

Marek LIBURA

Instytut Badań Systemowych
Polskiej Akademii Nauk

O OCENIE EFEKTYWNOŚCI ALGORYTMÓW ZRANDOMIZOWANYCH

Streszczenie. W pracy dyskutowane jest możliwość zastosowania podejścia bayesowskiego do oceny efektywności algorytmów zrandomizowanych. Podane są proste przykłady algorytmów tego typu dla zadań programowania całkowitoliczbowego i omówione są przyczyny przemawiające za ich stosowaniem. Następnie przedstawione są zasadnicze elementy bayesowskiej teorii decyzji i pokazane jest, jak bayesowskie modele decyzyjne mogą być użyte w konstrukcji kryteriów stopu algorytmów zrandomizowanych i ocenie ich dokładności.

WPROWADZENIE

Większość algorytmów stosowanych w programowaniu całkowitoliczbowym, to algorytmy deterministyczne. Algorytmy tego typu dają to samo rozwiązanie dla tych samych danych wejściowych i wszystkie kroki ich są jednoznacznie określone. Wśród algorytmów niedeterministycznych celowe wydaje się wyróżnienie dwóch klas algorytmów: algorytmów zrandomizowanych i schematów losowych.

Algorytm zrandomizowany charakteryzuje się tym, że jego schemat działania jest w istocie schematem deterministycznym, a jedynie niektóre kroki algorytmu zależne są od pewnych mechanizmów losowych. Przez schemat losowy będzie rozumiany algorytm, którego działanie opiera się na idei próbkowania zbioru rozwiązań. Podział taki nie jest oczywiście rozłączny, niemniej jednak wydaje się, że rozróżnienie obu klas algorytmów podkreśla ich podstawowe cechy.

W dalszej części będziemy zajmować się głównie algorytmami zrandomizowanymi. Przykładem tego typu algorytmu jest poniższa procedura RGREEDY dla wyznaczania elementu niezależnego o maksymalnej wadze. Do opisu tej procedury niezbędne jest wprowadzenie kilku pojęć.

Niech $E = \{e_1, \dots, e_n\}$ będzie danym zbiorem n -elementowym i niech (E, M) , gdzie $M \subseteq 2^E$ będzie systemem niezależności, co oznacza, że dla każdego $y \in M$ zachodzi warunek:

$$x \subseteq y \Rightarrow x \in M \quad (1)$$

Dodatkowo dana jest funkcja wag $c: 2^E \rightarrow R$ taka, że

$$c(y) = \sum_{e \in y} c(e) \quad (2)$$

Należy wyznaczyć element $x \in M$ (to znaczy element niezależny) o największej wadze. Odpowiednie zadanie programowania matematycznego ma postać następującą:

$$\max_{x \in M} c(x) \quad (3)$$

W powyższej postaci można sformułować wiele zadań programowania całkowitoliczbowego, nadając odpowiednią interpretację funkcji c i konstruując odpowiednio system niezależności (E, M) . Można w ten sposób zapisać na przykład binarne zadanie załadunku, które zazwyczaj formułuje się następująco:

$$\begin{aligned} \max \sum_{i=1}^n c_i z_i \\ \sum_{i=1}^n a_i z_i \leq b, \end{aligned} \quad (4)$$

$$z_i = 0 \text{ lub } 1 \text{ dla } i = 1, \dots, n$$

gdzie:

$$c_i, a_i \in R_+, \quad i = 1, \dots, n.$$

W tym przypadku zbiór $E = \{1, \dots, n\}$. Element $y \subseteq E$ należy do M wtedy i tylko wtedy, gdy $\sum_{i \in y} a_i \leq b$. Funkcja c dla $y \subseteq E$ określona jest następująco:

$$c(y) = \sum_{i \in y} c_i. \quad (5)$$

Jeśli x^* jest rozwiązaniem optymalnym zadania (3), wówczas rozwiązanie optymalne z^* zadania (4) jest określone następująco:

$$z_i^* = \begin{cases} 1 & \text{jeśli } i \in x^* \\ 0 & \text{w przeciwnym przypadku} \end{cases} \quad (6)$$

gdzie: $i = 1, \dots, n$.

