

Olga Waszczuk, Małgorzata Łukomska.

Projekt 1: Pivot Rules

1) Dokumentacja:

Rozwiązania do projektu znajdują się pod linkiem:

<https://github.com/olgawaszczuk/optimalizacja/tree/master/projekt1>

W pliku `methods(pivot rules).py` można znaleźć implementację następujących metod warunkujących wybór zmiennych wchodzących i wychodzących z programu liniowego (przy czym zachowaliśmy konwencję "nazwa_funkcji_entering" wybiera zmienną wchodzącą do programu, zaś "nazwa_funkcji_leaving" wybiera zmienną wychodzącą, bazując na zadanej regule) :

1) `largest_coefficient`:

Funkcja wybiera tę zmienną, przy której stoi największy współczynnik w funkcji celu.

2) `maximal_objective`:

Funkcja wybiera zmienną na podstawie wzorstu funkcji celu. Oznacza to, że funkcje `maximal_objective_entering` i `maximal_objective_leaving` zwracają dwie zmienne: wchodzącą i wychodzącą, tak, aby ich zamiana w programie dawała najwyższy wzrost funkcji celu spośród wszystkich możliwych par zmiennych wchodzących i wychodzących. Należy zauważyć, że zmienna wychodząca dobierana jest już do zmiennej wchodzącej.

3) `bland_rule`:

Funkcja `bland_rule_entering` zwraca zmienną wchodzącą o najniższym indeksie, zaś funkcja `bland_rule_leaving` dobiera do tej zmiennej możliwą zmienną wychodzącą, również na podstawie tego, aby jej index był najmniejszy w zbiorze możliwych zmiennych wychodzących.

4) `random_edge`:

Funkcja wybiera z prawdopodobieństwem jednostajnym losową zmienną, ze zmiennych, które mogą wejść lub wyjść z programu.

5) `smallest_coefficient`:

Funkcja zwraca zmienną, przy której w funkcji celu stoi najmniejszy współczynnik.

6) `max_bounds_difference`:

Porównujemy wartość bezwzględną różnicy współczynnika stojącego przy zmiennej w pierwszej funkcji ograniczającej, i sumy współczynników stojących przy zmiennej w pozostałych funkcjach ograniczających. Funkcja zwraca tę zmienną, dla której różnica ta jest największa.

7) `lexicographical_min`:

Funkcja zwraca najmniejszą zmienną wśród możliwych zmiennych wchodzących i wychodzących do programu, gdzie ułożenie zmiennych podlega pod porządek leksykograficzny.

8) `lexicographical_max`:

Funkcja zwraca największą zmienną wśród możliwych zmiennych wchodzących i wychodzących do programu, gdzie ułożenie zmiennych podlega pod porządek leksykograficzny.

W folderze "problems" znaleźć można 10 wybranych problemów liniowych, dla których przeprowadziłyśmy kolejno wybór zmiennych zgodnie z każdą z wymienionych powyżej metod.

W folderze "testy" zaprezentowane są wyniki wspomnianych testów i krótkie wnioski na temat metod wyboru zmiennych.

2) Wyniki testów:

W folderze „testy” dostępne są wykresy zestawiające liczby kroków wykonywanych przez poszczególne algorytmy przy rozwiązywaniu każdego z programów. W raporcie przedstawione zostaną wybrane z nich.

Wyniki testów przeprowadzonych kolejno dla każdej metody wyboru zmiennych jak i dla każdego problemu liniowego przedstawione są w tabeli "tabela_wyników". Tabela pokazuje ile kroków wykonał algorytm działający zgodnie z wybraną zasadą, przy rozwiązywaniu konkretnego przykładowego problemu.

