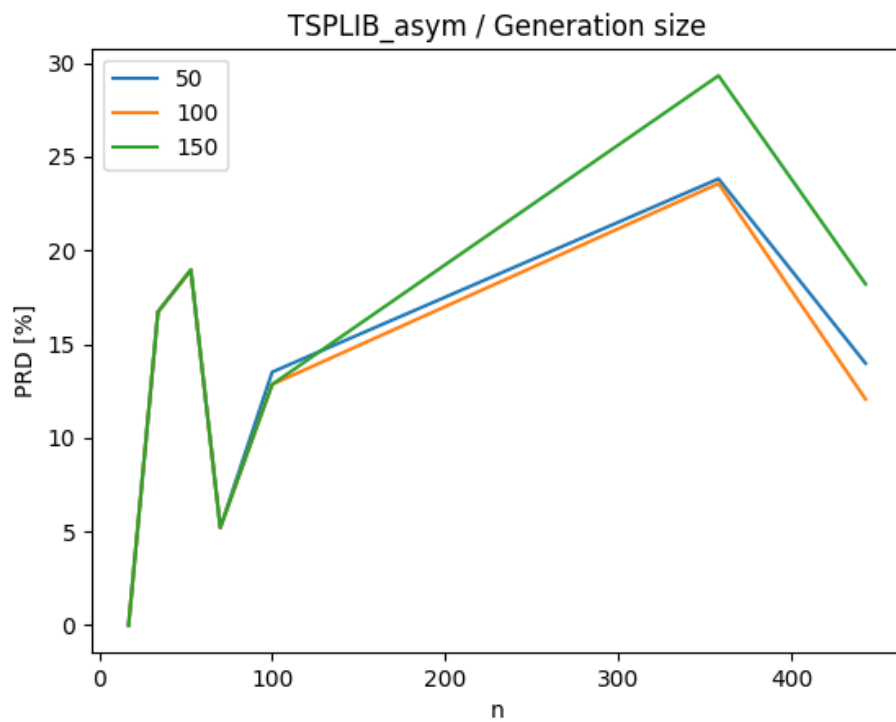
4.2 Przybliżenie początkowe

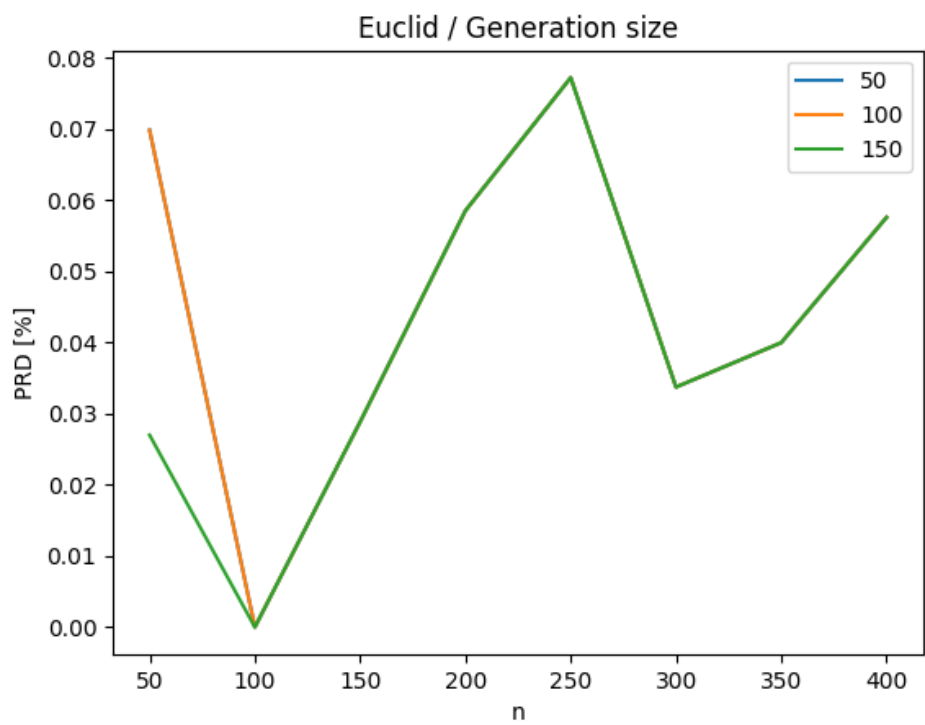
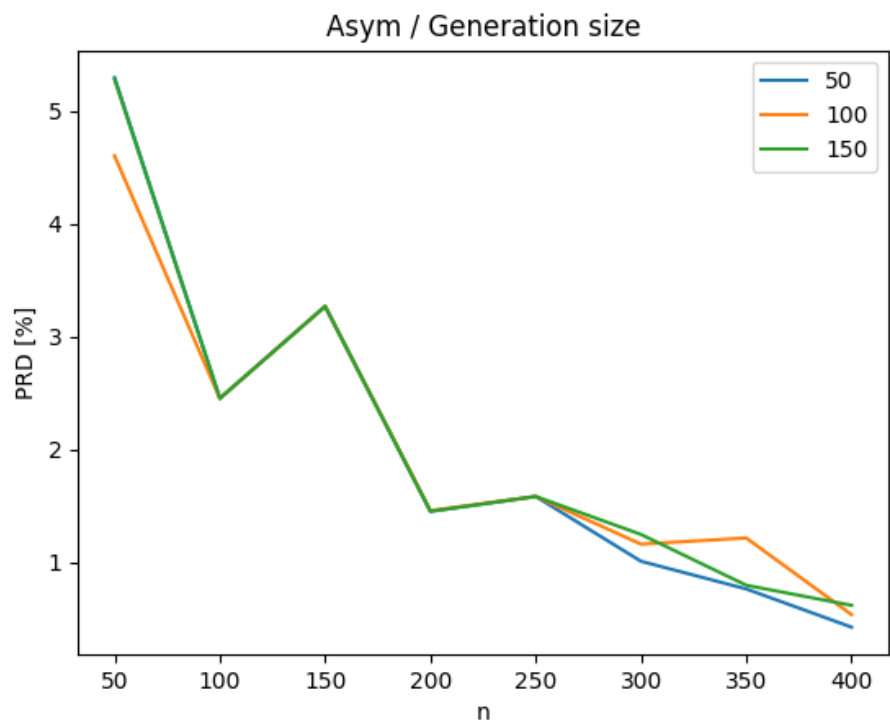