W podobny sposób można sformułować zadanie pakowania, skojarzenia w grafie itp.

Poniższa procedura RGREEDY korzysta z pewnych parametrów $u(e_i)$ zwanych efektywnościami elementów e_i , $i = 1, \dots, n$. Ich zdefiniowanie zależy od zadania, które jest formułowane w postaci (3). Przy rozważaniu sensu fizycznego konkretnego zadania mają one zwykle interpretację przyrostu wagi (zysku) wnoszonego przez dany element $e \in E$ na jednostkę zużywanego zasobu. Na przykład w przypadku zadania załadunku można przyjąć, że $u_i = c_i/a_i$, $i = 1, \dots, n$. W przypadku innych zadań obliczenie efektywności może być bardziej złożone. Ponadto parametry te mogą się zmieniać dynamicznie w trakcie obliczeń.

PROCEDURA RGREEDY

1^o $S := E$; rozwiązanie aktualne $x^R := \emptyset$ (zbiór pusty).

2^o Oblicz $u(e)$ dla $e \in S$ i ponumeruj elementy zbioru S w taki sposób, aby

$$u(e_1) \geq u(e_2) \geq \dots \geq u(e_{|S|}). \quad (7)$$

gdzie $|S|$ jest mocą zbioru S .

3^o Zgodnie z rozkładem $P_{|S|}$ wylosuj element $e \in S$.

4^o Jeśli $\{e\} \cup x^R \in M$ podstaw

$$\begin{aligned} x^R &: x^R \cup \{e\} \\ S &: S \setminus \{e\} \end{aligned}$$

5^o Jeśli $S = \emptyset$: STOP; w przeciwnym przypadku idź do 2^o.

Jedyną różnicą między typowym algorytmem GREEDY dla zadania (3) a powyższą procedurą jest użycie losowego mechanizmu w punkcie 3^o. Rozkład $P_{|S|}$ jest opisany wektorem prawdopodobieństw

$$(p_1, \dots, p_S), \quad \sum_{i=1}^{|S|} p_i = 1,$$

gdzie p_i , $i = 1, \dots, S$ jest prawdopodobieństwem wyboru elementu e_i w kroku 3^o. W zwykłym algorytmie GREEDY mamy $p_1 = 1$, $p_i = 0$ dla $i = 2, \dots, |S|$. Zauważmy, że jeśli prawdopodobieństwa p_i są dodatnie we wszystkich krokach algorytmu, to prawdopodobieństwo znalezienia rozwiązania optymalnego przez algorytm RGREEDY jest również dodatnie. Algorytm powyższy jest bardzo prosty. Zasadniczy schemat algorytmu ma złożoność liniową $O(n)$, jeśli

nie uwzględnia się porządkowań zbiorów, generowania liczb losowych i badania przynależności zbioru M . Nawet po uwzględnieniu tych wszystkich elementów algorytm zazwyczaj nie jest czasochłonny i przy stosunkowo małym nakładzie obliczeń możliwe jest wygenerowanie znacznej liczby rozwiązań przybliżonych przez wielokrotne użycie procedury RGREEDY dla tego samego zadania. Co więcej, rozwiązania otrzymywane w ten sposób są często dobrym przybliżeniem rozwiązania optymalnego. Przeświadczenie takie zwykle potwierdzone jest eksperymentami obliczeniowymi, natomiast brak często teoretycznego uzasadnienia tego faktu.

Powyższa procedura jest przykładem algorytmu zrandomizowanego przybliżonego, to znaczy takiego, który nie zawsze gwarantuje uzyskanie rozwiązania optymalnego. Randomizacja może być wprowadzona z korzyścią także w algorytmach dokładnych. Przykładem takiego algorytmu jest procedura podziału i oszacowań, w której gałęzienie (wybór wierzchołka drzewa do podziału) wykonywane jest w sposób losowy zgodnie z rozkładem prawdopodobieństwa, zależnym od aktualnego stanu przeglądu drzewa. Innym przedstawicielem takich algorytmów może być metoda sortowania zbioru, polegająca na wstępnym losowym przemieszaniu danych wejściowych, a następnie zastosowaniu dowolnego dokładnego algorytmu sortowania [7].