Tabela jest dostępna poniżej:

		problem name:									
method:		routes	paint	transportation	furniture	Whiskas 1	whiskas2	Hetmani 5	Hetmani 3	exercise	profit
	largest coefficient	5	2	2	2	2	2	16	9	2	5
	maximal objective value	8	2	2	2	2	7	inf	15	2	5
	bland rule	7	3	2	2	2	11	29	10	3	4
	random edge	6	3	3	2	2	8	14	10	3	6
	smallest coefficient	6	3	3	2	2	8	20	11	3	7
	max_bounds	7	2	2	2	2	10	31	10	2	3
	lexicographical_min	7	3	2	2	2	11	29	10	3	4
	lexicographical_max	8	2	3	2	2	2	31	15	2	5

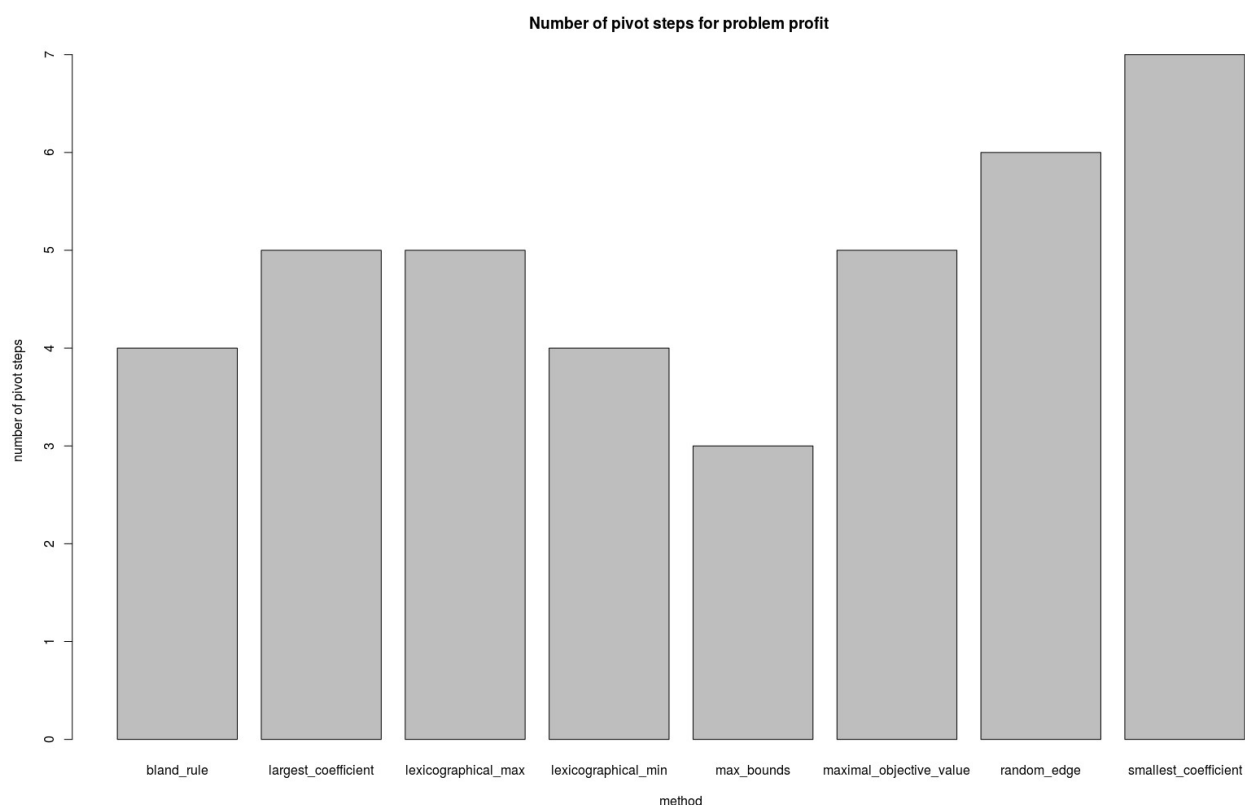
Widzimy, że problemy podzielić można na dwie kategorie: mało złożone i złożone.