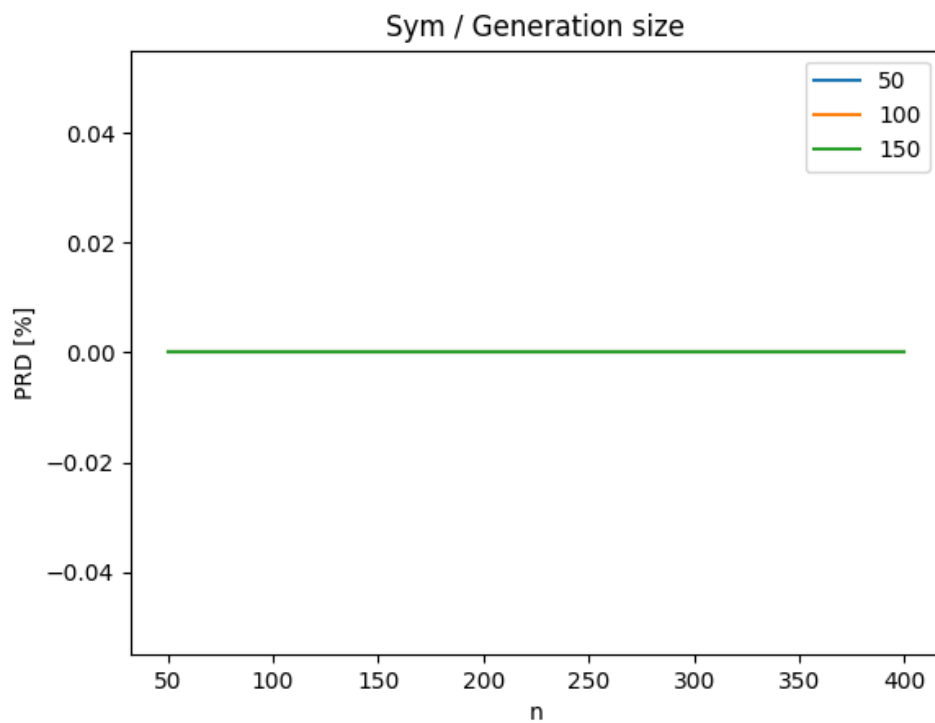




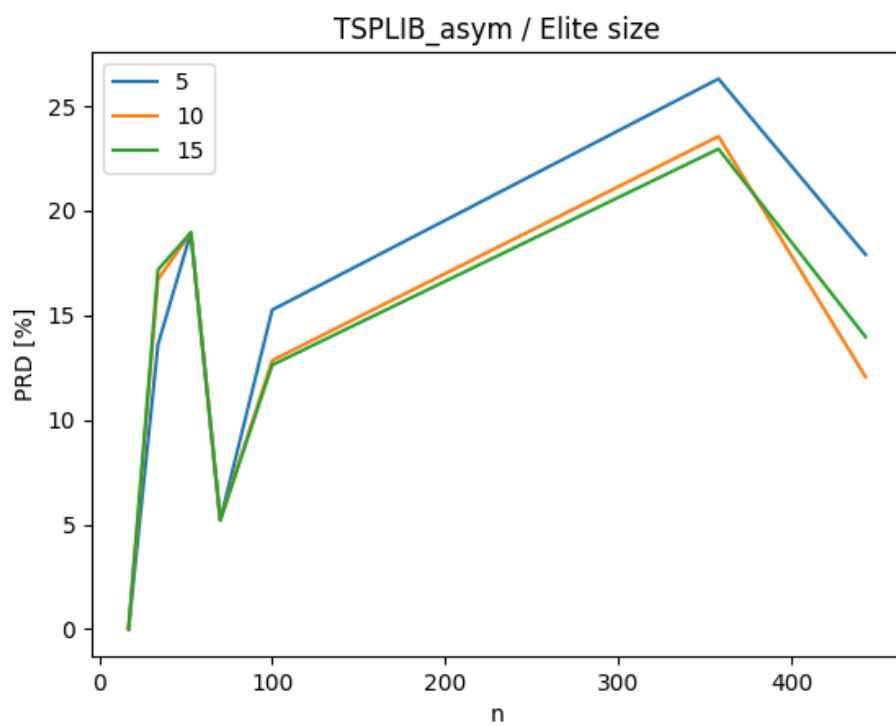
4.3 Wielkość pokolenia

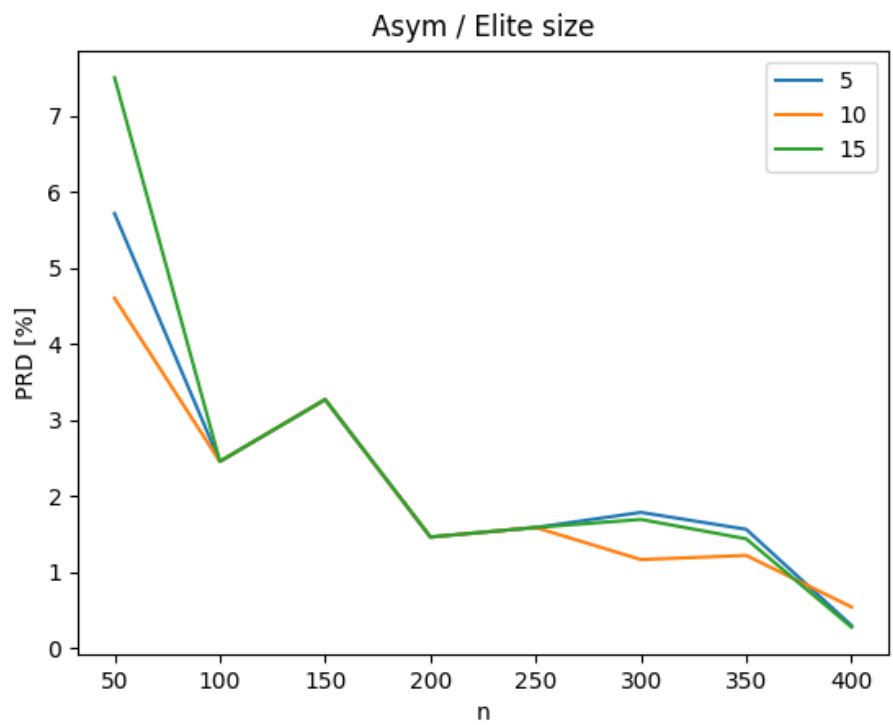
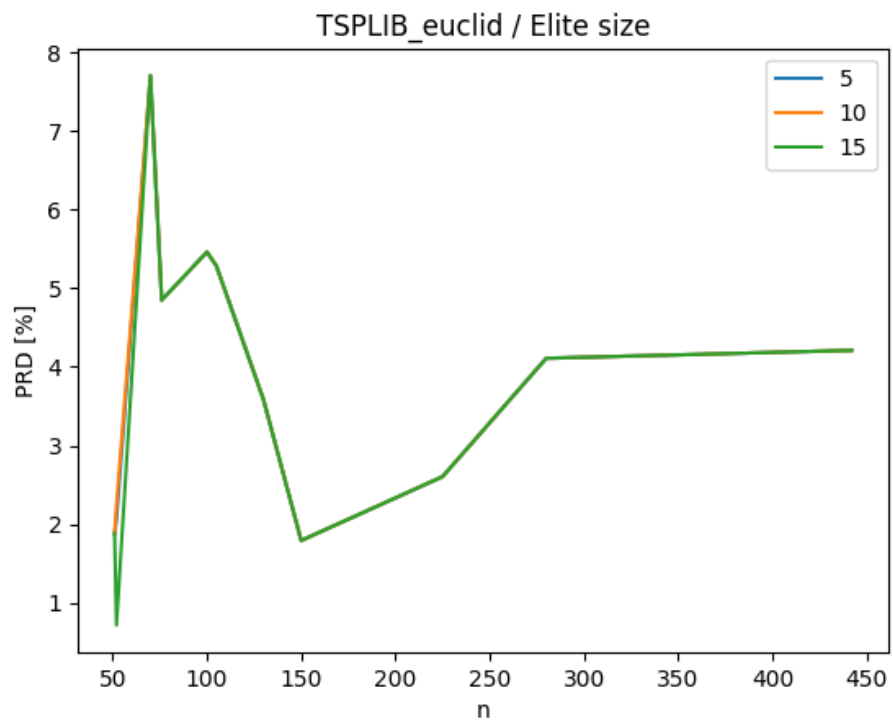


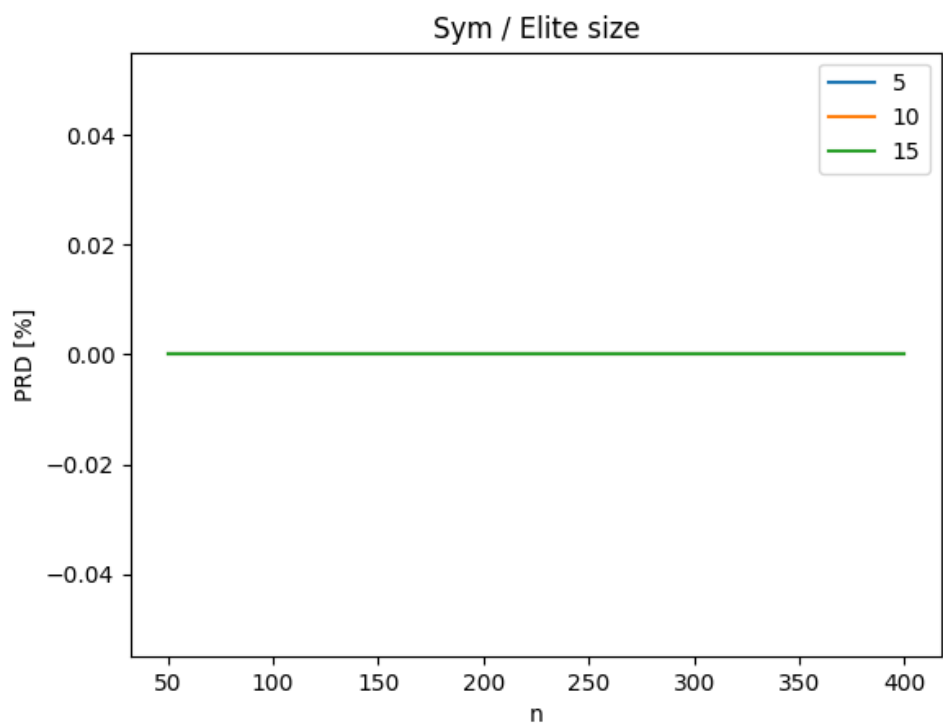
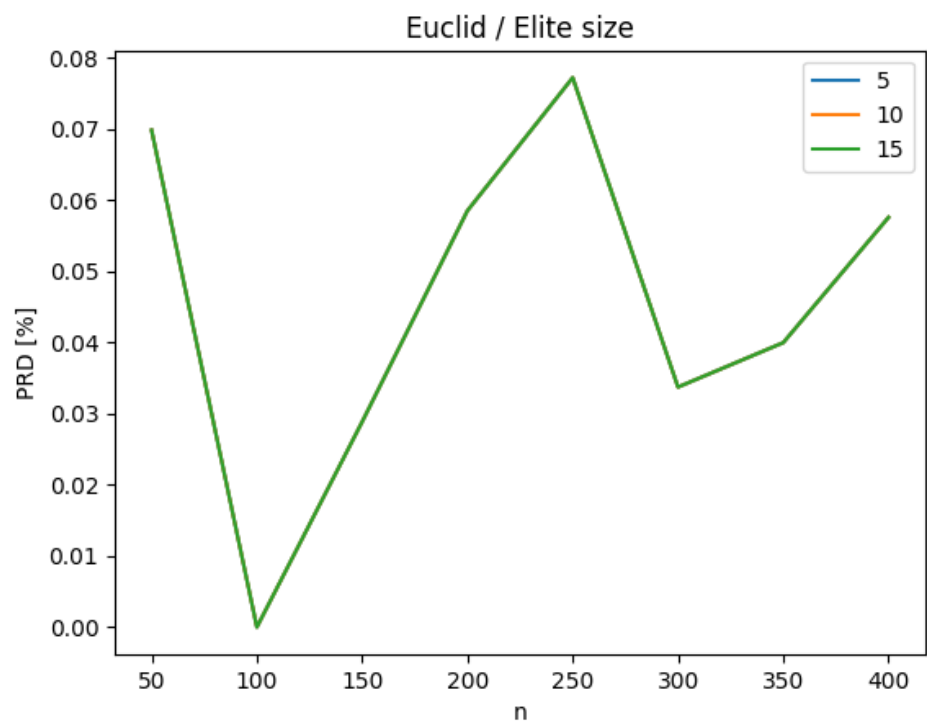




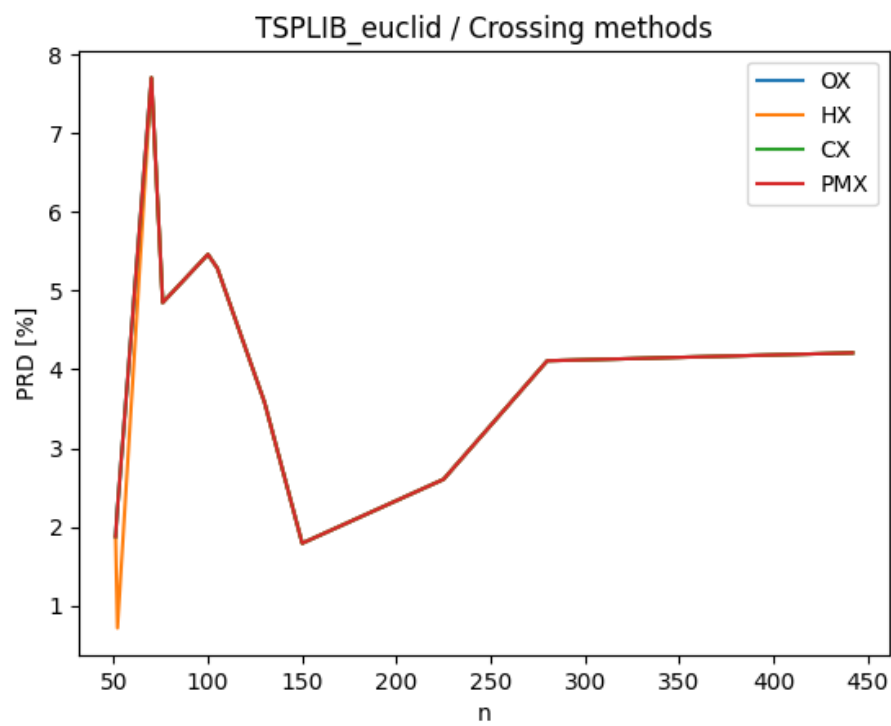
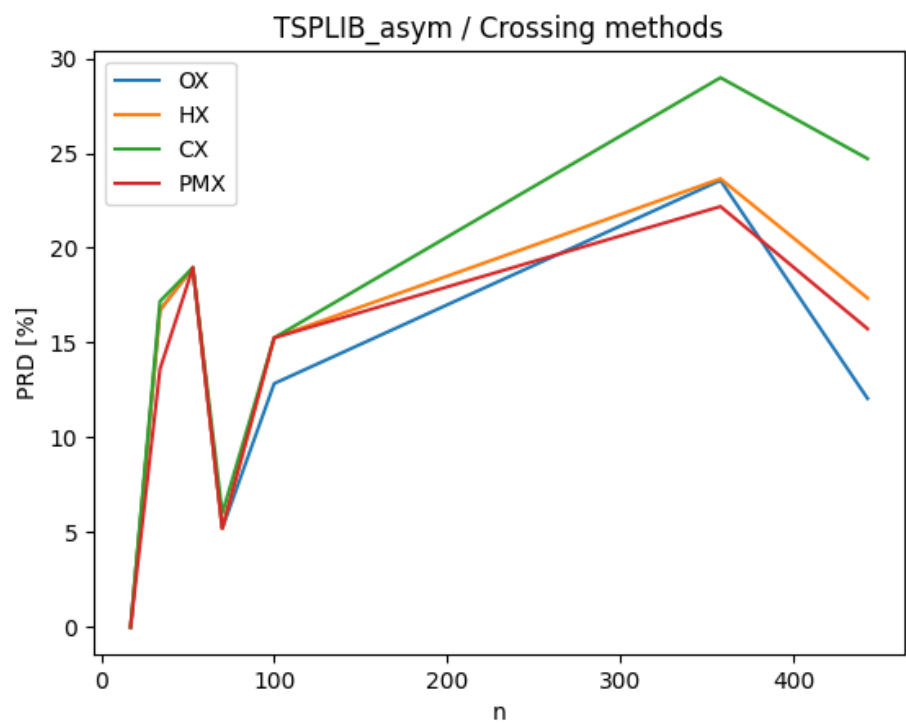
4.4 Wielkość elity