Randomizacja tego typu pozwala często na obniżenie złożoności obliczeniowej algorytmu w sensie wartości oczekiwanej nakładu obliczeń. Analiza średniej złożoności obliczeniowej tego typu algorytmów jest zwykle trudna i ogranicza się do stosunkowo prostych schematów obliczeń. Podstawowa intuicja, która przemawia za tego typu randomizacją, sprowadza się do tego, że algorytm z losową realizacją niektórych kroków jest mniej wrażliwy na występowanie patologicznych zadań. Potwierdzają to również eksperymenty obliczeniowe. Charakterystyczny jest tu przykład algorytmu simplex, który wykazuje praktycznie niską wielomianową złożoność obliczeniową, jednakże (ze względu na istnienie trudnych dla niego zadań) jego złożoność dla najgorszego przypadku jest wykładnicza przy stosowanych zwykle regułach wyboru zmiennej wchodzącej do bazy. Tego typu trudne przykłady zostały skonstruowane w pracy [3] i rozwiązanie ich wymaga wykładniczej (w funkcji liczby zmiennych) liczby iteracji algorytmu simplex. Jednakże eksperymenty obliczeniowe [2] pokazują, że wprowadzenia randomizacji polegającej na losowym wyborze zmiennej wchodzącej do bazy daje algorytm o niskiej wielomianowej złożoności obliczeniowej.

Technika oceny algorytmów opisana w dalszej części pracy może być stosowana do zrandomizowanych algorytmów przybliżonych. W wersji deterministycznej algorytmy te mają czasem znaną gwarantowaną dokładność (rozumianą jako iloraz maksymalnego dla danej klasy zadań błędu wnoszonego przez algorytm, przez wartość rozwiązania optymalnego), z tym że dokładność ta zwykle nie jest praktycznie akceptowalna. Na przykład, najprostszy algorytm GREEDY dla zadania (3) ma gwarantowaną dokładność $1/k$ (patrz [4]), gdzie k jest minimalną liczbą matroidów, których przecięciem jest system nie-

zależności M . W innych zadaniach gwarantowana dokładność może być dowolnie niska dla odpowiednio dużej liczby zmiennych. Co więcej, bardzo mało prawdopodobne jest skonstruowanie wielomianowych algorytmów, dających nietrywialną gwarantowaną dokładność dla szerokiej klasy interesujących praktycznie zadań określonych jako silnie NP-zupełne [9].

Mimo tych pesymistycznych wyników teoretycznych przybliżone algorytmy zrandomizowane dla tej klasy zadań dają często całkiem dobre rozwiązania. Bayesowska teoria decyzji (patrz np. [5]) może być tu traktowana jako pewne sformalizowane uzasadnienie przekonań i obserwacji dotyczących efektywności tego typu algorytmów.

Istotnymi warunkami stosowalności tej techniki oceny algorytmów są:

1° Istnienie praktycznie efektywnego przybliżonego algorytmu zrandomizowanego dla danego typu zadania (to znaczy algorytmu dającego dobra przybliżenie w stosunkowo krótkim czasie).

2° Dobre rozeznanie dotyczące zachowania się tego algorytmu dla zadań z danej grupy, pozwalające na założenie a priori niezbędnych rozkładów prawdopodobieństwa.

Pierwsze próby zastosowania podejścia bayesowskiego dla zadań nieważonego pokrycia zbiorów podjęto w pracy [6]. Podejście to było inspirowane pracą [8] i dobrymi wynikami losowych algorytmów dla zadań optymalizacji globalnej uzyskanymi w [1].

