УДК 519.854

С. В. Чупов(Ужгородський нац. ун-т)

СТОХАСТИЧНИЙ АЛГОРИТМ ЛЕКСИКОГРАФІЧНОГО ПОШУКУ ДЛЯ ЗАДАЧІ ДИСКРЕТНОГО ПРОГРАМУВАННЯ.

The article deals with a new scheme of algorithm of lexicographical search of the optimal solution to the of discrete programming problem, based on the determination of the probability that while moving in the given lexicographical order in the investigated subset of the set of feasible solutions to the problem the best solutions will be received. The structural scheme of the new method is being presented, its justification is conducted.

В статье рассматривается новая схема алгоритма лексикографического поиска оптимального решения задачи дискретного программирования, основанная на определении вероятности того что при движении в данном лексикографическом порядке в исследуемом подмножестве множества допустимых решений задачи будут получены лучшие решения. Представлена структурная схема нового метода, проводится его обоснование.

1. Вступ. Розглядається задача дискретного програмування:
максимізувати
$$x_0 \ = \ f_0\left(x\right), \tag{1}$$
за умов
$$x \in X^D, \tag{2}$$
де $X^D = D \cap X$, $X \subseteq R^n$, $D \subset P$ — дискретна множина, $P = \{x \in R^n \,|\bar{0} \leq x \leq u\}$ — n - вимірний гіперпаралелепіпед. Існує багато точних та наближених методів розв‘язання даної задачі [4].

На основі поняття лексикографічного максимуму множини [3, 5] процес відшукання оптимального розв‘язку задачі (1),(2) можна звести до відшукання лексикографічних максимумів послідовності множин $X^0, X^1, ..., X^k, ...$, де $X^0 = X^D$, $X^k = \left\{x \in X^D \middle|\, x \leq^L x^{k-1}, f_0(x) > x_0^{k-1}\right\}$, $k = 1, 2, \ldots$, $x^k = \max^L X^k$, $x_0^k = f_0(x^k)$, $k = 0, 1, \ldots$ [5]. Якщо $X^0 \neq \emptyset$, то в результаті застосування стандартного алгоритму лексикографічного пошуку [2] будується лексикографічно спадна послідовність допустимих розв‘язків $x^0 >^L x^1 >^L ... >^L x^k >^L ...$, якій відповідає зростаюча послідовність $x_0^0 > x_0^1 > ... > x_0^k > ...$ значень цільової функції (1). Процес розв‘язання задачі завершується як тільки на деякому кроці $k+1\,(k \geq 0)$, виявиться, що $X^{k+1} = \emptyset$. В цьому разі оптимальним розв‘язком задачі (1),(2) буде точка x^k, отримана на попередньому кроці.

2. Основний результат. Розглянемо більш детально алгоритм пошуку лексикографічного максимуму множини X^k, тобто алгоритм відшукання розв‘язку $x^k = max^L X^k$, $k = 1, 2, \ldots$. Для цього визначимо множину
$$\bar{X}^D\left(\hat{x}_0, \tilde{x}, s\right) = \left\{x \in X^D \middle|\, f_0\left(x\right) > \hat{x}_0, \bar{x} \leq^L x \leq^L \tilde{x}\right\}, \tag{3}$$
де $\hat{x} \in X^D$ — найкращий серед отриманих на даний момент розв‘язків, $\hat{x}_0 = f_0\left(\hat{x}\right)$ — значення цільової функції в точці $\hat{x}$, $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_{s-1}, 0, u_{s+1}, \ldots, u_n) \in B^n$, $\bar{x} = \left(\tilde{x}_1, ..., \tilde{x}_{s-1}, 0, \underbrace{0, ..., 0}_{n-s}\right) \in B^n$, $\bar{x} \leq^L x \leq^L \tilde{x}$ — лексикографічні межі

пошуку, $s = 0, 1, \ldots, n-1$. Якщо $s = 0$, тоді будемо вважати, що $\tilde{x} = (u_1, \ldots, u_n)$, $\bar{x} = \underbrace{(0, \ldots, 0)}_{n}$. Потрібно визначити лексикографічний максимум множини (3) або встановити, що $\bar{X}^D(\hat{x}_0, \tilde{x}, s) = \emptyset$.

Припустимо, що для задачі (1),(2) визначений алгоритм пошуку лексикографічного максимуму множини $Y_t(y) = \{x \in Y | \, x_j = y_j, j = 1, \ldots, t-1, x_t = 0\}$, де $Y \subseteq B^n$. В процесі пошуку лексикографічного максимуму множини (3) потрібно на певних кроках визначати лексикографічні максимуми множин вигляду $\bar{X}(f', x') = \left\{x \in B^n \middle| \, f_0(x) > f', x \leq^L x'\right\}$ та $\bar{X}^D(x'') = \left\{x \in X^D \middle| \, x \leq^L x''\right\}$, де $f' \in R$, $x', x'' \in B^n$ — фіксовані значення. За теоремою 1 [1] існування лексикографічного максимуму множини $\bar{X}(f', x')$ випливає з існування максимального індексу $t \in \{1, 2, \ldots, n-1\}$ для якого множина

$$\bar{X}_t(f', x') = \left\{x \in B^n \middle| \, f_0(x) > f', x_j = x'_j, j = 1, .., t-1, x_t = 0\right\}$$