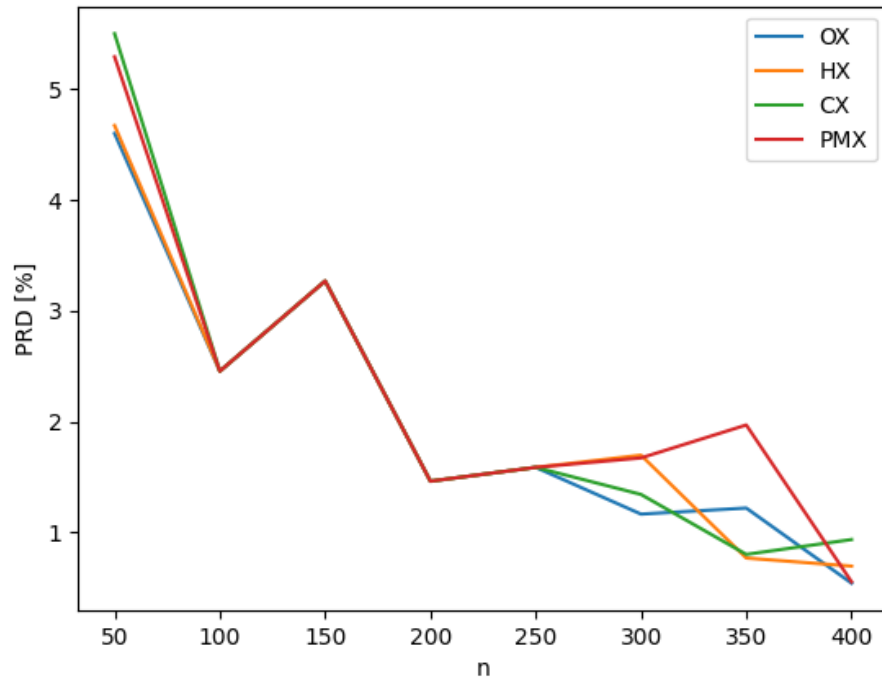




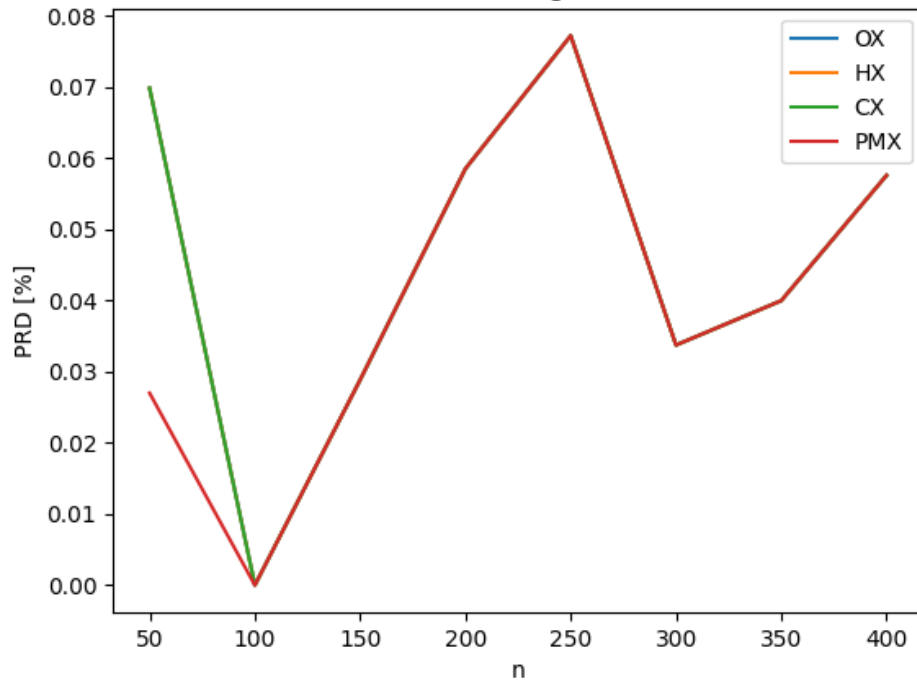
4.5 Krzyżowanie

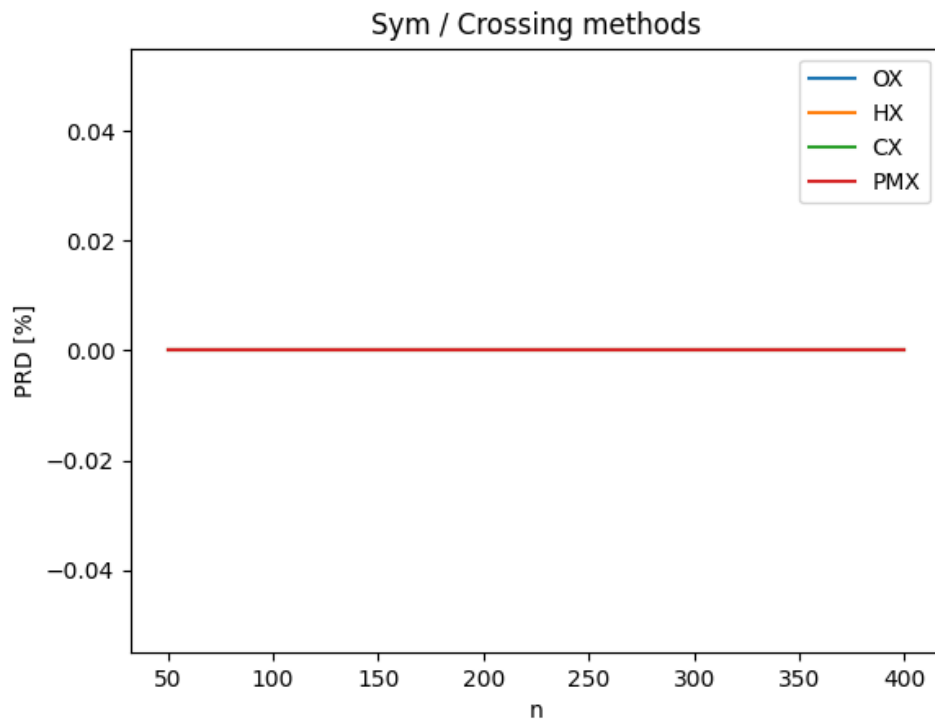


Asym / Crossing methods

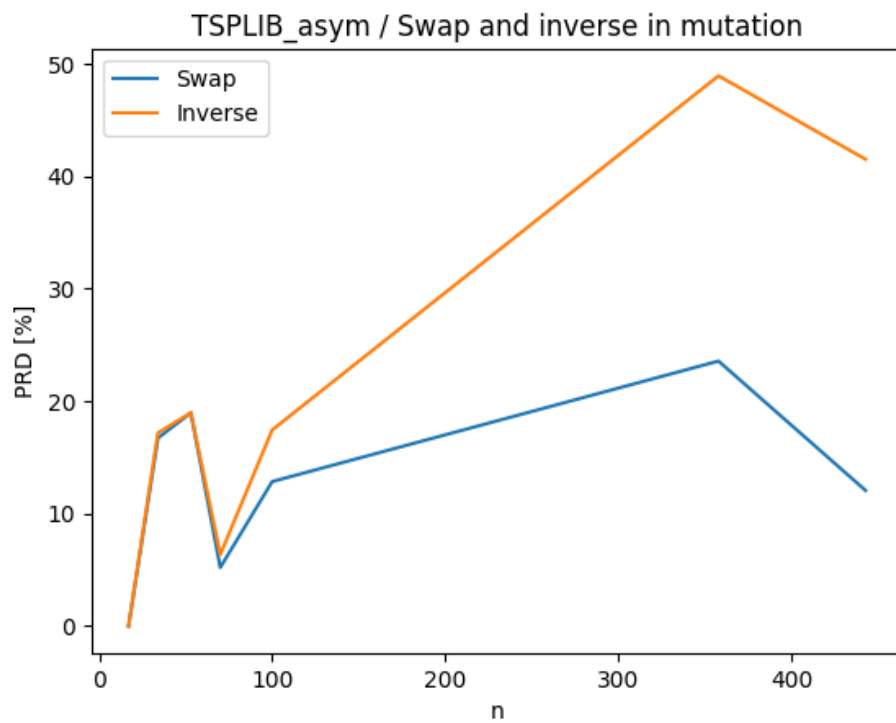


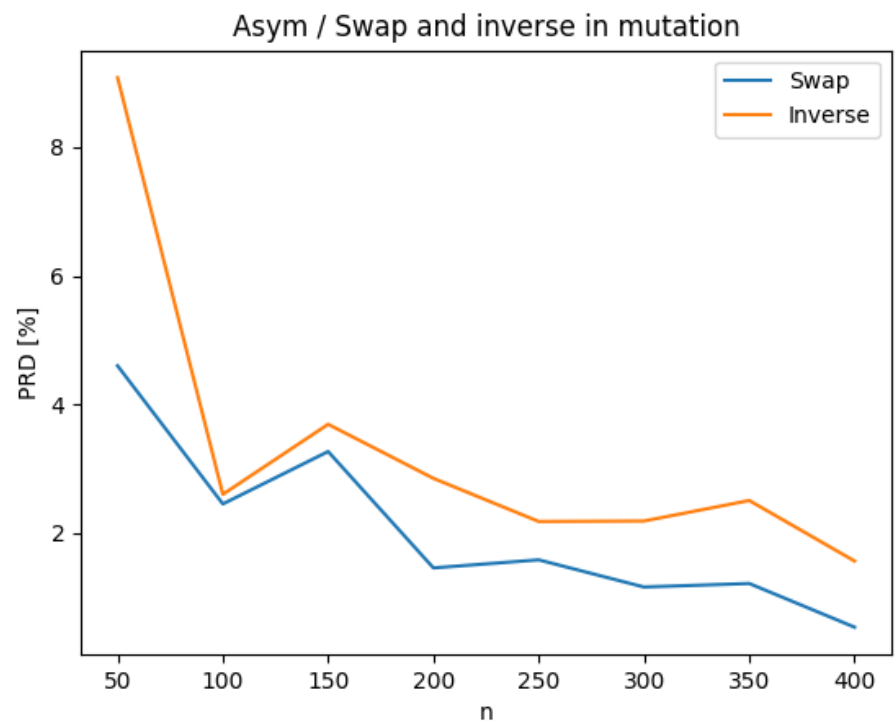
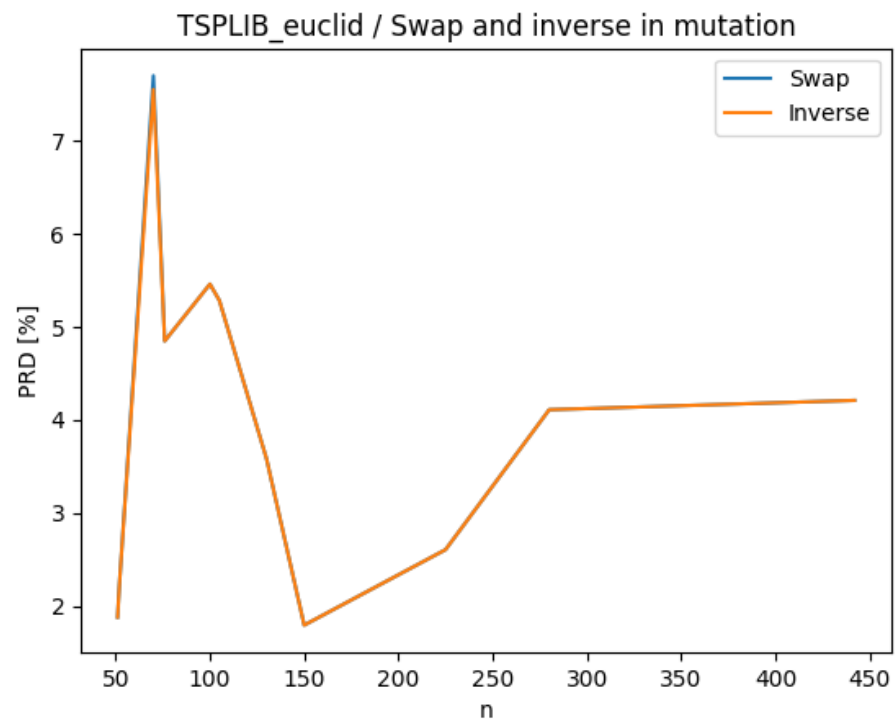
Euclid / Crossing methods