W dalszej części pracy przyjmujemy następujący sposób oznaczania zmiennych losowych i rozkładów prawdopodobieństwa. Zmienne losowe są oznaczane dużymi literami alfabetu łacińskiego (np. Y, Z), ich realizacje odpowiednimi małymi literami (np. y, z), a zbiory realizacji dużymi literami piśmianymi (np. \mathcal{Y}, \mathcal{Z}). Symbol $P(\mathcal{Y})$ oznacza prawdopodobieństwo zdarzenia \mathcal{Y} . Dla rozkładów zmiennych dyskretnych $P_Y(y) = P\{Y=y\}$.

Rozkłady warunkowe są oznaczone następująco:

$$P_{Y|Z}(y, z) = P\{Y=y|Z=z\}.$$

ELEMENTY BAYESOWSKIEJ TEORII DECYZJI DLA OCENY ALGORYTMÓW ZRANDOMIZOWANYCH

Rozważmy zadanie programowanie dyskretne

$$\max \{f(x) \mid x \in X\} \quad (P)$$

należące do dobrze określonej klasy zadań \mathcal{K} .

Założmy, że $x \in \mathcal{X}^n$ i $f: \mathcal{X}^n \rightarrow \mathcal{Z}$ gdzie \mathcal{Z} oznacza zbiór liczb całkowitych. Przyjmijmy, że $X \neq \emptyset$ i że dysponujemy algorytmem zrandomizowanym A do rozwiązywania zadań z klasy \mathcal{K} .

Rozważmy następującą sytuację decyzyjną:
Stosując algorytm A generujemy ciąg k (k jest wcześniej przyjętą liczbą) rozwiązań zadania (P)

$$x_1, x_2, \dots, x_k \quad (8)$$

i odpowiadający mu ciąg wartości rozwiązań

$$v_1, v_2, \dots, v_k \quad (9)$$

gdzie $v_i = f(x_i)$, $i = 1, \dots, k$.

Liczby v_i , $i = 1, \dots, k$, można traktować jako realizacje zmiennej losowej V o (nieznanym) rozkładzie $P_V(v) = P\{V = v\}$, $v \in \mathbb{Z}$. Ponieważ generowanie poszczególnych rozwiązań wykonywane jest w sposób niezależny od siebie, możemy przyjąć, że dysponujemy orobą losową złożoną z k niezależnych obserwacji. Na podstawie tej próby i posiadanych a priori danych o klasie zadań i algorytmie A należy podjąć decyzję co do kontynuowania próby, a w orzypadku zatrzymania algorytmu należy oszacować wartość optymalną W rozwiązania zadania P. Problem sprowadza się więc do określenia kryterium stopu dla sekwencji zastosowań algorytmu A oraz podania sposobu estymacji parametru W .

Załóżmy, że wartość optymalna W rozwiązania zadania (P) jest zmienną losową. Założenie to jest podyktowane z jednej strony technicznymi wymogami stosowalności podejścia bayesowskiego, a z drugiej strony może być uzasadnione tym, że zadanie (P) jest wybierane w sposób w pewnym sensie orzypadkowy z klasy zadań \mathcal{K} . Rozkład prawdopodobieństwa P_W zmiennej W (tak zwany rozkład a priori) można założyć na podstawie znajomości klasy zadań oraz przeprowadzonej analizy przedoptymalizacyjnej. Rozwiązujący zadanie dysponuje zwykle pewnymi informacjami dotyczącymi wartości optymalnej zadania, jej położenia w stosunku do wartości łatwo rozwiązywalnej relaksacji zadania (P) itp. Wynikiem analizy przedoptymalizacyjnej są zwykle oszacowania \bar{W} i \underline{W} wartości W odpowiednio od góry i od dołu. Można więc przyjąć, że zbiór \mathcal{K}^* realizacji zmiennej W jest określony następująco:

$$\mathcal{K}^* = \left\{ w \in \mathbb{Z} : \underline{W} \leq w \leq \bar{W} \right\} \quad (10)$$

