

Jacek Dąbrowski\*

## Algorytm selekcji klonalnej w marszrutowaniu pojazdów

### 1. Wprowadzenie

Sztuczne systemy immunologiczne – *Artificial Immune Systems* (AIS) to określenie obejmujące szereg technik obliczeniowych inspirowanych procesami zachodzącymi w układzie odpornościowym człowieka. Ich zastosowania obejmują analizę i kompresję danych, rozpoznawanie wzorców oraz optymalizację: dyskretną, jedno- i wielomodalną. W pracy przedstawiono algorytm inspirowany mechanizmem selekcji klonalnej limfocytów, który umożliwiłby układowi immunologicznemu szybką reakcję na pojawienie się patogenów oraz dopasowywanie zestawu przeciwciał do nowych zagrożeń (np. nowych szczepów wirusów czy bakterii). Algorytmy oparte na selekcji klonalnej wykazują pewne podobieństwa do algorytmów genetycznych i wielościeżkowych metod przeszukiwania. Podobnie jak one, mogą być skutecznymi metodami znajdowania rozwiązań NP-trudnych problemów optymalizacyjnych. Informacje na temat pozostałych rodzajów AIS oraz analizie ich przydatności dla różnych klas problemów można znaleźć m.in. w [1].

Problemy marszrutyzacji to określenie obejmujące całą rodzinę problemów optymalizacji dyskretnych, które łączy ogólna postać zagadnienia: wyznaczenie dostępnym pojazdom tras umożliwiających realizację określonego zadania transportowego w ramach przyjętych ograniczeń. Problemy należące do tej kategorii można bardzo łatwo odnieść do konkretnych, praktycznych zastosowań (np. doręczanie przesyłek). W pracy zbadano *Capacitated Vehicle Routing Problem* (CVRP), w którym każdy z pojazdów ma zdefiniowaną pojemność, a każdy z klientów oczekuje pewnej ilości jednostkowych towarów.

W rozdziale 2 przedstawiono rozpatrywany wariant problemu marszrutowania pojazdów oraz jego formalną definicję. Rozdział 3 zawiera podstawowe informacje na temat podstaw biologicznych selekcji klonalnej oraz szczegółowe informacje na temat algorytmu komputerowego. W rozdziale czwartym przedstawiono wyniki eksperymentów komputerowych, które miały na celu zbadanie wpływu parametrów algorytmu SK na uzyskiwane wyniki oraz porównanie ich z wynikami uzyskiwanymi przy użyciu innej techniki optymalizacyjnej – symulowanego wyżarzania. Pracę zamyka rozdział piąty, w którym zawarto najważniejsze wnioski płynące z przeprowadzonych badań.

---

\* Katedra Systemów Geoinformatycznych, Politechnika Gdańska

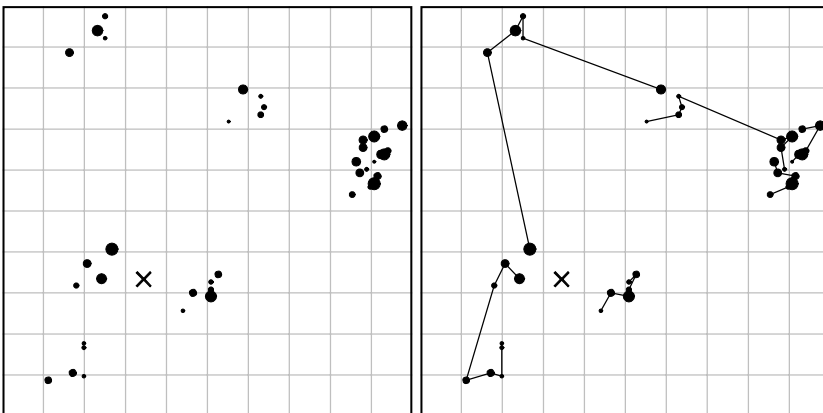
## 2. Marszrutowanie pojazdów

Problemy marszrutyzacji można postrzegać jako ogólniejszą, bardziej praktyczną, wersję problemu komiwojażera, w której również zdefiniowany mamy pewien graf reprezentujący węzły oraz połączenia pomiędzy nimi, natomiast samo zadanie transportowe zależy od konkretnych uwarunkowań: problemy marszrutyzacji różnią się między sobą między innymi ilością towarów do przewiezienia, liczbą pojazdów, ich pojemnością, ograniczeniami czasowymi czy też samym modelem transportu (np. dostawa, odbiór, obie czynności równolegle). Liczba możliwych teoretycznych wariantów jest bardzo duża, co więcej, każda praktyczna sytuacja generować będzie dodatkowe wymagania względem rozwiązań. Jest to główna przyczyna, dla której badania naukowe w znakomitej większości dotyczą jednego z kilku ściśle zdefiniowanych problemów marszrutowania.