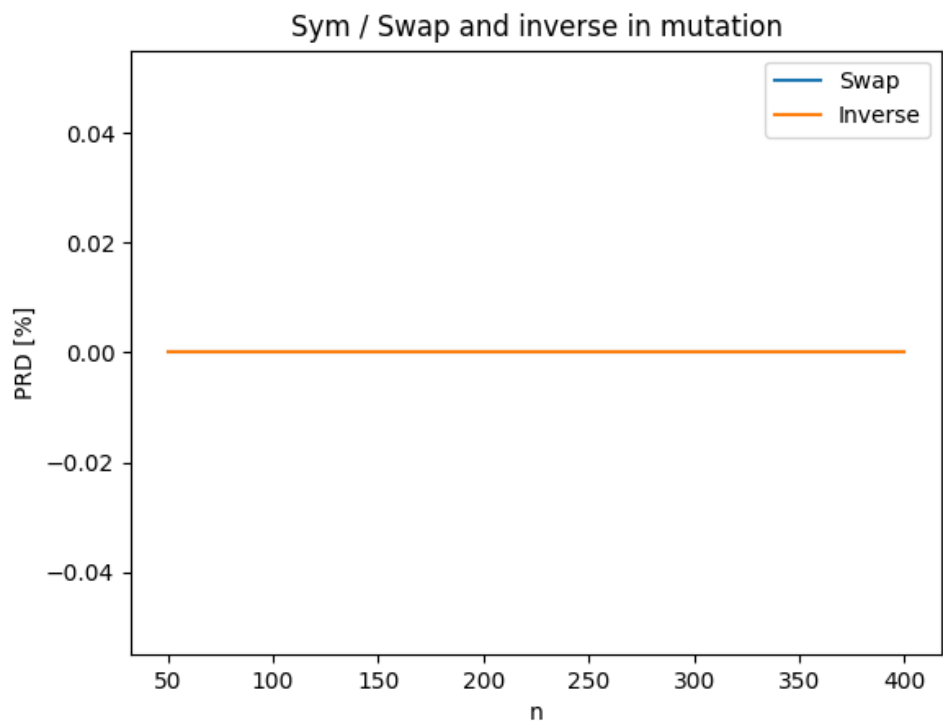




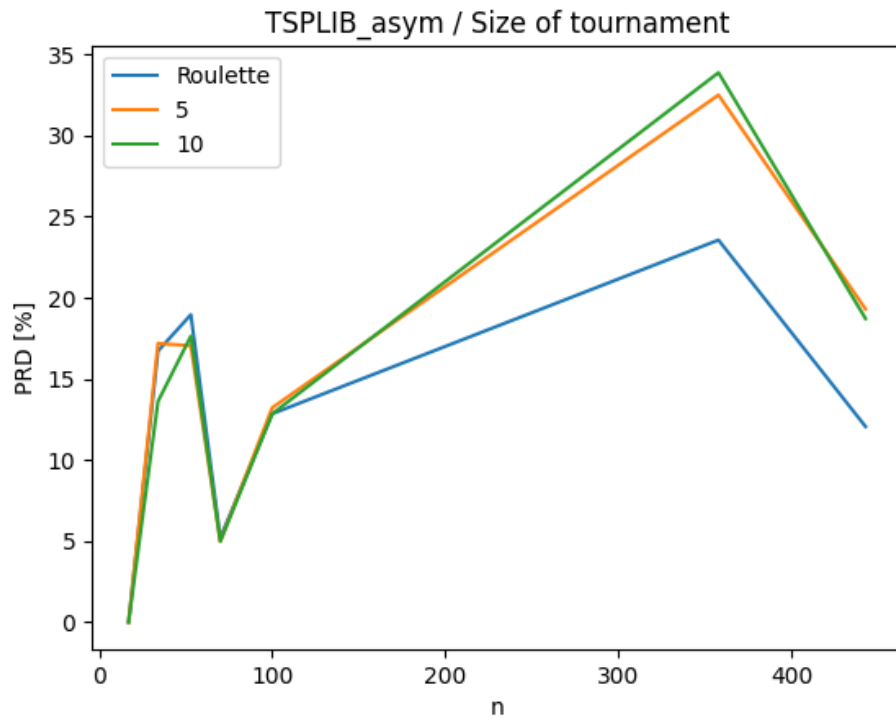
4.6 Swap i inverse



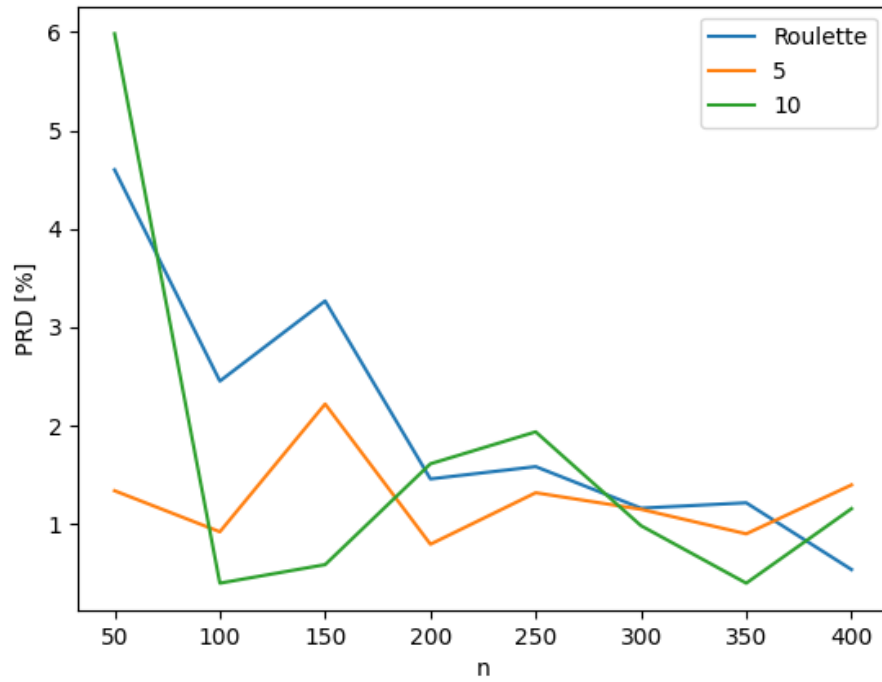




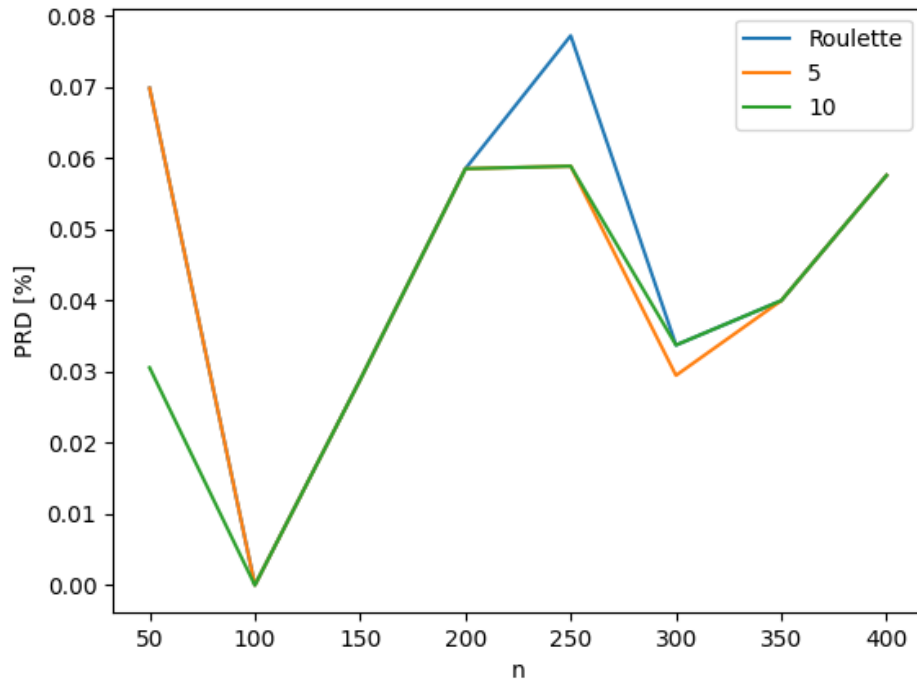
4.7 Turniej

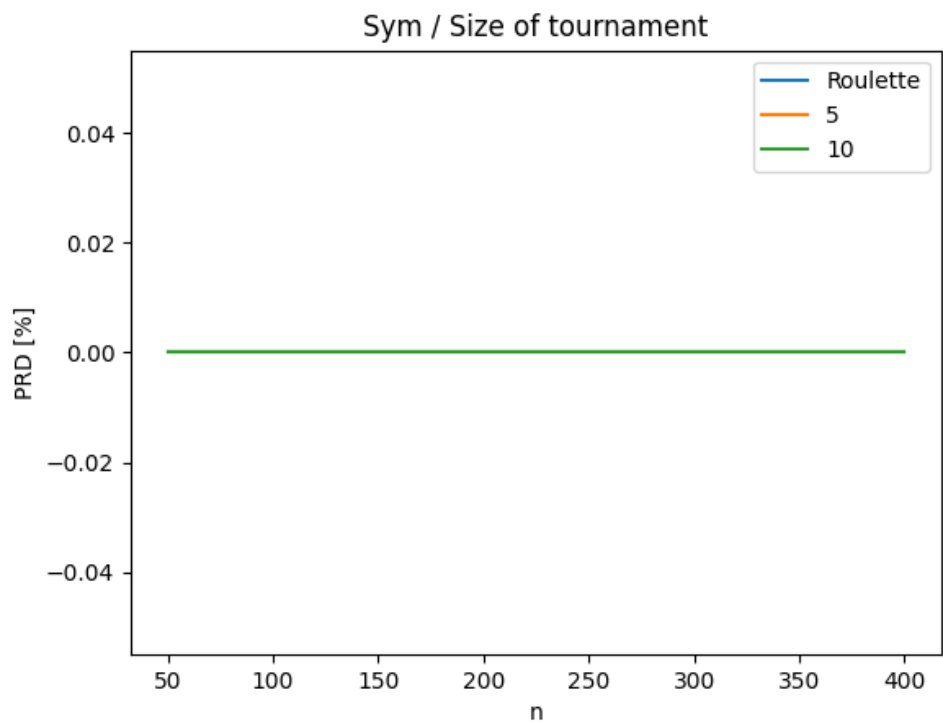


Asym / Size of tournament

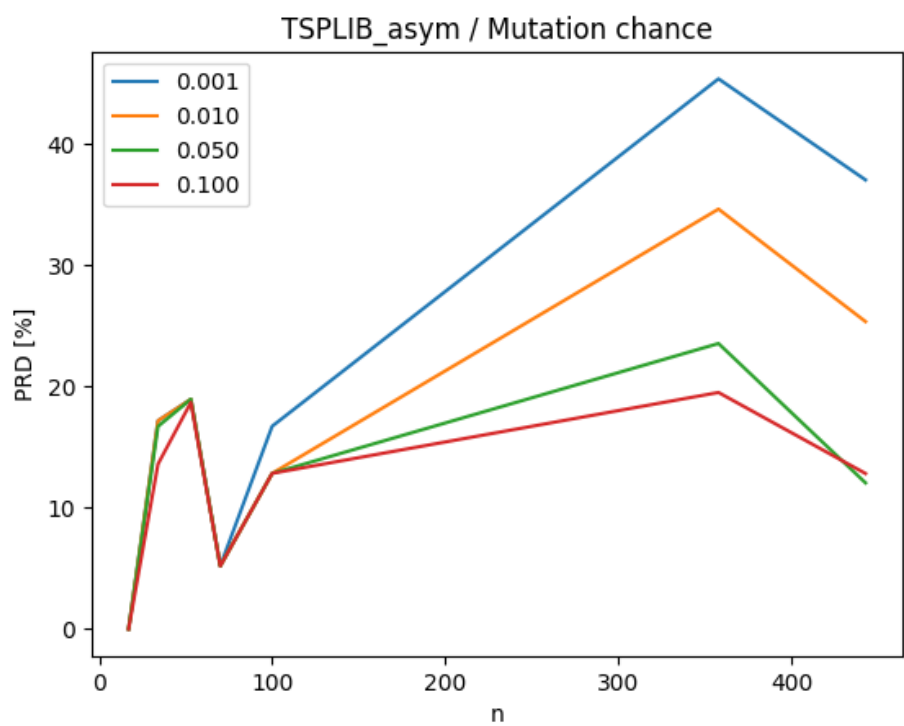