Załóżmy na przykład, że rozwiązujemy zadanie należące do klasy wielowymiarowych zadań załadunku, występujących przy bilansowaniu nakładów inwestycyjnych i na podstawie doświadczeń z rozwiązywaniem zadań tej klasy mamy informacje następujące: dla znacznej większości (80%) zadań rozwiązanie optymalne zadania różni się o mniej niż (15%) od wartości optymalnej ciągłej relaksacji zadania. Wynikiem analizy przedoptymalizacyjnej mogą być na przykład: oszacowanie wartości optymalnej zadania od góry,

otrzymane przez rozwiązanie ciągłej relaksacji zadania, oszacowanie od dołu, wyznaczone przez znalezienie pewnego dopuszczalnego rozwiązania, np. algorytmem typu GREEDY. Dane takie i intuicja rozwiązującego zadanie są podstawą do przyjęcia rozkładu a priori zmiennej W .

Szczególnym przypadkiem takiego rozkładu jest rozkład równomierny na zbiorze \mathcal{N} , którego przyjęcie może być w niektórych przypadkach spowodowane brakiem dostatecznej ilości informacji wstępnych.

Następnym elementem, który należy założyć, jest rozkład błędów wnoszonych przez algorytm zrandomizowany. Ta charakterystyka algorytmu może być wprowadzona w różny sposób. Jedną z możliwości jest założenie rozkładu warunkowego $P_{V|W}$. Inna możliwość polega na podaniu rozkładu błędu $B = 1 - V/W$, przy założeniu, że procentowy błąd wnoszony przez algorytm nie zależy od wartości W . W dalszych rozważaniach będziemy się posługiwać rozkładem warunkowym $P_{V|W}$.

Rozważmy uproszczoną sytuację decyzyjną polegającą na tym, że dysponując realizacją z próby losowej (gdzie $z = (v_1, \dots, v_k)$) należy estymować wartość rozwiązania optymalnego zadania P . Niech $l(w^*, W)$ będzie funkcją strat wynikających z przyjęcia zamiast rzeczywistej wartości rozwiązania optymalnego W - wartości w^* . Można na przykład wziąć $l = l'$, gdzie

$$l'(w^*, W) = a(w^* - W)^2, \quad (11)$$

przy czym $a > 0$ jest pewną stałą.

Rozwiązanie w^* można traktować jako wartość funkcji decyzyjnej $d: Z^k \rightarrow Z$ dla wyniku próby z . Mamy

$$w^* = d(z). \quad (12)$$

Oznaczmy symbolem D zbiór możliwych funkcji decyzyjnych.

Dla danej funkcji decyzyjnej d strata $l(d(Z), W)$ jest funkcją dwóch zmiennych losowych: wyniku próby $Z = (v_1, \dots, v_k)$ i wartości rozwiązania W . Uśredniając tę stratę po możliwych próbach, otrzymujemy funkcję ryzyka r

$$r(d, W) = \sum_{z \in Z} l(d(z), W) P_{Z|W}(z, W). \quad (13)$$

Uśredniając funkcję ryzyka po W otrzymujemy ryzyko bayesowskie

$$R(d) = \sum_{w \in \mathcal{N}} r(d, w) P_W(w). \quad (14)$$

Funkcja $R: D \rightarrow R$ wprowadza porządek w zbiorze funkcji decyzyjnych D . Najlepszą w sensie wartości ryzyka bayesowskiego funkcją decyzyjną jest funkcja d^* , dla której

$$R(d^*) = \min_{d \in D} R(d) \quad (15)$$

Podstawiając w (14) wyrażenie na $r(d, W)$ z (13), otrzymujemy

$$R(d) = \sum_{w \in \mathcal{W}} \sum_{z \in \mathcal{Z}} l(d(z), w) P_{Z|W}(z, w) P_W(w). \quad (16)$$

Z twierdzenia Bayesa

$$P_{Z|W}(z, w) = P_{W|Z}(w, z) \frac{P_Z(z)}{P_W(w)}, \quad (17)$$

gdzie prawdopodobieństwa warunkowe $P_{W|Z}$ noszą nazwę prawdopodobieństw a posteriori.