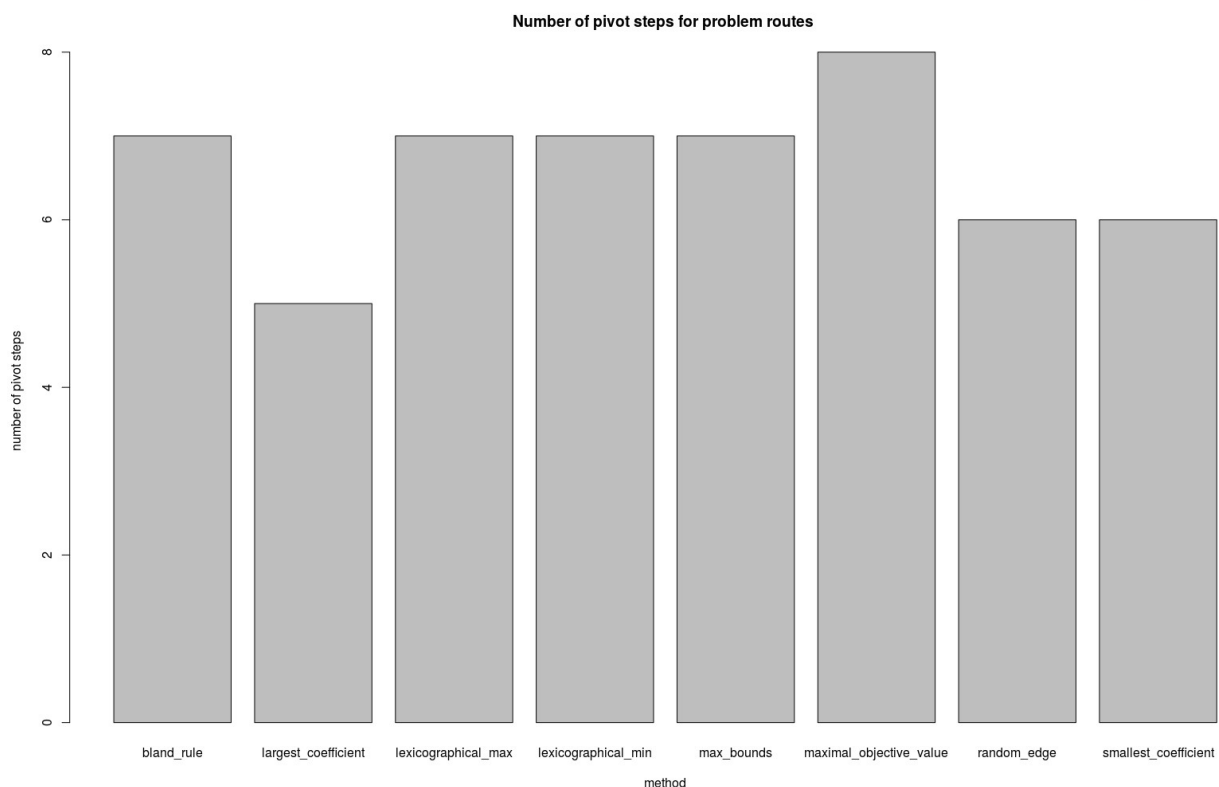
W przypadku mało złożonych problemów takich, jak np. paint i furniture wszystkie reguły wyboru zmiennych mają zbliżoną efektywność, czyli znajdują optymalne rozwiązanie w 2-3 krokach.

Na szczególną uwagę zasługują problemy, dla których algorytm musiał wykonać większą liczbę iteracji. W zależności od programu, różnice w efektywności metod były zasadnicze, lub znikome.