### 2.1. Definicja problemu

*Capacitated Vehicle Routing Problem* można zdefiniować następująco: dany jest graf  $G = (V, E)$ , gdzie  $V$  jest zbiorem wierzchołków reprezentujących centralę ( $v_0$ ) oraz odbiorców ( $v_1, \dots, v_n$ ), a każdemu elementowi ze zbioru krawędzi  $E$  przyporządkowano nieujemną liczbę rzeczywistą – wagę połączenia. Ponadto wektor  $d: V \setminus \{v_0\} \rightarrow N_+$  zawiera informację o ilości towarów oczekiwanych przez każdego z odbiorców, a  $C$  jest stałą określającą pojemność pojedynczego pojazdu. Poprawnym rozwiązaniem CVRP jest zbiór takich ścieżek w grafie  $G$ , że każdy z wierzchołków  $v_1, \dots, v_n$  znajduje się na dokładnie jednej ścieżce, a suma oczekiwań odbiorców na każdej z nich nie przekracza pojemności pojazdu. Kosztem rozwiązania jest suma wag wszystkich krawędzi we wszystkich ścieżkach powiększona o wagi krawędzi pomiędzy centralą a początkami i zakończeniami ścieżek.

Rysunek 1 zawiera przykładową instancję problemu. Krzyżykiem oznaczono centralę, natomiast kropki reprezentują odbiorców, przy czym ich rozmiar jest proporcjonalny do oczekiwanej ilości towarów. Optymalne rozwiązanie uzyskano przy użyciu algorytmu typu *Branch-and-Cut* przedstawionego w [2].

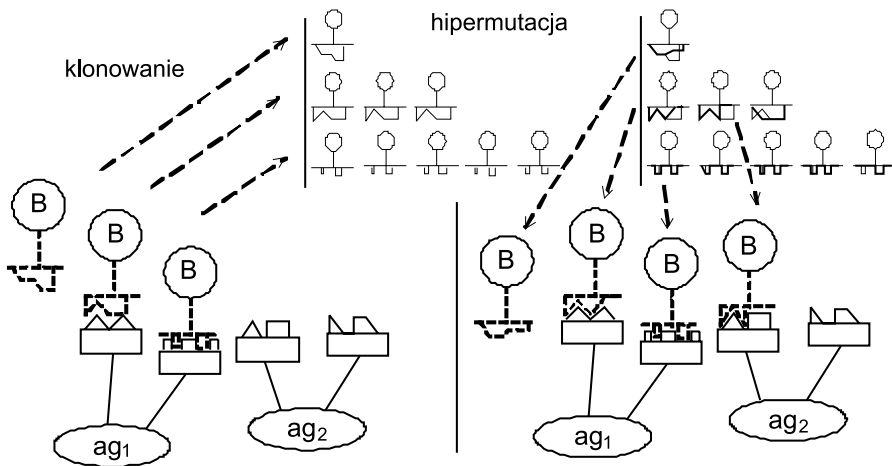


Rys. 1. Przykładowa instancja CVRP (B-n41-k6) oraz jej optymalne rozwiązanie

### 3. Selekcja klonalna

W ostatnich latach zaproponowano bardzo wiele nowych technik obliczeniowych inspirowanych funkcjonowaniem różnorodnych systemów biologicznych, których cechą wspólną jest umiejętność zdobywania i gromadzenia informacji. Jednym z takich systemów jest z pewnością układ odpornościowy (UO), którego podstawową funkcją jest ochrona organizmu przed zagrożeniami takimi jak bakterie czy wirusy (organizmy i cząstki mogące spowodować szkody w organizmie zbiorczo określa się mianem patogenów). Co ważne, ze względu na ogromną różnicę w tempie ewolucji między kręgowcami a organizmami jednokomórkowymi nie ma możliwości, aby pełna odporność została wypracowana w toku ewolucji kręgowców. Wynika to z obserwacji, że większość rodzajów patogenów, na które może trafić dany osobnik w czasie swojego życia, prawdopodobnie jeszcze nie istniała w momencie jego przyjścia na świat. Fakt, że pomimo to UO jest w stanie skutecznie chronić organizm, wyjaśnia teoria selekcji klonalnej limfocytów sformułowana przez Jerne w [3].