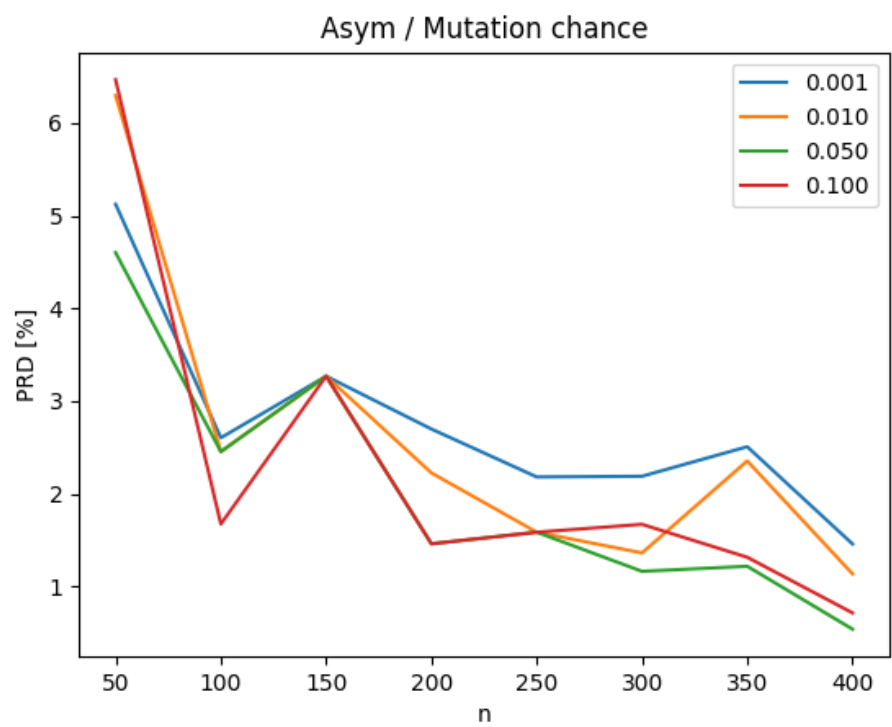
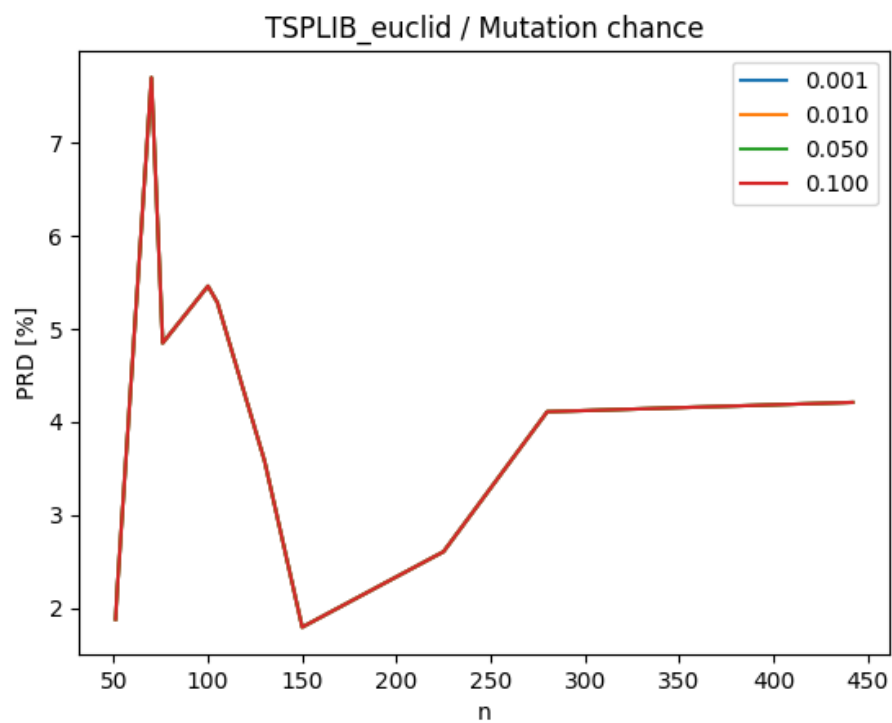
Euclid / Size of tournament

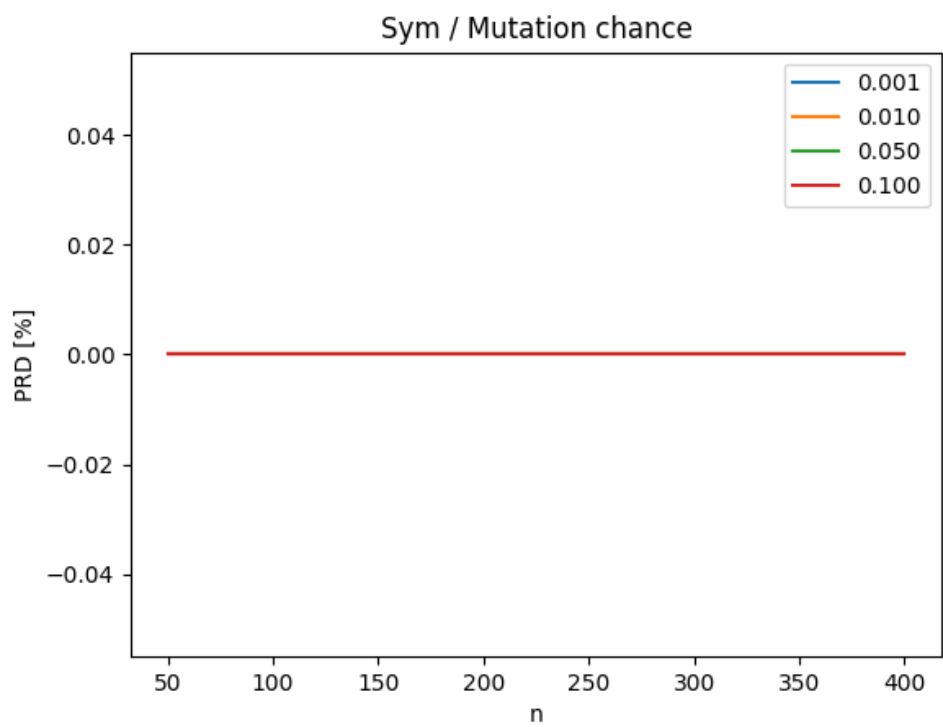
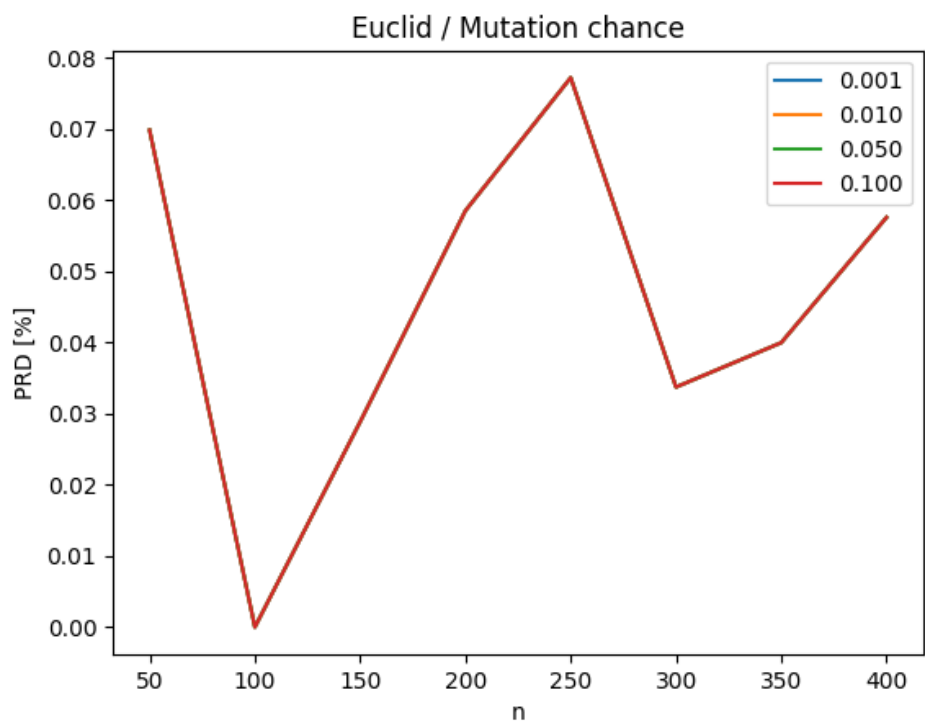




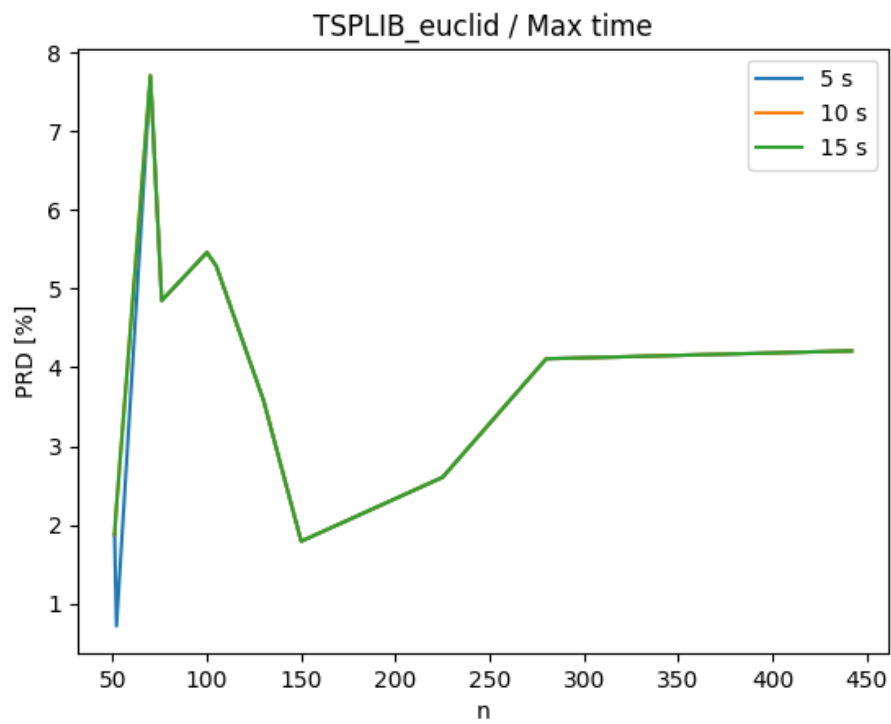
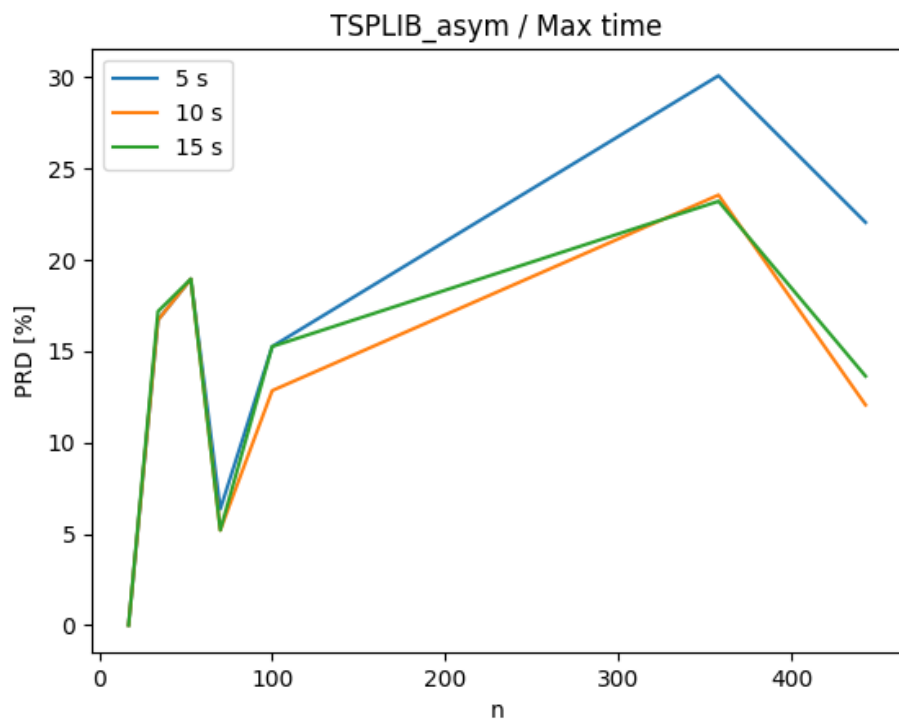
4.8 Szansa mutacji

