

Algorytmy mrówkowe – wprowadzenie.

Jakub Zajkowski

1 *Wstęp i rys historyczny*

Algorytmy mrówkowe to grupa procesów służących przede wszystkim do poszukiwania dróg w grafie. Z formalnego punktu widzenia algorytmy mrówkowe, pomimo swojej nazwy, nie są algorytmami. Z definicji algorytmu wynika, że powinien on zwracać zawsze rozwiązanie poprawne. Natomiast heurystyka oparta o naśladowanie mrówek, ze względu na swoją charakterystykę, nie gwarantuje znalezienia rozwiązania optymalnego. Przykładowo, przy zadaniu poszukiwania najkrótszej drogi możemy nie otrzymać właściwej odpowiedzi. Jednak w polskiej literaturze utarło się używanie określenia „algorytmy mrówkowe” i w ramach tego referatu podtrzymana zostanie ta konwencja. Algorytmy mrówkowe należą do procesów przetwarzania, które klasyfikuje się do tzw. obliczeń inteligentnych (ang. intelligent computing). Są to metody oparte o zasadę inteligencji roju (ang. swarm intelligence).

Algorytm mrówkowy został pierwotnie zaproponowany w ramach pracy doktorskiej Marco Dorigo w 1992. Pierwotnie służył do rozwiązywania problemu szukania optymalnej drogi w grafie. Początkowo była to technika nieszczególnie doceniana, ze względu na problem szybkiej zbieżności do optimum lokalnych w podstawowej wersji. Przedstawiono jednak modyfikacje podstawowego algorytmu mrówkowego, co pozwoliło na szersze zastosowanie tej techniki. Stosunkowo szybko znalazła zastosowania w szerszej gamie problemów, np. harmonogramowania, poszukiwania partycji i pokrycia zbiorów, czy nawet wykrywania krawędzi na obrazach rastrowych.

Jak już wcześniej wspomniano, algorytmy mrówkowe nie zwracają zawsze optymalnego rozwiązania. Dlatego znajdują one zastosowanie w przypadkach gdy rozwiązanie „prawie optymalne” jest wystarczające, oraz gdy użycie algorytmów faktycznie zwracających optymalne rozwiązanie jest albo niemożliwe, albo zbyt złożone obliczeniowo.

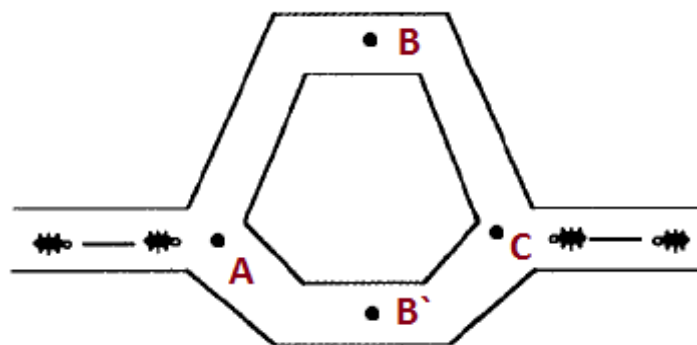
2 *Zasada działania, przedstawienie heurystyki*

Heurystyka algorytmu mrówkowego polega na imitacji współdziałania mrówek przy zbieraniu pokarmu w otoczeniu mrowiska. Przez heurystykę rozumie się tu rozwiązanie pewnego problemu nie w oparciu o udowodniony matematycznie algorytm, lecz przez imitację zjawiska naturalnego. Robotnica opuszczająca mrowisko w celu poszukiwania żywności nie ma wiedzy o otaczającym je świecie. Jej zmysły ogarniają niewielką przestrzeń wokół jej aktualnej pozycji. W jaki więc sposób mrówki-robotnice są w stanie wyżywić mrowisko zdobywając pożywienie w sposób zorganizowany? Wyobraźmy sobie sytuację : w pewnym mrowisku zaczyna się nowy dzień. Pierwsza mrówka wychodzi z mrowiska. Z jednej strony nie „widzi” zbyt daleko, z drugiej zaś inne mrówki nie pozostawiły jej żadnych informacji (w końcu jako pierwsza wyszła tego dnia z mrowiska). Podejmuje więc wędrówkę w losowym kierunku. Oczywiście nasza mrówka nie jest zupełnie ślepa i posiada pewien węch. Dlatego jeśli minie pożywienie w swoim „bezpośrednim otoczeniu” (w zasięgu swoich skromnych zmysłów), weźmie część tego pożywienia i wróci z nim do mrowiska. Ale właściwie w jaki sposób mrówka wie jak wrócić do mrowiska ? Otóż na końcu odwłoka mrówki znajduje się gruczoł, który zostawia