W dużym uproszczeniu krążące w organizmie limfocyty typu B (jeden z rodzajów tzw. białych komórek krwi) stanowią rodzaj detektorów potrafiących rozpoznawać patogeny, przy czym każdy z limfocytów ma ściśle określony kształt, który określa podzbiór rozpoznawalnych przezeń patogenów. Suma tych podzbiorów definiuje w pewnym sensie odporność całego organizmu. O ile zestaw limfocytów dostępny zaraz po urodzeniu zapewnia ochronę przed niektórymi zagrożeniami, to obrony przed innymi UO musi się nauczyć. Z pomocą przychodzi tu właśnie mechanizm SK, który został przedstawiony na rysunku 2.



Rys. 2. Mechanizm biologicznej selekcji klonalnej limfocytów

Limfocyty, które wejdą w fizyczny kontakt z pasującym do nich antygenem, zostają „aktywowane”, co powoduje wzrost tempa ich podziału. Im lepsze dopasowanie, tym więcej kopii danego limfocytu powstaje – ten etap nazywany jest ekspansją klonalną. Przy po-

dziale zawsze może dojść do mutacji kształtu detektorów, której stopień również zależy od stopnia aktywacji – im lepsze dopasowanie, tym mniejsza mutacja, im gorsze, tym większa. Drugi etap nazywany jest hipermutacją. Po prawej stronie rysunku zobrazowano oczekiwany efekt, w którym mutacje doprowadziły do powstania nowych, lepiej dopasowujących się limfocytów.

### 3.1. Algorytm selekcji klonalnej

W implementacji komputerowej algorytmu SK dla problemów optymalizacyjnych przyjmuje się, że istnieje pewien zbiór przeciwciał odpowiadających punktom w przestrzeni rozwiązań. Poszukuje się przeciwciał optymalizujących (minimalizujących lub maksymalizujących) wartość funkcji celu. Na podstawie jej wartości dla każdego przeciwciała wyznaczane jest pobudzenie, a co za tym idzie, liczba klonów i stopień mutacji. Najlepiej dopasowane przeciwciała (wyższa wartość funkcji celu) są klonowane (kopiowane) wiele razy, a klony poddawane są niewielkim mutacjom. W ten sposób dokładniej przegląda się okolice dobrych rozwiązań. Dla przeciwciał słabo dopasowanych tworzona jest mniejsza liczba kopii, które podlegają dużej mutacji, co odpowiada większemu przesunięciu w przestrzeni rozwiązań problemu. Poniższa lista przedstawia najważniejsze kroki algorytmu.

**Krok 1.** Zainicjalizuj  $P_0$ ;

**Krok 2.**  $t := 0$ ;

**Krok 3.**  $P_{t+1} := \emptyset$ ;

**Krok 4.** dla każdego  $Ab \in P_t$

4a.  $n_c :=$  liczba klonów  $Ab$ ;

4b.  $clones := \emptyset$ ;

4c. for  $i := 1$  to  $n_c$

$clones := clones \cup \text{Hipermutacja}(Ab)$ ;

4d.  $P_{t+1} := P_{t+1} \cup \text{Wybierz}(clones)$ ;

**Krok 5.**  $t := t + 1$ ;

**Krok 6.** jeśli nie spełniono warunków zakończenia: goto 3.

W kolejnych obiegach pętli tworzone są kopie dla każdego z przeciwciał w dotychczasowej puli. Zarówno liczba kopii, jak i stopień mutacji zależą od pozycji w rankingu przeciwciał. Nowa pula przeciwciał tworzona jest przez dokonanie selekcji turniejowej (rozmiar turnieju: 3) w każdym kolejnym zbiorze klonów. Warunkiem zakończenia pracy algorytmu jest przekroczenie limitu czasu.