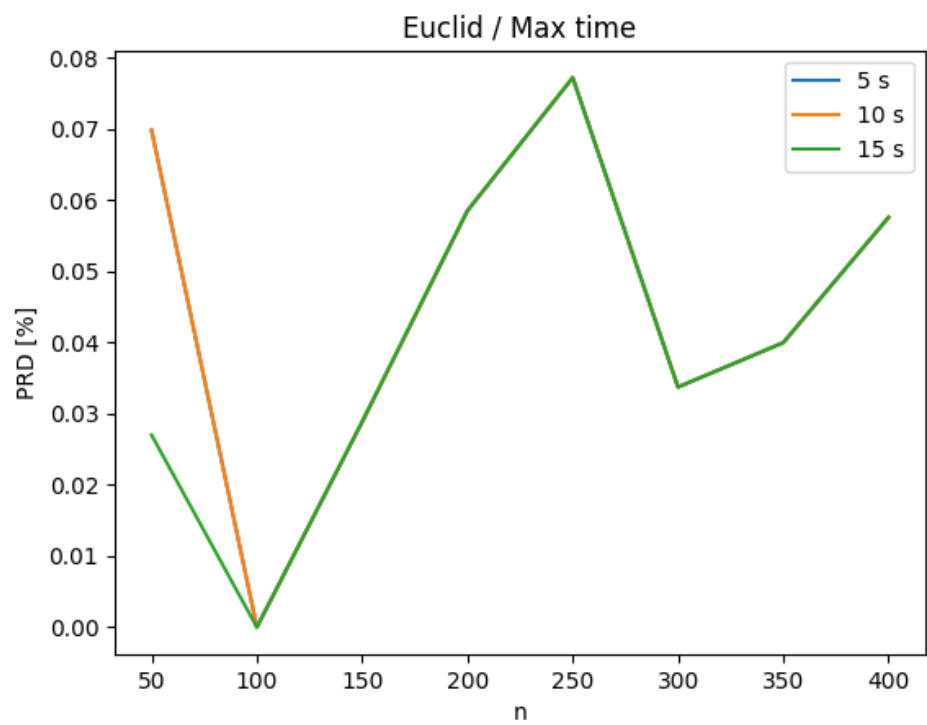
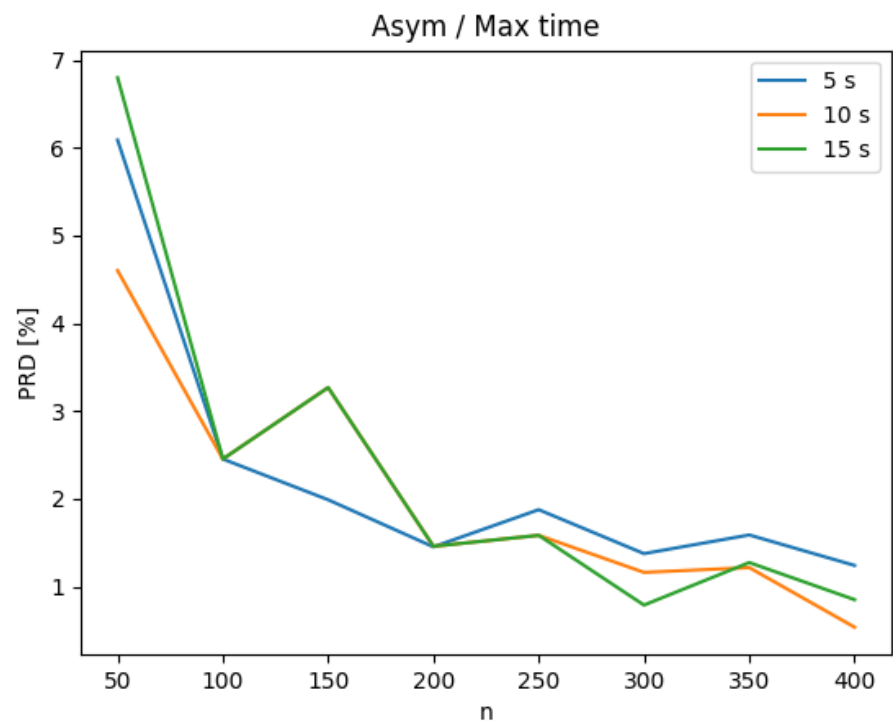


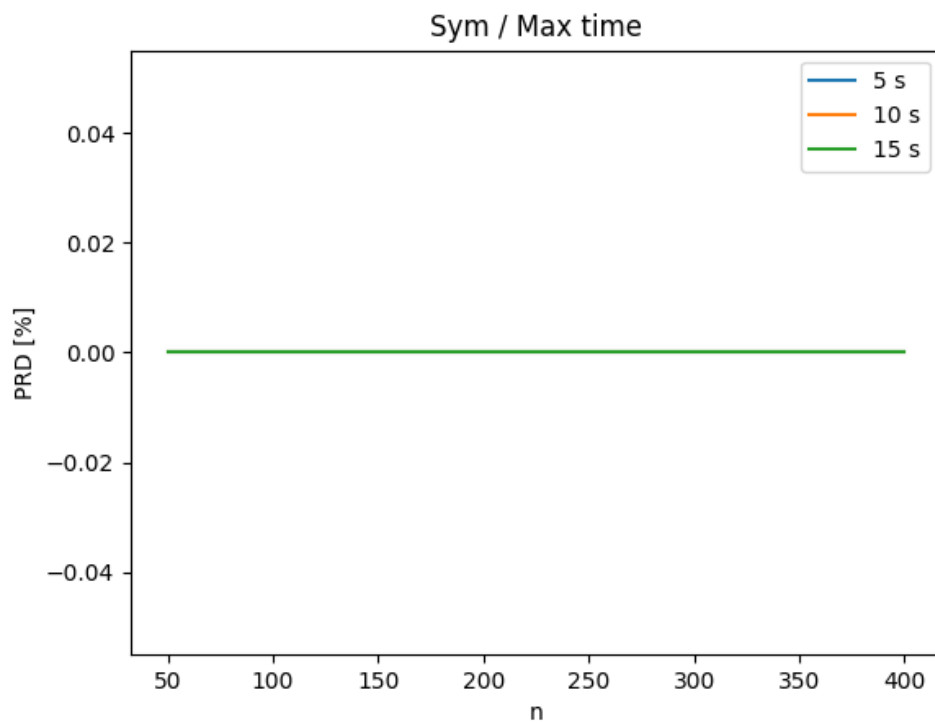




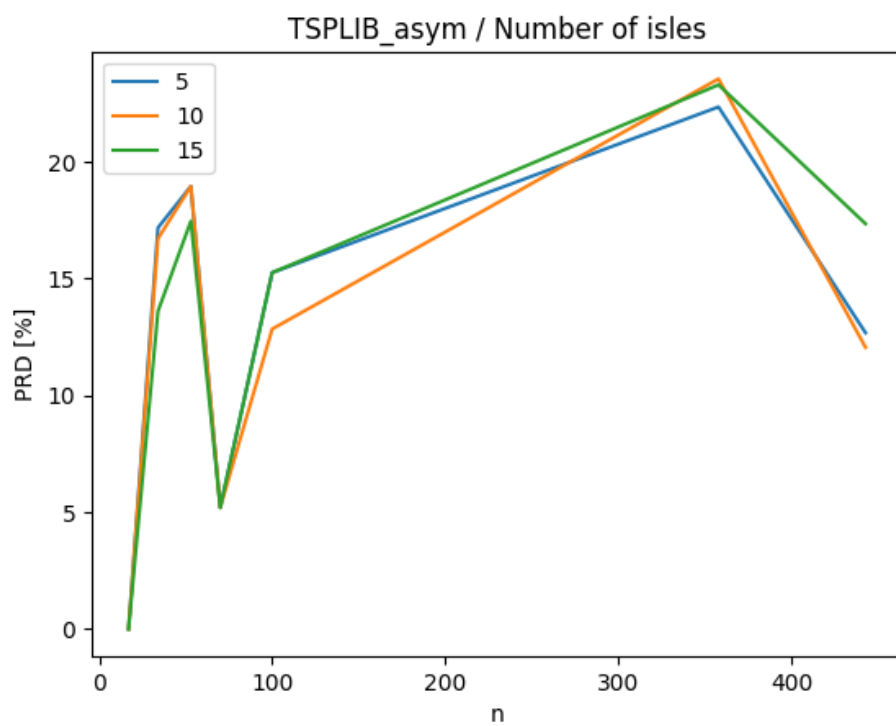
4.9 Maksymalny czas

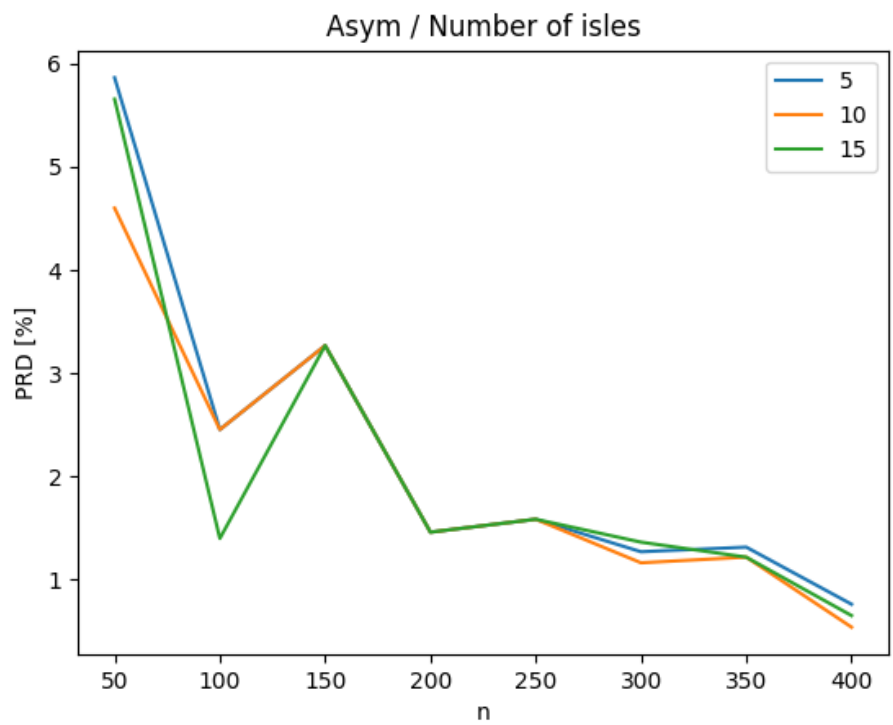
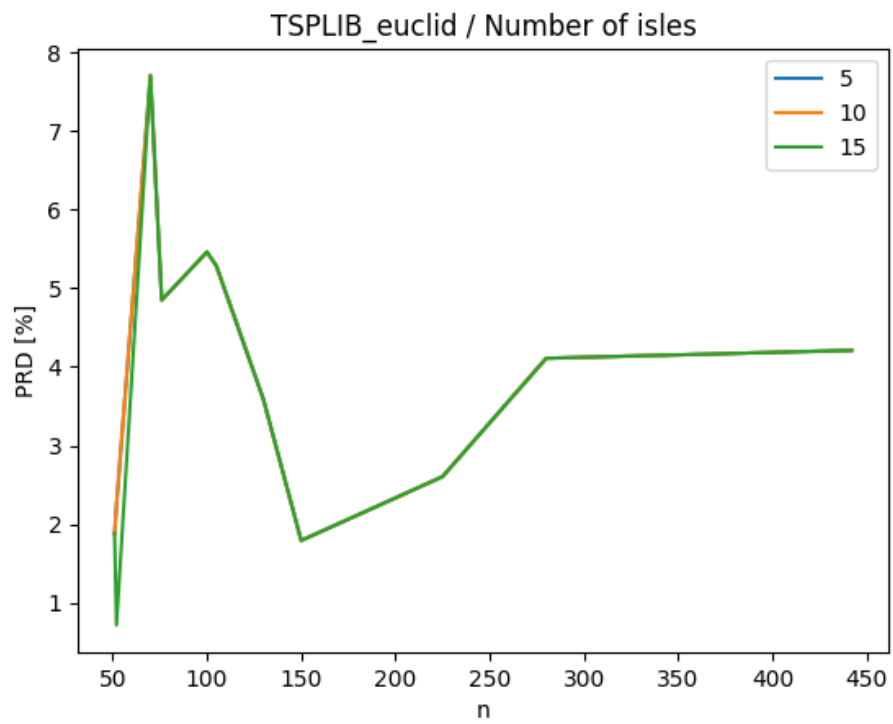