Podstawiając to wyrażenie do (16) i zmieniając kolejność sumowania otrzymujemy

$$R(d) = \sum_{z \in \mathcal{Z}} \left(\sum_{w \in \mathcal{W}} l(d(z), w) P_{W|Z}(w, z) \right) P_Z(z). \quad (18)$$

Ponieważ $P_Z(z) \geq 0$ dla każdego $z \in \mathcal{Z}$, zatem funkcja d minimalizuje wyrażenie (18), jeśli dla każdego $z \in \mathcal{Z}$ minimalizuje wyrażenie

$$L(d, z) = \sum_{w \in \mathcal{W}} l(d(z), w) P_{W|Z}(w, z), \quad (19)$$

to znaczy

$$d^* = \arg \min_{d \in D} L(d, z). \quad (20)$$

Zauważmy, że $L(d, z)$ jest wartością średnią strat $l(d(z), w)$ liczoną względem rozkładu a posteriori $P_{W|Z}$.

W przypadku, gdy $l(d(z), w) = a|d(z) - w|^2$ wyznaczenie funkcji d^* jest szczególnie proste, bowiem wyrażenie $E(d(z) - W)^2$ jest minimalizowane dla $d(z) = E(W)$. A zatem w tym przypadku

$$w^* = d^*(z) = \sum_{w \in \mathcal{W}} w P_{W|Z}(w, z). \quad (21)$$

W przypadku innych funkcji strat, rozwiązania zadania (20) mogą być inne. Jeśli na przykład

$$l = l''(d(z), w) = a|d(z) - w|, \quad (22)$$

gdzie $|x|$ oznacza wartość bezwzględną x , wówczas optymalnym rozwiązaniem jest wybranie jako funkcji decyzyjnej takiej funkcji d , która zaobserwowanej próbie z przyporządkuje jako w^* wartość mediany rozkładu a posteriori $P_{W|Z}$.

Z twierdzenia Bayesa rozkład a posteriori $P_{W|Z}$ wyraża się wzorem

$$P_{W|Z}(w, z) = P_{Z|W}(z, w) \frac{P_W(w)}{\sum_{w \in \mathcal{W}} P_{Z|W}(z, w) P_W(w)}. \quad (23)$$

Zakładając, że charakterystyka probabilistyczna algorytmu podawana jest w postaci rozkładu warunkowego $P_{V|W}$, dla realizacji $z = (v_1, \dots, v_k)$ próby $Z = (v_1, \dots, v_k)$ otrzymujemy

$$P_{Z|W}(z, w) = \prod_{i=1}^k P_{V|W}(v_i, w) \quad \text{dla } w \in \mathcal{W} \quad (24)$$

Jeśli przyjmowane a priori rozkłady prawdopodobieństw pamiętane są w postaci tablic, wówczas utworzenie tablicy prawdopodobieństw $P_{Z|W}(z, w)$ dla danego z wymaga $|\mathcal{W}|^k$ mnożeń. Dla obliczenia wartości $P_Z(z)$, gdzie

$$P_Z(z) = \sum_{w \in \mathcal{W}} P_{Z|W}(z, w) P_W(w), \quad (25)$$

potrzeba dodatkowo $|\mathcal{W}|$ mnożeń i $|\mathcal{W}|$ dodawań. Zatem obliczenie elementów tablicy $P_{W|Z}$ dla każdego $w \in \mathcal{W}$ i danego z wymaga $|\mathcal{W}|^k (k+2)$ mnożeń, $|\mathcal{W}|^k$ dodawań, i $|\mathcal{W}|^k$ dzieleni, a dla wyznaczenia wartości w^* według wzoru (21) potrzeba dodatkowo $|\mathcal{W}|^k$ mnożeń i $|\mathcal{W}|^k$ dodawań. Łącznie więc obliczenie w^* według zależności (21) wymaga $(k+3) |\mathcal{W}|^k$ mnożeń
 $2 |\mathcal{W}|^k$ dodawań
 $|\mathcal{W}|^k$ dzieleni.