### 3.2. Adaptacja do CVRP

Na adaptację metody SK do konkretnego problemu składają się przede wszystkim sposób reprezentacji i oceny rozwiązań oraz operator hipermutacji modyfikujący te rozwiązania. Prezentowany w pracy algorytm operuje na pełnych, prawidłowych rozwiązaniach CVRP. W pamięci komputera przechowywane są one w postaci zbioru wektorów, z których



każdy zawiera indeksy kolejnych klientów do odwiedzenia na danej trasie. Koszt rozwiązania to po prostu sumaryczna długość wszystkich tras pojazdów, z jednym ważnym zastrzeżeniem: jeśli wymagana liczba pojazdów przekracza dopuszczalną ich ilość, to koszt tras nadmiarowych jest podwajany. Rozwiązania inicjalizowane są losowo – początkowo dla każdego odbiorcy tworzona jest jedna trasa.

Zastosowany operator mutacji to jeden z najprostszych wariantów sąsiedztwa ( $M,P$ ) przedstawionego w [4] – operator (1,0). Operator losowo wybiera jednego z odbiorców, a następnie przenosi go do losowo wybranej trasy docelowej. Pamiętać należy, że operator sprawdza, czy ewentualne przeniesienie nie złamie ograniczenia na rozmiar pojazdu (w takim wypadku nie wykonuje operacji) oraz że sam punkt umieszczenia odbiorcy w trasie docelowej nie jest losowy – tylko taki, który gwarantuje minimalną długość uzupełnionej trasy.

## 4. Eksperymenty komputerowe

W celu sprawdzenia wydajności zaprezentowanych algorytmów przeprowadzono szereg eksperymentów komputerowych z użyciem różnorodnych instancji pochodzących z literatury. Zestaw, z którego skorzystano został skompletowany na potrzeby pracy [2] i jest dostępny w Internecie<sup>1</sup>. Badania przeprowadzono na komputerze klasy PC wyposażonym w procesor Intel Core 2 Duo 2.13 GHz.

### 4.1. Parametry algorytmu

W pierwszym etapie eksperymentów zadaniem było określenie takich parametrów algorytmu SK, które pozwoliłyby na uzyskiwanie dobrych wyników dla benchmarkowych instancji CVRP. Bazując na wcześniejszych doświadczeniach z problemem wierzchołkowego kolorowania grafu [5], spodziewaliśmy się, że najistotniejszy wpływ na jakość rozwiązań mają wielkość populacji oraz liczba klonów generowanych w każdej z iteracji. Eksperymenty potwierdziły tę zależność, zwrócić jednak należy uwagę, że o ile dla największych problemów w kolorowaniu rekomendowaliśmy rozmiar populacji rzędu pięciu osobników, to w przypadku CVRP i operatora (1,0) rekomendujemy wykorzystanie populacji o przynajmniej 15 osobnikach. Wynika to najprawdopodobniej z faktu, że rozmiar binarnej reprezentacji pojedynczego rozwiązania CVRP nawet dla największych instancji jest o rząd wielkości mniejszy od reprezentacji największych grafów benchmarkowych dla GCP, co przekłada się na czas potrzebny do wykonania kopii czy samej oceny nowego rozwiązania.

Średnia liczba klonów generowana na jedno przeciwiało również ma bardzo istotne znaczenie dla wyników generowanych przez algorytm SK. Różnica jakości rozwiązań uzyskiwanych w tym samym czasie przy średniej liczbie klonów 5 a 20 sięgała 10–20%. Dalsze

---

<sup>1</sup> <http://branchandcut.org/VRP>



zwiększanie liczby klonów nie miało większego znaczenia, ze względu na wybraną metodę selekcji (turniejową). Zastosowanie innej metody selekcji poprawia wydajność jedynie w początkowej fazie przeszukiwania.

Sumaryczną liczbę klonów można zredukować dzięki zróżnicowaniu liczby klonów na przeciwią. Jest to o tyle istotne, że w literaturze (np. [6]) spotyka się opisy algorytmów SK, w których ta cecha została „uproszczona”. Nasze badania wyraźnie wskazują, że jeśli lepsze rozwiązania otrzymują większą ilość kopii niż te gorsze, to ogólna wydajność algorytmu wzrasta.