za nią ślad feromonowy. Zatem robotnica może wrócić, podążając za własnym śladem feromonowym (mrówki potrafią odróżniać swój osobisty feromon). Po powrocie do mrowiska, robotnica przetwarza pożywienie. Przyjmijmy, że druga mrówka opuszcza mrowisko po powrocie pierwszej. W jaki sposób następna robotnica podejmie decyzję o kierunku ? Otoczenie wejścia mrowiska nie jest takie samo, jak dla jej poprzedniczki. Co prawda wciąż niewiele wie o otaczającym ją środowisku, ale wyczuwa ślad feromonu, który pozostawiła jej prekursorka. Dlatego, gdy ta mrówka będzie podejmować decyzję, uwzględni widoczny dla niej ślad feromonowy. Im silniejszy ślad feromonu, tym większe prawdopodobieństwo, że druga mrówka podąży śladem pierwszej. Kiedy zaś ślad feromonowy będzie silniejszy ? Mają na to wpływ dwa czynniki:

- odległość od mrowiska do żerowiska jest niewielka. Dzieje się tak, gdy mrówki pokonują niewielką odległość od mrowiska do żerowiska i z powrotem. Zajmuje im to więc niewiele czasu, zatem zostawiony przez nie feromon nie zdąży wyparować.
- częstość przechodzenia mrówek po danej ścieżce jest wysoka. Oczywiście dla naszej hipotetycznej sytuacji nie jest to adekwatne (na razie tylko jednak mrówka w ogóle wyszła z mrowiska). Jednak dla wielu robotnic jest szansa, że więcej niż jedna przejdzie tą samą ścieżką (lub taką ścieżką, która ma część wspólną z inną, już „wydeptaną” ścieżką). Każda następna mrówka zostawia swój ślad feromonowy, który dodaje się do już zostawionego na tej ścieżce.

Powyżej wymieniono mechanizmy wzmacniania śladu feromonowego i wspomniano o mechanizmie jego osłabiania. Feromonom paruje, w wyniku czego słabiej przyciąga uwagę mrówek.



Rys. 1 Przykładowa alternatywa ścieżek do żerowiska

Postarajmy uogólnić te proste zasady na sytuację, w której po różnych ścieżkach podróżuje większa liczba robotnic. Załóżmy istnienie dwóch dróg, które prowadzą do jednego żerowiska, jednak jedna z nich (ABC) jest wyraźnie dłuższa niż druga ($AB'C$) (Rys. 1). Mrówka podążająca dłuższą ścieżką, pozostawi mniej intensywny ślad feromonowy, ponieważ czas jaki zajmie jej dotarcie do żerowiska będzie dłuższy, niż gdyby obrała drogę krótszą. Zanim wróci do mrowiska więcej z jej feromonu wyparuje. Przy większej liczbie robotnic ma to kluczowe znaczenie dla motywowania działań kolejnych robotnic.

3 Algorytm mrówkowy, przedstawienie

Z przedstawionego w rozdziale 2 schematu można wysnuć, iż mrówkom nie zależy na odnalezieniu najlepszego źródła pokarmu w okolicy, do której mogą dojść z kolonii. Zadowolają się rozwiązaniem „rozsądnym” - gdy pokarm jest stosunkowo blisko i w ilości wystarczającej, by jakaś robotnica się nim zainteresowała. Można to zauważyć podczas obserwacji występujących w naturze mrówek. Zdecydowana większość kieruje się na kilka ustalonych szlaków, pojedyncze osobniki zbaczają z nich aby szukać alternatywnych ścieżek. Jednak przy poszukiwaniu rozwiązań problemów sformalizowanych matematycznie potrzebujemy większego rozproszenia „mrówek”. Chcemy, żeby w sposób reprezentatywny eksplorowały one uniwersum rozwiązań. Jest to potrzebne, aby algorytm mrówkowy mógł odnaleźć “odpowiednio dobre” rozwiązania, które mogą być daleko od optimum lokalnych. Należy przypomnieć, że algorytm mrówkowy nie zawsze zwraca rozwiązanie optymalne. Zachowując wysoki poziom ogólności, kroki heurystyki algorytmu mrówkowego są następujące:

procedura AlgorytmMrówkowy

dopóki nie nastąpił warunek końca

generujRozwiązania()

zaktualizujŚcieżkiFeromonów()

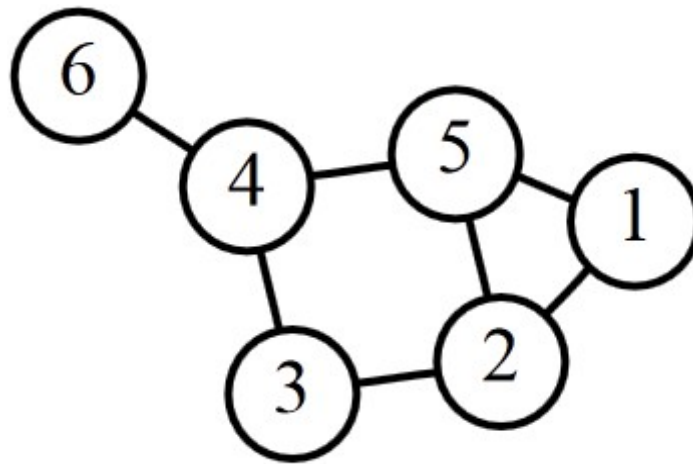
koniec dopóki

koniec

Powyższe sformułowania mogą wydać się dość enigmatyczne, jest to jednak postać możliwie ogólna, która pasuje praktycznie do wszystkich zastosowań AM. Spróbujmy uściślić poszczególne kroki.

3.1 generujRozwiązania()

Jest to część algorytmu mrówkowego, która naśladuje „rozchodzenie” mrówek wokół mrowiska. W przeciwieństwie do sytuacji naturalnej, działanie mrówek w algorytmie mrówkowym jest iteracyjne. Przy każdej iteracji każda mrówka generuje pełne rozwiązanie. Generuje jest tu kluczowym określeniem, ponieważ mrówka będzie poruszać się po przestrzeni rozwiązań w sposób uwzględniający czynniki losowe.



Rys 2 Przykładowy graf do przeszukiwania

Założmy, że chcemy, aby nasz algorytm mrówkowy szukał optymalnej ścieżki między węzłami 1 i 6 w grafie przedstawionym na rys. 2. Każda mrówka w fazie generujRozwiązania będzie musiała przejść graf od węzła 1 do 6. Dopiero cała ścieżka będzie uwzględniona jako wynik działań mrówki. Przejdźmy więc do omówienia jak mrówka będzie podejmować decyzje podczas przechodzenia grafu, czyli podczas fazy generujRozwiązania. Mrówka zaczyna podróż w węźle 1. Może przejść do węzła 2 lub 5. Aby określić w którym się znajdzie, należy ustalić prawdopodobieństwa przejścia. Formalnie prawdopodobieństwo przejścia z węzła i do węzła j dla k -tej mrówki określa wzór :

$$p_{i,j}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\sigma_{ij})^\beta}{\sum_{h \notin \text{tabu}_k} ((\tau_{ih})^\alpha (\sigma_{ih})^\beta)}, & \text{jeśli } j \in \text{tabu}_k \\ 0, & \text{w przeciwnym przypadku} \end{cases}$$

gdzie :

τ_{ij} to ilość feromonu na łuku ze stanu i do stanu j

σ_{ij} to funkcja wskazująca tak zwaną „atrakcyjność” przejścia z i do j . Używa się jej w dość specyficznych przypadkach, żeby oddać semantykę optymalizowanej dziedziny. Przykładowo, dla poszukiwania najkrótszej drogi w ważonym grafie, σ_{ij} może być najwyższe dla tam, gdzie waga łuku jest najniższa. W ten sposób modelujemy zachowanie mrówki, która będzie preferowała krótkie drogi między węzłami.