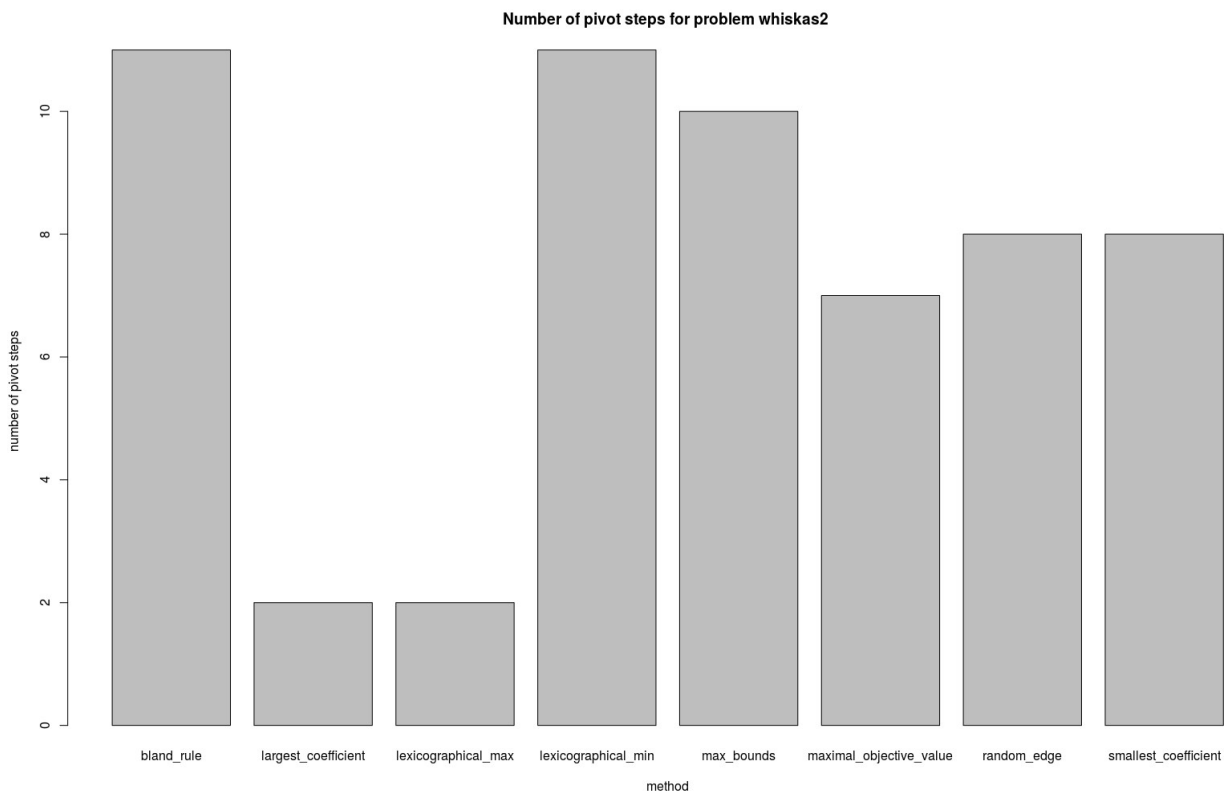
Gdy spojrzymy na wyniki dla problemów routes i profit możemy zauważyć, że każdy z algorytmów wykonał zbliżoną liczbę kroków, nie przekraczającą 8 kroków. Najlepiej poradziły sobie algorytmy largest_coefficient i max_bounds wykonując dla obu testów średnio 5 kroków.

Zestawienie liczby kroków dla obu metod znajduje się poniżej:



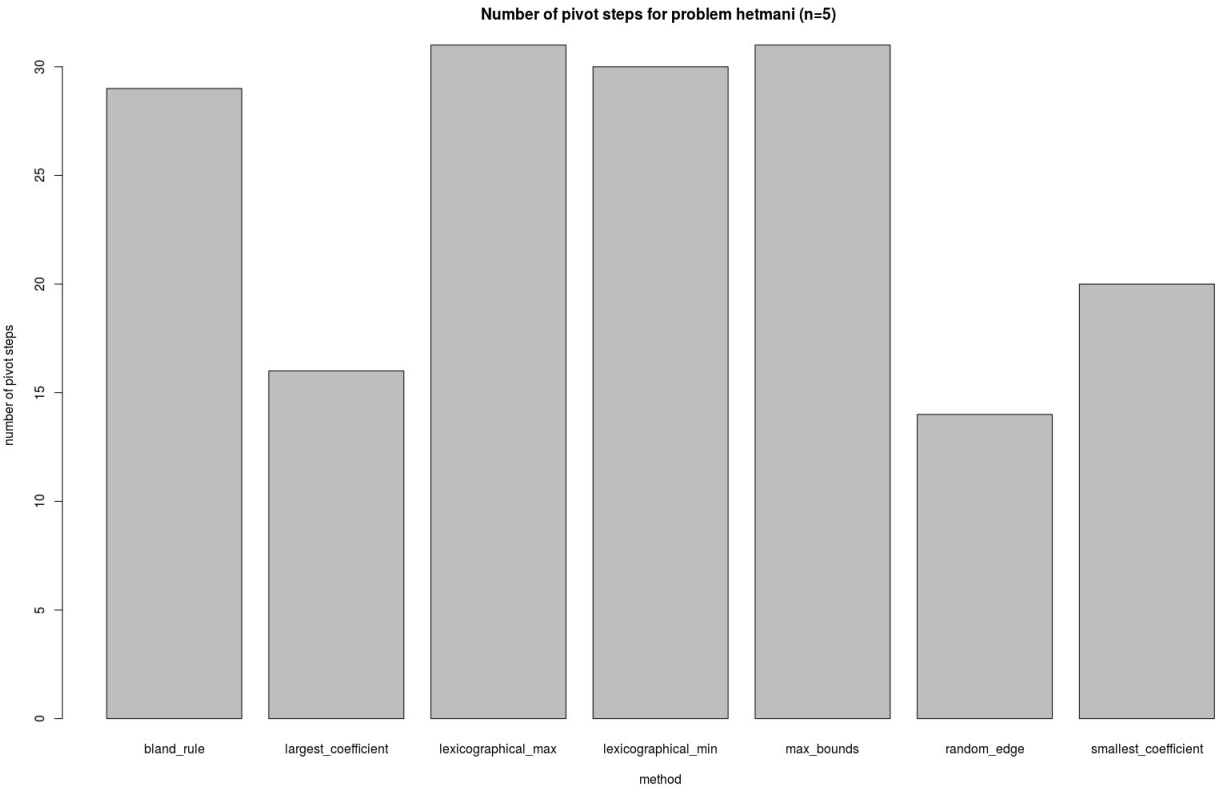
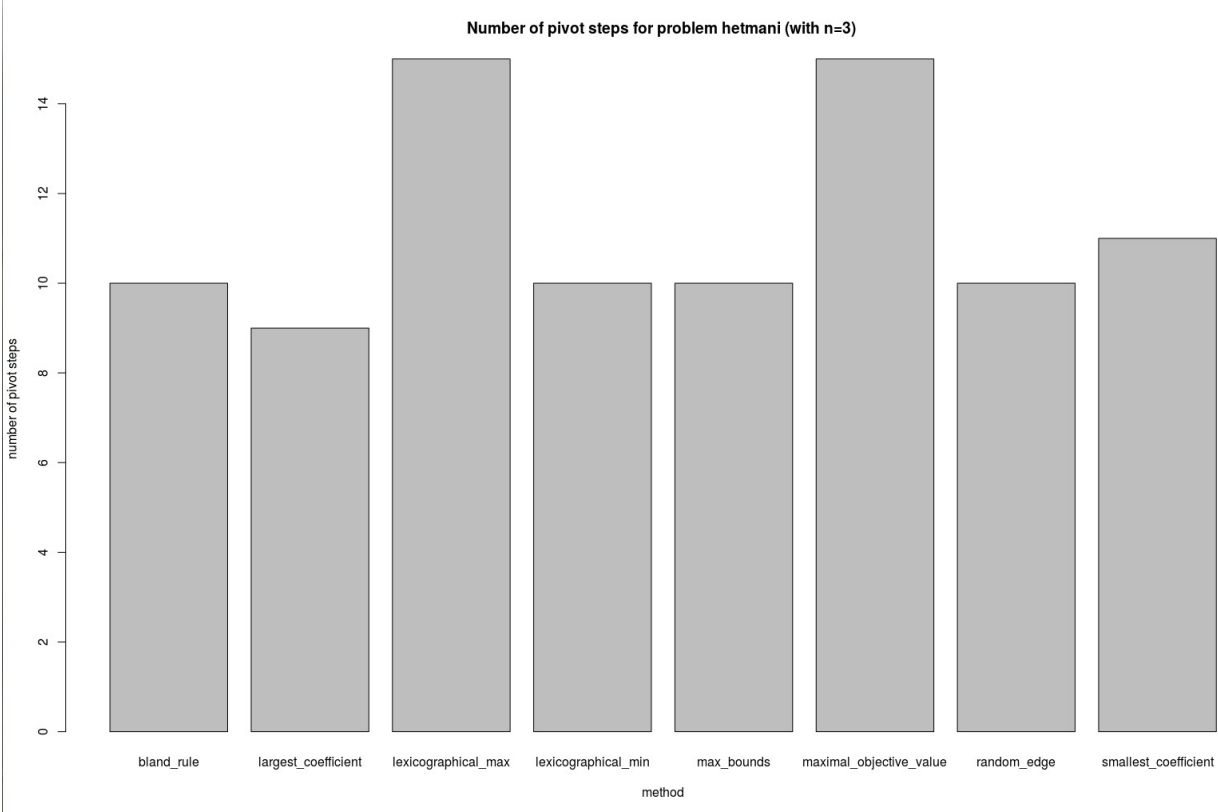