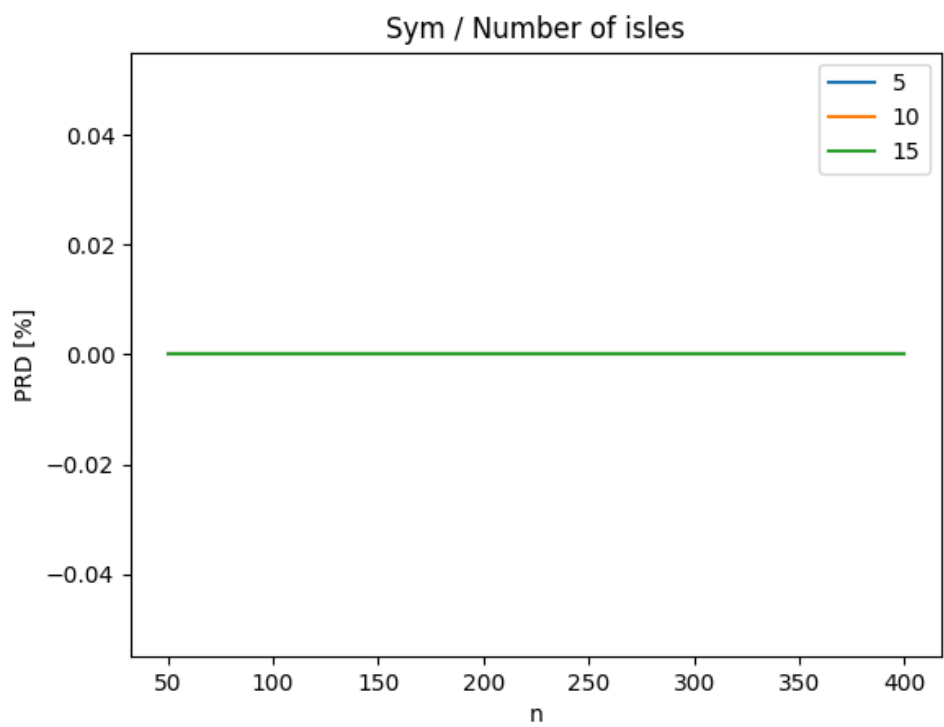
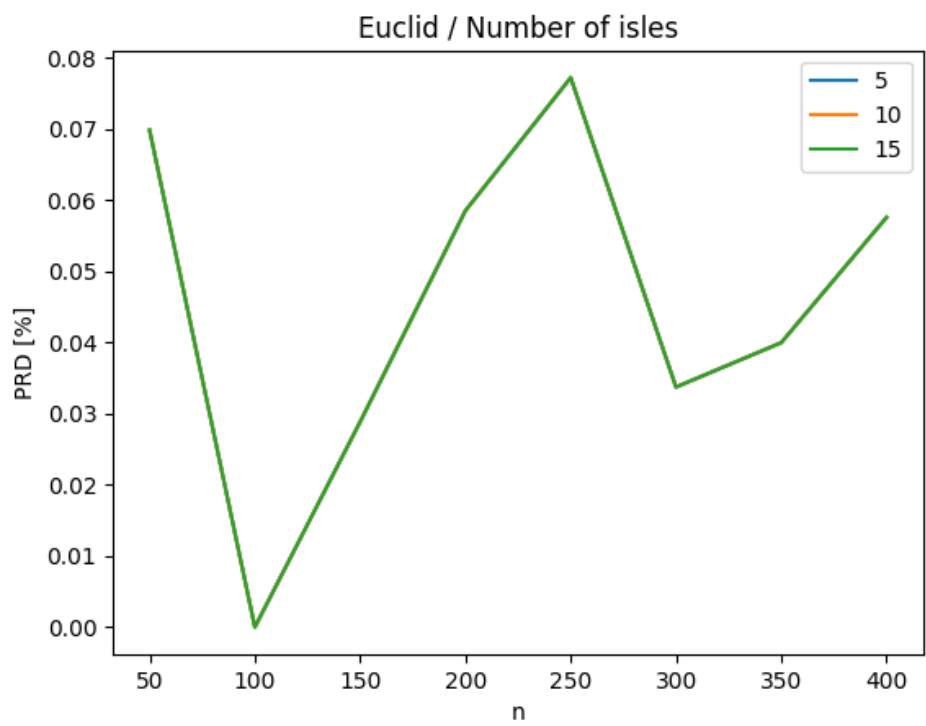




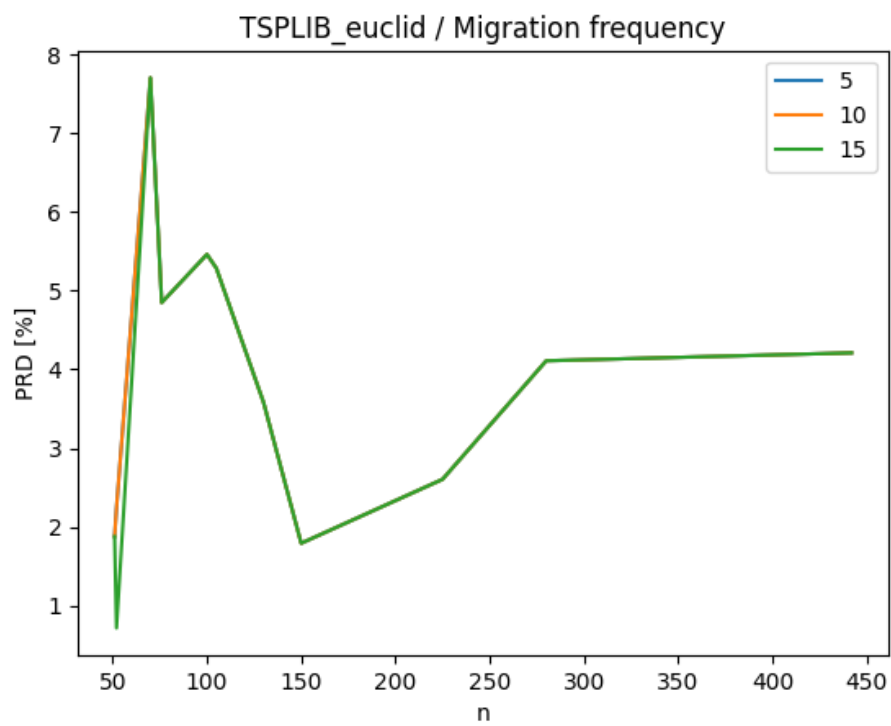
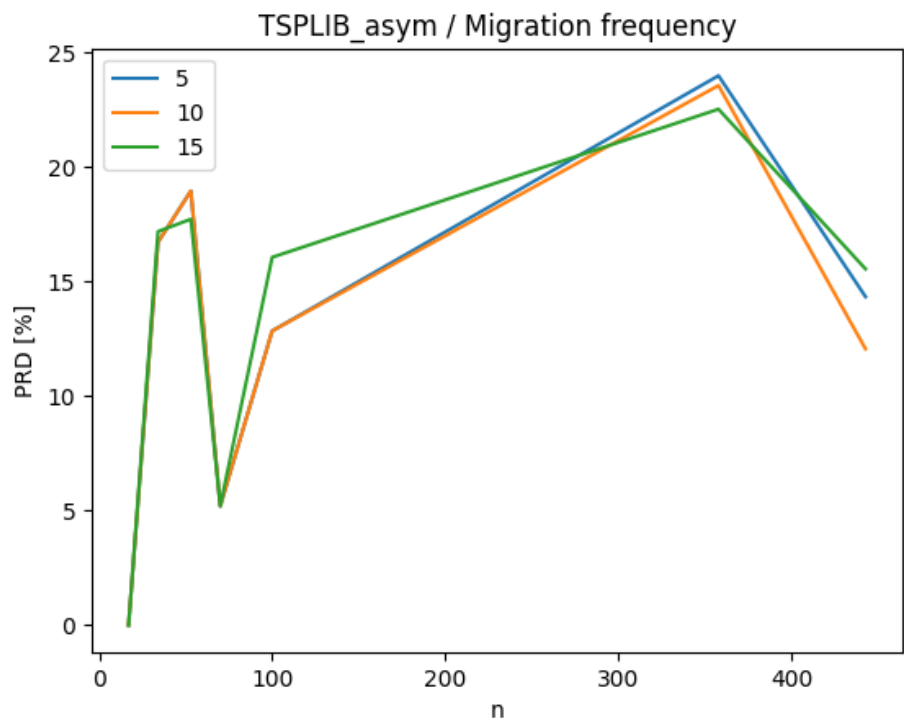
4.10 Liczba wysp

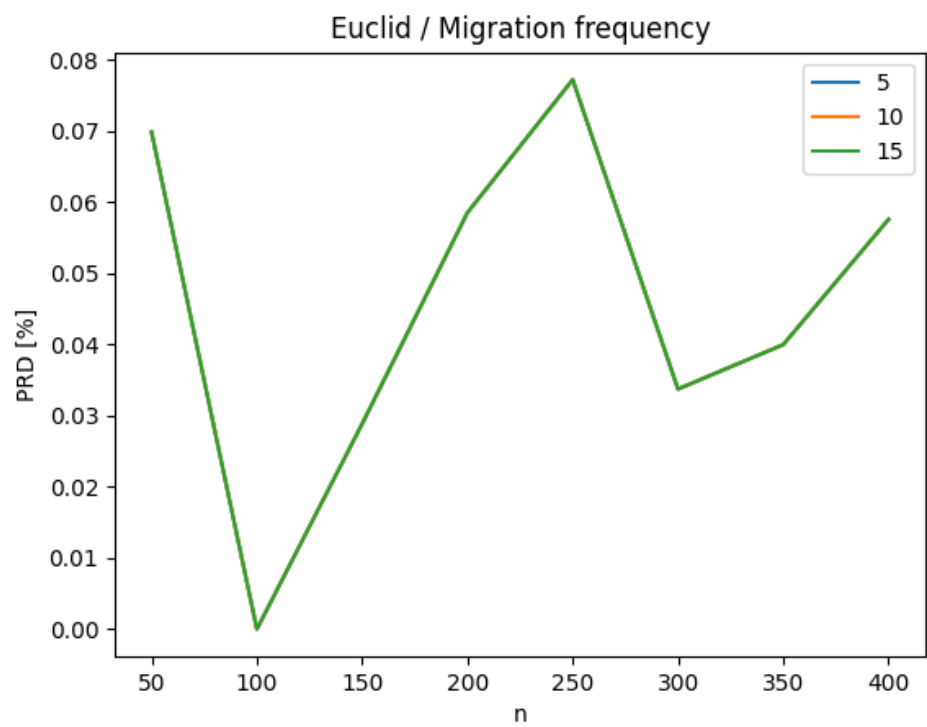
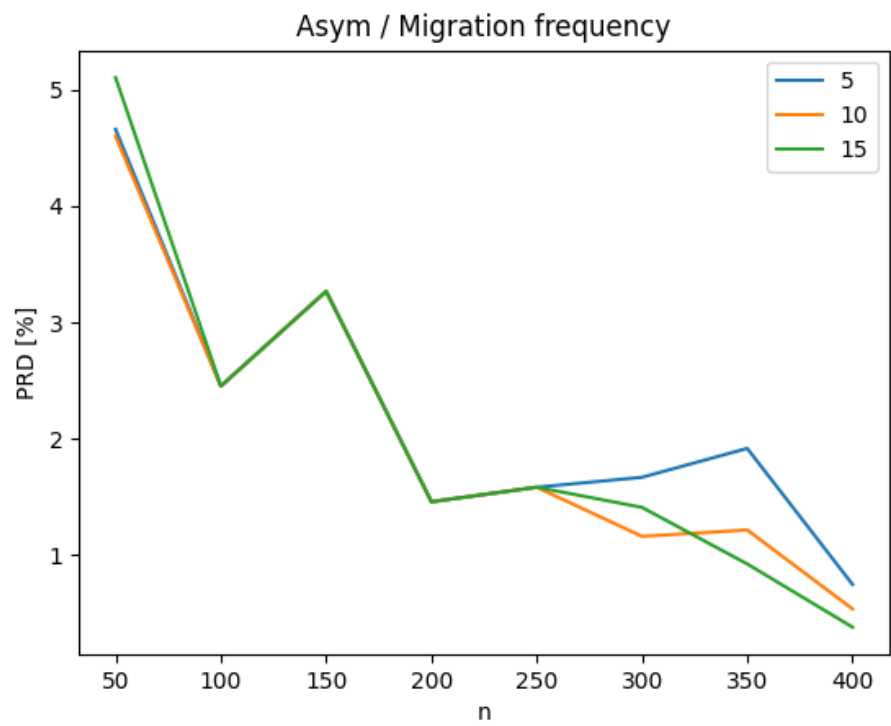