## 4.2. Instancje benchmarkowe

W drugim etapie badań porównano wyniki osiągnięte przez algorytm SK z wynikami uzyskiwanymi przy użyciu innej techniki optymalizacyjnej: symulowanego wyżarzania – *simulated annealing* (SA). SA jest stosunkowo prostą techniką przeszukiwania lokalnego, w której w każdej kolejnej iteracji algorytm dokonuje pewnej losowej modyfikacji w stosunku do bieżącego rozwiązania (w tym wypadku – jeden krok operatora (1, 0)). Jeśli nowe rozwiązanie jest lepsze od poprzedniego – automatycznie jest akceptowane. Istnieje również możliwość, że rozwiązanie gorsze zostanie zaakceptowane, przy czym prawdopodobieństwo takiego zdarzenia maleje wraz z upływem czasu, zgodnie z przyjętym schematem schładzania.

**Tabela 1**  
Wyniki dla instancji benchmarkowych

Subset	tight	CW	SA	SK
<i>A</i>	0.92	1.54	1.55	1.48
<i>B</i>	0.92	1.71	1.64	1.62
<i>E</i>	0.92	1.57	1.52	1.37
<i>F</i>	0.94	1.34	1.45	1.33
<i>M</i>	0.94	1.47	1.82	1.42
<i>P</i>	0.94	1.60	1.48	1.41

Tabela 1 prezentuje uśrednione dane uzyskane dla instancji pochodzących ze zbioru testowego. Pierwsza kolumna określa podzbiór którego dotyczy dany wiersz, druga zawiera średnie zagęszczenie instancji (stosunek całkowitych oczekiwań odbiorców do sumarycznej pojemności pojazdów). Kolejne trzy kolumny to średni błąd względny najlepszego znalezionej rozwiązania, uzyskanego przy użyciu kolejno: zachłannej heurystyki Clarke–Wright [7], symulowanego wyżarzania oraz SK. Błąd względny obliczany jest względem kosztu optymalnego rozwiązania wyznaczonego w [2]. Algorytmy SA i SK pracowały przez pięć minut, zaczynając od rozwiązania losowego. Wyniki uzyskiwane przy użyciu algorytmu SK są zauważalnie lepsze od tych uzyskiwanych przez SA, który dla niektórych podzbiorów nie jest nawet w stanie uzyskać lepszych wyników od heurystyki zachłannej (w założonym czasie obliczeń). W przypadku inicjalizacji rozwiązań algorytmem CW



w oczywisty sposób jakość rozwiązań jest wyraźnie lepsza w pierwszych fazach przeszukiwania, jednak w dalszych etapach wyraźnie widać spadek jakości uzyskiwanych rozwiązań związany najprawdopodobniej z brakiem ich różnorodności.

## 5. Zakończenie

W pracy przedstawiono adaptację metody SK do rozwiązywania jednego z wariantów problemu marszrutowania pojazdów. Przeprowadzone badania pozwoliły na sformułowanie w rozdziale czwartym szeregu sugestii co do parametrów populacji algorytmu. Co więcej, algorytm oparty na SK uzyskuje w tym samym czasie lepsze rozwiązania niż algorytm symulowanego wyżarzania oparty na sąsiedztwie analogicznym do przedstawionego operatora mutacji.

Dalsze badania w tym temacie mogą obejmować sprawdzenie zachowania innych operatorów mutacji dla konkretnego wariantu problemu marszrutyzacji oraz zachowanie algorytmu w środowisku równoległym.

## Literatura

- [1] Garret S., *How do we evaluate artificial immune systems?* Evolutionary Computation 13(2), 2005, 145.
- [2] Ralphs T.K., Koller G., Julstrom B.A., *On the capacitated vehicle routing problem*. Mathematical Programming 94 (2–3), 2003, 343.
- [3] Jerne N.K., *Towards a network theory of the immune system*. Annales d'immunologie 1–2, 1974, 373.
- [4] Taillard É., *Parallel iterative search methods for vehicle routing problems*. Networks 23, 1993, 661.
- [5] Dąbrowski J., Kubale M., *Computer experiments with a parallel clonal selection algorithm for the Graph Coloring Problem*. Proc. of IEEE International Symposium on Parallel and Distributed Processing 2008.
- [6] de Castro L.N., von Zuben F.J., *Learning and optimization using the clonal selection principle*. IEEE Trans. Evolutionary Computation 6 (3), 2002, 239.
- [7] Aronson L.D., *Algorithms for vehicle routing – a survey*. Delft University of Technology Technical Report 1996, 21.