Jeśli interesuje nas tylko w^* , wówczas $|\mathcal{W}|^k$ - 1 dzieleni można wyeliminować. W rozważanej przez nas najprostszej sytuacji decyzyjnej obliczenie wartości w^* kończy ocenę rozwiązania. Informacje, którymi dysponujemy po zakończeniu obliczeń, są następujące:

a) Znamy wartość \bar{w} najlepszego rozwiązania wygenerowanego w próbie, gdzie

$$\bar{w} = \max \{v_i, \quad i = 1, \dots, k\}. \quad (26)$$

Ponadto znamy odpowiadające \bar{w} rozwiązanie zadania (P).

- b) Na podstawie analizy próby znamy również bayesowskie oszacowanie w^* wartości rozwiązania optymalnego. W ogólnym przypadku nie znamy tu samego rozwiązania, które odpowiada wartości w^* , a nawet rozwiązanie takie może nie istnieć.

Zapiszmy schematycznie powyższy model decyzyjny.

Model decyzyjny D1

- 1^o Ustal k (liczebność próby).
- 2^o Wygeneruj k rozwiązań, stosując algorytm A i oblicz \bar{w} według (26).
- 3^o Oblicz wartość $w^* = d^*(z)$.
- 4^o STOP: najlepiej znane rozwiązanie ma wartość \bar{w} , a oszacowanie wartości rozwiązania optymalnego jest równe w^* .

Najprostszym rozszerzeniem powyższego modelu decyzyjnego jest dodatkowe wykorzystanie wartości w^* i \bar{w} w kryterium stopu dla sekwencji zastosowań algorytmu A. Spowoduje to modyfikację kroku 4^o w modelu D1. Na przykład:

Model decyzyjny D2

- 1^o, 2^o, 3^o jak w modelu D1.
 - 4^o Jeśli $s(w^*, \bar{w}, k) \leq 0$, to STOP;
- w przeciwnym przypadku wygeneruj następne k' rozwiązań, stosując algorytm A i podstaw $k: k + k'$. Oblicz \bar{w} i idź do 3^o.

Funkcja s może mieć różną postać. Na przykład można przyjąć, że

$$s(w^*, \bar{w}, k) = \varepsilon w^* - \bar{w}, \quad (27)$$

gdzie $\varepsilon \leq 1$ jest pewną stałą, którą można interpretować jako akceptowalną dokładność otrzymanego rozwiązania w stosunku do estymowanej wartości optymalnej. Stałą ε można również dobierać w różny sposób. W szczególności jeśli $k' = k$, gdzie k jest stałą dobieraną w kroku 1^o, wówczas sekwencja rozwiązań generowanych przez algorytm składa się z bloków o stałej długości.

Szczególny przypadek kryterium stopu otrzymujemy, jeśli funkcja s uwzględnia koszty związane z generacją kolejnych rozwiązań i porównuje je z zyskami, które wiążą się z możliwością uzyskania rozwiązania lepszego niż to, którym dotychczas dysponujemy. W takim przypadku kryterium stopu może być zależne nie tylko od aktualnej wartości parametru k oraz w^* i \bar{w} , ale również od rozkładów prawdopodobieństw opisujących algorytm i zadanie.

UWAGI KOŃCOWE

Przedstawione w pracy podejście do oceny efektywności algorytmów zrandomizowanych daje pewną technikę, która może być użyta dla potwierdzenia intuicji rozwiązującego zadanie o dobrych wynikach działania tego typu algorytmów. Wprowadzenie rozkładów a priori pozwala na włączenie informacji o klasie zadań i użytym algorytmie w proces rozwiązywania i oceny rozwiązania.

W pracy zaproponowano użycie prostych funkcji strat, dla których postać rozwiązania zadania pomocniczego jest z góry znana. Równocześnie korzysta się w istotny sposób z dyskretności zadania; a poszczególne rozkłady prawdopodobieństwa są pamiętane w postaci tablic. Dzięki temu sprawdzenie testu stopu algorytmu nie jest pracochłonne (liczba elementarnych działań jest rzędu $k |N^k|$). Odróżnia to w istotny sposób prezentowane podejście od idei zawartych w [6] i [1], gdzie test stopu wymaga bardzo znacznego nakładu obliczeń w stosunku do ilości obliczeń niezbędnych do uzyskania jednego elementu próby Z . Ponadto użycie innych postaci funkcji decyzyjnych prowadzi do konieczności rozwiązywania zadania (20), co w wielu przypadkach może być skomplikowane.