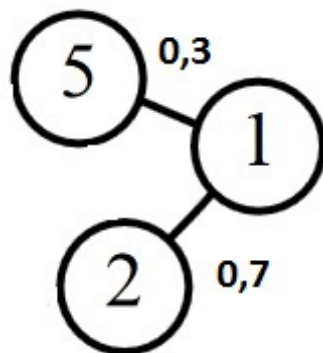
α, β to współczynniki istotności powyższych dwóch wpływów na decyzję mrówki. Dzięki nim możemy sterować, czy mrówka będzie kierować się raczej feromonem, czy raczej informacją o „atrakcyjności” wynikającą z wiedzy o problemie. Nie ma takich wartości tych parametrów, które byłyby poprawne dla wszystkich rozwiązań, należy dobrać je do konkretnego problemu.

tabu_k to zbiór odwiedzonych już przez mrówkę węzłów w danej iteracji generujRozwiązania. Zapamiętanie poprzednich węzłów jest potrzebne w celu uniknięcia zapętlania się mrówki. Przykładowo, dla grafu z rys. 2 robotnica mogłaby w nieskończoność poruszać się między węzłami 1, 2 i 5

Na potrzeby niniejszych rozważań nie jest istotne, w jaki sposób obliczane są

prawdopodobieństwa obrania poszczególnych ścieżek. Ważne jest natomiast, że uwzględniają one siłę feromonu i lokalną strukturę grafu. Celowo pominięto omawianie w szczegółach ten aspekt, ponieważ w dużym stopniu zależy od dziedziny rozwiązywanego problemu.

Przyjmijmy, że mrówka w węźle 1 ma taki rozkład prawdopodobieństwa przejścia do węzłów 5 i 2 jak na rys. 3. Aby określić do jakiego węzła przejdzie mrówka, musimy wygenerować liczbę losową z przedziału $<0;1>$ i na tej podstawie przenieść ją do następnego węzła. Wyższa wartość prawdopodobieństwa nie oznacza wcale, że mrówka musi przejść po „bardziej prawdopodobnym” łuku. W tym miejscu procedury objawia się jej probabilistyczny charakter. Generalnie mrówki będą podążać po takich ścieżkach, które są bardziej „prawdopodobne” (uwzględniając feromon i lokalną topologię). Jednak istnieje zawsze szansa, że robotnica podąży w innym kierunku, odrzucając dotychczasowe rozwiązania o wysokim prawdopodobieństwie przejścia po łukach i będzie eksplorować rozwiązanie dotychczas niesprawdzone.



Rys 3 Przykładowy rozkład prawdopodobieństwa obrania ścieżek przez mrówkę

Przyjmijmy, że znalazła się w stanie 2. W kolejnych krokach mrówka będzie powtarzać tę czynność. Wpierw „ustali”, jakie są prawdopodobieństwa przejścia do kolejnych węzłów (z pominięciem tych węzłów, które już odwiedziła), następnie przejdzie do wylosowanego węzła. Będzie się to powtarzać do momentu, w którym znajdzie warunek końca. Dla naszych rozważań warunek końca znajdzie wtedy, gdy mrówka znajdzie się w węźle 6.

3.2 zaktualizujŚcieżkiFeromonów()

Gdy każda mrówka znajdzie rozwiązanie problemu, następuje aktualizacja feromonów na łukach grafu. W przeciwieństwie do sytuacji naturalnej, mrówki w algorytmie mrówkowym nie zostawiają feromonu w trakcie podróży. Dopiero po odnalezieniu pełnego rozwiązania możemy określić ile feromonu można wprowadzić na łuki od węzła początkowego do końcowego. Dzieje się tak z dwóch powodów. Po pierwsze, dla niektórych problemów mrówka może nie znaleźć rozwiązania, jeśli będzie losowo poruszać się po uniwersum problemu. W takim wypadku nie chcemy żeby zostawiała feromon, ponieważ będzie wprowadzać inne mrówki w błąd. Z drugiej zaś strony rozwiązanie niektórych mrówek może być gorsze niż innych i chcemy promować rozwiązania lepsze. Przez promocję rozwiązania rozumiemy tu dodanie większej ilości feromonu na ścieżce do rozwiązania lepszego. Przykładowo, jeśli naszym grafem byłaby mapa drogowa polski i chcielibyśmy znaleźć trasę z Wrocławia do Warszawy, to jedna mrówka mogłaby przejść przez Łódź, a druga przez Kraków. Krótsza oczywiście byłaby