W przypadku problemu whiskas 2 metody dzielą się na dwie wyraźnie zróżnicowane grupy – metody które dobrze poradziły sobie z rozwiązaniem problemu (largest_coefficient i lexicographical_max, które znalazły optymalne rozwiązanie w 2 krokach) oraz pozostałe metody, które aby znaleźć optymalne rozwiązanie potrzebowały wykonać dużo więcej operacji.



Najcięższe do rozwiązania okazały się problemy hetmanów. Podczas gdy wszystkie pozostałe problemy liniowe rozwiązywane były w nie więcej niż 10-11 krokach, w przypadku problemu

hetmanów, nawet małych (rozpatrywane były problemy dla szachownicy 3x3 i 5x5), do rozwiązania problemu potrzeba było co najmniej kilkunastu kroków. Nieodpowiednią do tego typu problemów okazała się szczególnie metoda `maximal_objective_value`, która przy problemie `hetmani3` (problem hetmanów dla szachownicy 3x3) potrzebowała aż 15 kroków (najwięcej wśród rozpatrywanych metod), natomiast dla problemu `hetmanów5` nie zwróciła żadnej wartości (znajdowanie rozwiązania zostało zaniechane po ponad godzinie oczekiwania).



3) Wnioski:

Metodą która najlepiej poradziła sobie z rozwiązaniem przykładowych problemów liniowych jest `largest_coefficient`. Jak widać w większości problemów w oparciu o tę regułę znaleźć można było rozwiązanie optymalne w najmniejszej liczbie kroków (lub zbliżonej do najmniejszej). Dodatkowo w przypadku problemu hetmanów i `whiskas1` liczbę wykonywanych w ramach tej metody operacji była znacząco mniejsza niż przy większości pozostałych metod.

Dodatkowo średnia liczba kroków wykonanych przez metodę `largest_coefficient` we wszystkich testach operacji wyniosła 4,7 kroków, co jest najmniejszą średnią wartością kroków na cykl testów.