Użycie w obliczeniach zapisu rozkładów prawdopodobieństwa w postaci tablic nie oznacza oczywiście, że żąda się tej samej postaci danych od użytkownika, który określa a priori probabilistyczne charakterystyki klasy zadań \mathcal{K} i algorytmu A . Dla użytkownika może być prostsze (choć formalnie niezbyt poprawno) zadanie rozkładów wartości optymalnej rozwiązania W i rozkładu błędów wnoszonych przez algorytm w postaci rozkładów ciągłych poprzez odpowiednie określenie rodziny rozkładów (np. rozkładów beta) i wyseparowanie niezbędnych parametrów.

Czynnością poprzedzającą właściwe rozwiązywanie zadania byłoby przetransformowanie tych informacji do żądanej tablicowej postaci. Dopuszczenie tego rodzaju nieformalności wyniku z realistycznego, jak się wydaje założenia, że dane jakie będzie można uzyskać od użytkownika, będą dotyczyć raczej charakteru rozkładów, natomiast rzadko będzie można otrzymać dokładne wartości prawdopodobieństw.

LITERATURA

- [1] Botrò B.: Bayesian testing of nonparametric hypotheses and its application to global optimization problems. Istituto per Applicazioni della Matematica e della Informatica - CNR, Milano 1981.
- [2] Kelly D.G.: Some results on random linear programs. Praca z V Symposium über Operations Research, Köln, 25-27.VII.1981.
- [3] Klee V., Minty G.J.: How good is the simplex algorithm. W Inequalities II, ed. O. Shisha, Academic Press, 1971.
- [4] Korte B., Hauemann D.: Analysis of the greedy heuristics for independence systems. Annals of Discrete Mathematics 2, 1978, s. 65-74.

- [5] Lindgren B.W.: Elements of decision theory. New York, 1971.
- [6] Vercellis C.: A probabilistic stopping rule for randomized approximation algorithms. Istituto di Matematica, Università di Milano June, 1981. Praca przedstawiona na SIAM Conference on the Applications of Discrete Mathematics, Troy, June 11-12, 1981.
- [7] Weide B.W.: Statistical methods in algorithm design and analysis. Ph.D. Thesis, Carnegie - Mellon University, Pittsburg 1978, CMU-CS-78-142.
- [8] Zieliński R.: A Monte-Carlo method for seeking minima in multiextremal problems. Università di Milano, maj 1980, nr 48/S (praca złożona do Mathematical Programming).
- [9] Garey M.R., Johnson D.S.: Computers and intractability. A guide to the theory of NP-completeness, W.H. Freeman and Company, 1979, San Francisco.

Recenzent: Doc. dr hab. inż. Andrzej GOŚCIŃSKI

Wpłynęło do Redakcji 15.05.1982 r.

О ОЦЕНКЕ РАБОТЫ РАНДОМИЗИРОВАННЫХ АЛГОРИТМОВ

Р е з ю м е

В работе представлен Байесовый подход к оценке рандомизированных алгоритмов. Даются примеры алгоритмов такого типа и рассматриваются их преимущества. Приведены некоторые элементы Байесовой теории принятия решений и изложены её применения к определению критерия останова работы алгоритма и оценки его точности.

ON THE EVALUATION OF EFFICIENCY OF RANDOMIZED ALGORITHMS

S u m m a r y

In the paper the bayesian approach to evaluation of the efficiency of randomized algorithms is presented. Simple examples of randomized algorithms for integer programming problems are given and their advantages are discussed. Some elements of bayesian decision theory are briefly described and a possibility of its application for the construction of stop criteria as well as for evaluation of the accuracy of randomized algorithm is considered.