droga przez Łódź. W środowisku naturalnym dyskryminacja rozwiązań gorszych następuje automatycznie ponieważ pozostawiony feromon od razu zaczyna parować. Zatem im dalej mrówka pójdzie tym mniej feromonu za nią zostanie. W symulacji komputerowej nie możemy na tym mechanizmie polegać, gdyż operujemy w zdyskretyzowanych iteracjach, a nie na bazie ciągłego czasu. Regułę aktualizacji feromonu między węzłami i, j można zapisać jako:

$$\tau_{i,j}^k := (1 - \rho) \tau_{i,j}^k + \Delta \tau_{i,j}^k$$

ρ to współczynnik wyparowywania feromonu (im szybciej, tym szybciej będą „zapominane” stare rozwiązania). W publikacjach przedstawia się różne, często specjalizowane pod konkretne problemy, formuły obliczania $\Delta \tau_{i,j}^k$. My podamy najbardziej uniwersalny, „klasyczny” przykład:

$$\Delta \tau_{i,j}^k = \begin{cases} \frac{Q}{L_k} & \text{jeśli węzeł } k \text{ należy do rozwiązania związanego z mrówką } k \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

W powyższym równaniu Q to pewna dobrana stała (parametr procedury), natomiast L_k to koszt rozwiązania związanego z mrówką k . Dla problemu poszukiwania optymalnej ścieżki w grafie, L_k może być długością (lub ważoną długością) tej ścieżki. Po zakończeniu aktualizacji ścieżek feromonowych mrówki są „cofane” do punktu wyjścia, skąd rozpoczynają znowu poszukiwanie rozwiązań.

3.3 warunekKońcowy

Warunek końcowy określa, kiedy należy przerwać powtarzania akcji szukania rozwiązań i aktualizacji feromonów. Zazwyczaj jest to z góry określona liczba iteracji (np. 20 powtórzeń). Jednak można wyobrazić sobie bardziej wysublimowane przykłady, na przykład procedura zakończy się, jeśli przez 5 kolejnych iteracji wartość najlepszego znalezionej rozwiązania nie poprawi się. Dokładne dopasowanie warunku końcowego wymaga analizy problemu i eksperymentów.

4 Podsumowanie

Algorytmy mrówkowe to stosunkowo nowe podejście w dziedzinie obliczeń inteligentnych. Służy głównie optymalizacji, jednak jego zastosowanie nie daje gwarancji znalezienia optimum globalnego. Algorytmy mrówkowe można stosować do szerokiego spektrum problemów, w których rozwiązanie można przedstawić jako szereg kroków, które może zrobić wirtualna mrówka. Istnieje wiele wariacji tej techniki, często dużo bardziej skomplikowane niż zaprezentowany w tej pracy schemat działań. Algorytmy mrówkowe znalazły zastosowanie w różnych dziedzinach, np. problemach poszukiwania dróg w grafie, szukania pokryw i partycji zbiorów, problemy alokacji zasobów, harmonogramowania

5 Bibliografia

[1] http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms, 7.10.2011

[2] *Structural topology optimization using ant colony optimization algorithm*, G.

Luh, C. Lin, Applied Soft Computing (2009)

[3] *Ant colony algorithm for traffic signal timing optimization*, H. Jiajia, H. Zalen, Advances in Engineering Software (2011)

[4] *Novel Approach to Nonlinear PID Parameter Optimization Using Ant Colony Optimization Algorithm*, H. Duan, Wang D., Yu X.

[5] Program AntSim v. 1.1, www.nightlab.ch/antsim.php