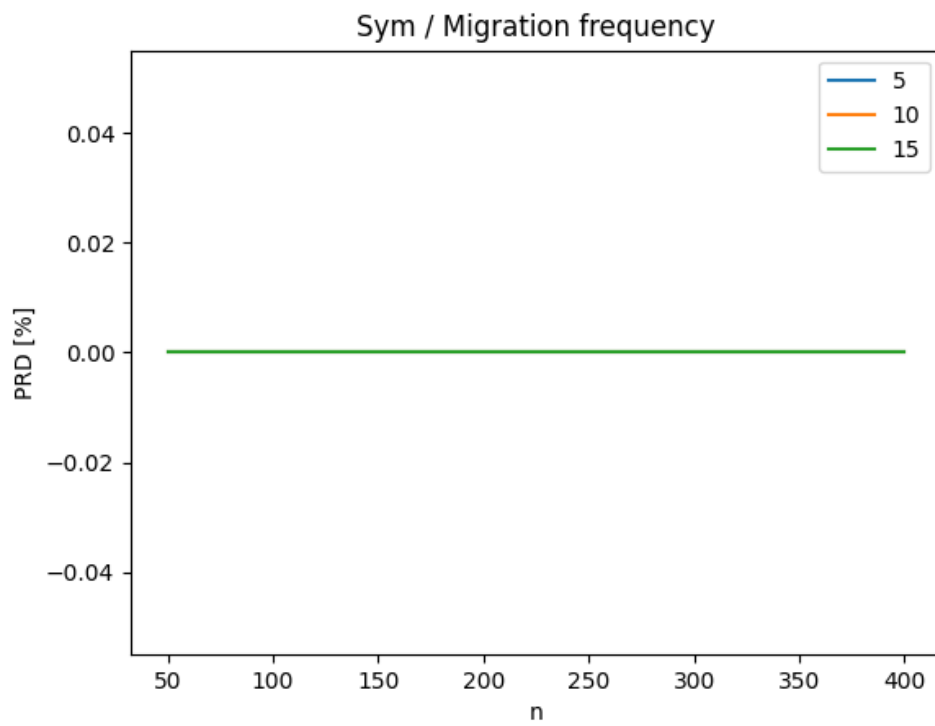




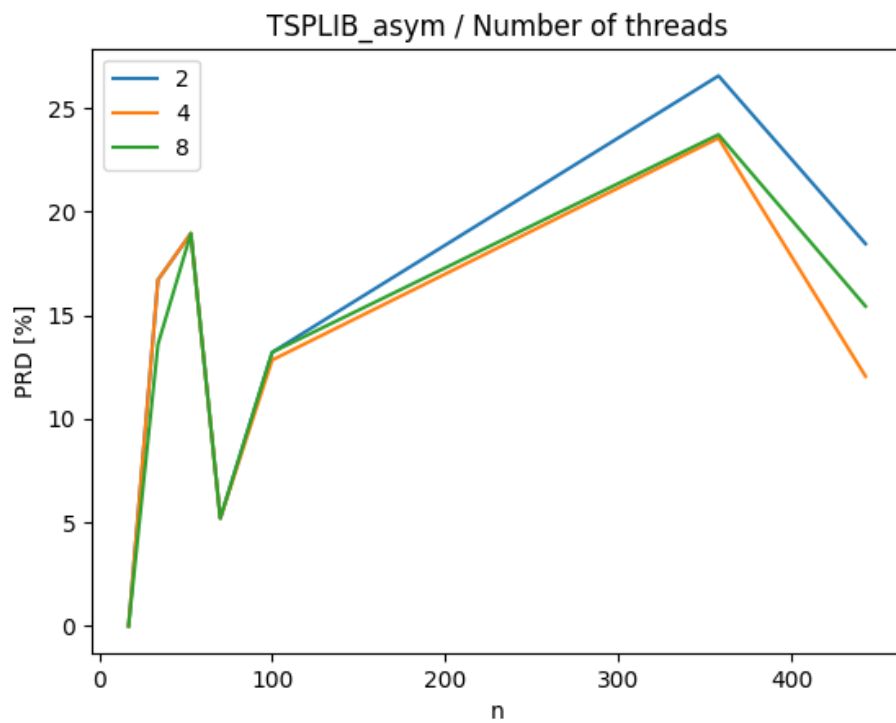
4.11 Częstotliwość migracji

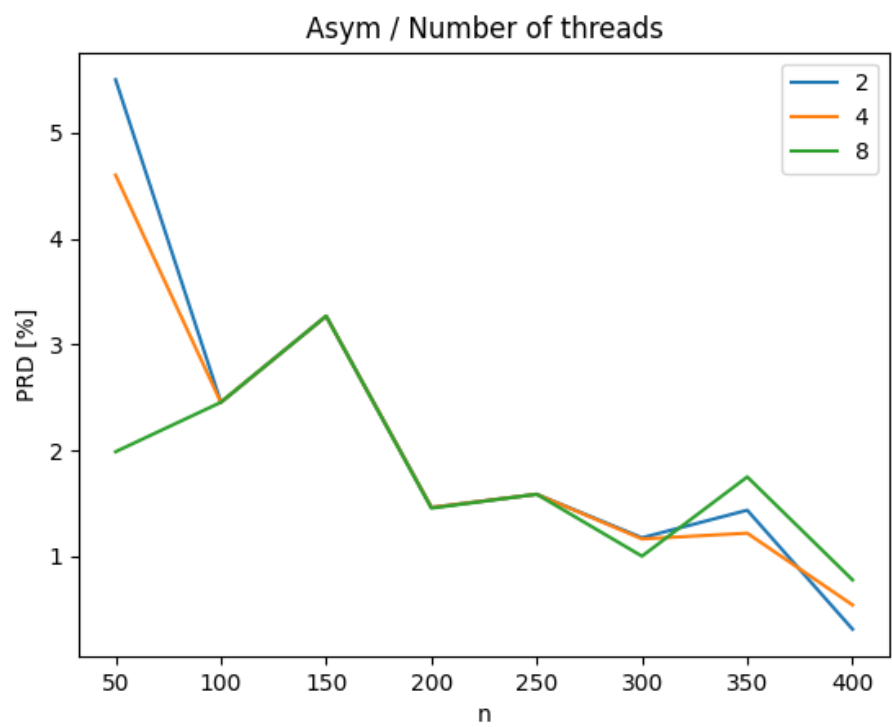
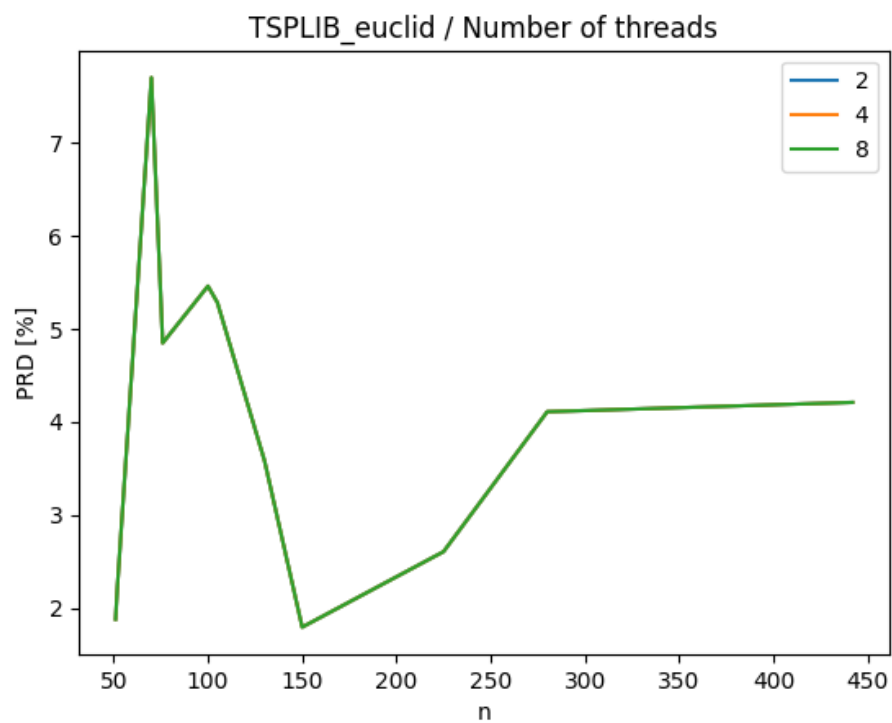


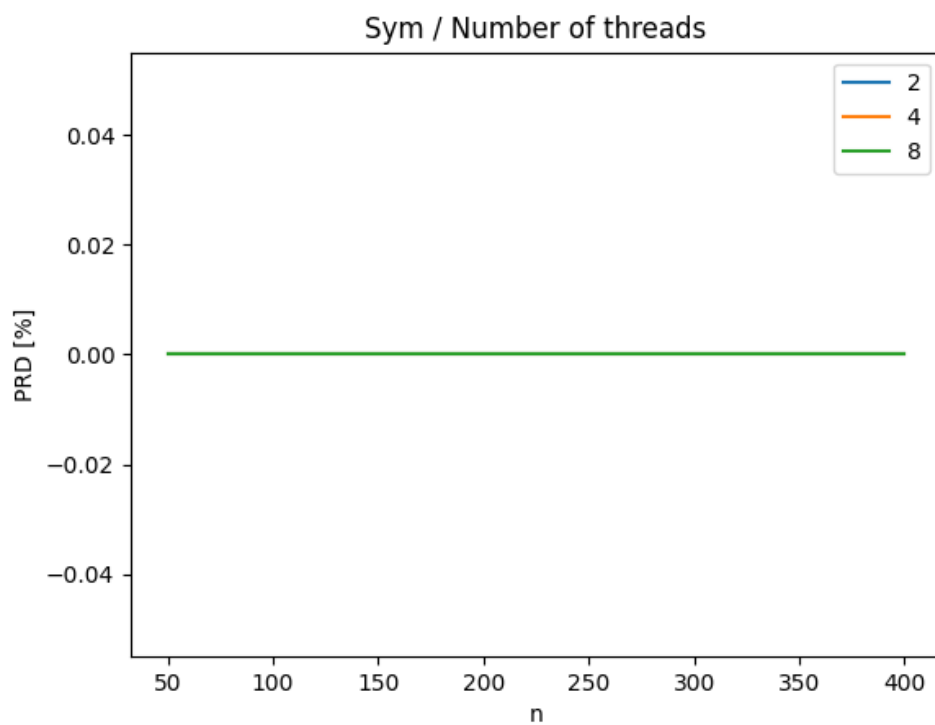
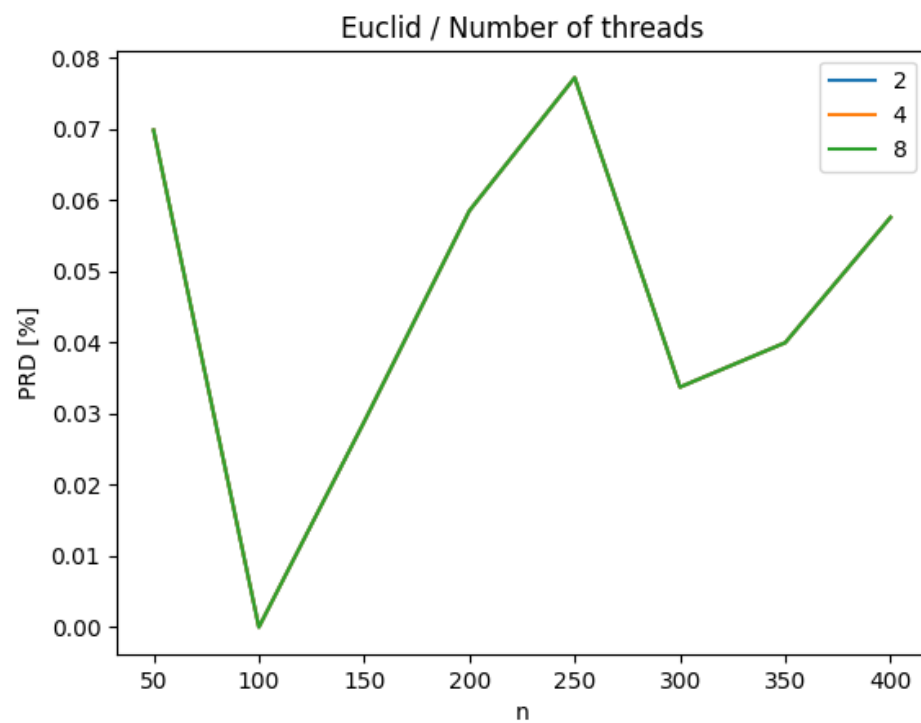




4.12 Liczba wątków







5 Obserwacje i wnioski

- Obserwacje i wnioski - do zrobienia