не порожня. Тоді, якщо $\bar{X}_t(f', x') \neq \emptyset$, $\max^L \bar{X}_t(f', x') = \max^L \bar{X}(f', x')$. Аналогічно для множини $\bar{X}^D(x'')$ існування максимального індексу t дає можливість визначити лексикографічний максимум цієї множини через визначення лексикографічного максимуму множини

$$\bar{X}_t^D(x'') = \left\{x \in X^D \middle| \, x_j = x''_j, j = 1, .., t-1, x_t = 0\right\}.$$

Не важко помітити, що для пошуку лексикографічних максимумів множин $\bar{X}_t(f', x')$ та $\bar{X}_t^D(x'')$ можна скористатися визначеним за припущенням алгоритмом пошуку для множини $Y_t(y)$. Для множини $\bar{X}_t(f', x')$ множина $Y = \left\{x \in B^n \middle| \, f_0(x) > f'\right\}$ та $y = x'$, для множини $\bar{X}_t^D(x'')$ множина $Y = X^D$, $y = x''$. Попередні визначення дають змогу в нових термінах описати загальну схему точного алгоритму пошуку лексикографічного максимуму множини (3).

Алгоритм $AStdLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$.
Крок 0.

Визначимо лексикографічний максимум множини $\bar{X}^D(\tilde{x})$. Якщо $\bar{X}^D(\tilde{x}) = \emptyset$, тоді припиняємо подальші обчислення за відсутністю допустимих точок у множині $\bar{X}^D(\tilde{x})$. Інакше отримаємо $x^0 = \max^L \bar{X}^D(\tilde{x})$. Якщо $x^0 <^L \bar{x}$ тоді також припиняємо обчислення, оскільки x^0 не належить лексикографічному проміжку пошуку, а отже $\bar{X}^D(\hat{x}_0, \tilde{x}, s) = \emptyset$. Якщо ж $\bar{x} \leq^L x^0 \leq^L \tilde{x}$, тоді визначаємо значення $f_0^0 = f_0(x^0)$ і переходимо до першого кроку алгоритму.
Крок $k, (k > 0)$.

На початку кожного кроку покладаємо $y^0 = x^{k-1}$, $f_0^{k-1} = f_0\left(x^{k-1}\right)$ та переходимо до першого етапу кроку k.
Етап $r, (r > 0)$.

Визначимо $z^r = \max^L \bar{X}\left(f_0^{k-1}, y^{r-1}\right)$. Для цього визначаємо максимальний індекс l_r для якого множина $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right)$ не порожня. Якщо індекс l_r не існує або $l_r \leq s$, тоді це означає, що множина $\left\{x \in X^D \middle| \, f_0(x) > f_0^{k-1}, \bar{x} \leq^L x \leq^L y^{r-1}\right\}$ — порожня і таким чином лексикографічним максимумом множини (3) буде точка x^{k-1}. Якщо ж індекс l_r вдасться визначити, тоді знайдемо точку $z^r = \max^L \bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) = \max^L \bar{X}\left(f_0^{k-1}, y^{r-1}\right)$. Зауважимо, що, як правило, процедура з'ясування існування індексу l_r полягає в одночасному визначенні точки z^r. Дана процедура представляє собою цикл. Робота циклу починається із значення $l_r = n$. На кожному кроці циклу, якщо $y_{l_r}^{r-1} = 0$ або $y_{l_r}^{r-1} > 0$ та $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) = \emptyset$, тоді значення l_r зменшуємо на одиницю та переходимо до наступного кроку циклу. Цикл завершується як тільки отримаємо, що $l_r \leq s$ або $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) \neq \emptyset$ за умови, що $y_{l_r}^{r-1} > 0$.

Якщо $z^r \notin X^D$, тоді визначимо $y^r = \max^L \bar{X}^D(z^r)$. Для цього визначимо лексикографічний максимум множини $\bar{X}_{p_r}^D(z^r)$, де $l_r < p_r \leq n$ — максимальний індекс для якого множина $\bar{X}_{p_r}^D(z^r)$ не порожня. Отримаємо $y^r = max^L \bar{X}_{p_r}^D(z^r)$ $= \max^L \bar{X}^D(z^r)$. З визначення множини $\bar{X}_{p_r}^D(z^r)$ випливає, що $y^r \in X^D$, $y^r <^L z^r$ але не обов'язково $f_0(y^r) > f_0^{k-1}$. Тому, якщо $f_0(y^r) > f_0^{k-1}$, тоді $x^k = y^r$ і на цьому k-й крок завершується. В супротивному випадку переходимо до наступного етапу алгоритму.

Скінченність алгоритму випливає з побудови на кожному кроці лексикографічно спадної послідовності точок $y^0 >^L z^1 >^L y^1 >^L ... >^L y^{r-1} >^L z^r >^L ...$ яка обмежена знизу точкою $\bar{x}$. Точність алгоритму випливає з того, що тільки тоді коли на черговому етапі r множина $\left\{x \in X^D \middle| f_0(x) > f_0^{k-1}, y^r \leq^L x \leq^L y^{r-1}\right\}$ є порожньою тільки тоді буде здійснено перехід до наступного етапу алгоритму, тобто лексикографічний максимум множини (3) буде однозначно знайдено у разі не порожності цієї множини.

Якщо правило вибору максимального індексу l_r на кожному з етапів алгоритму $AStdLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$ не обмежувати тільки правилом $y_{l_r}^{r-1} > 0 \wedge \bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) \neq \emptyset$, а додати ще деякі обмеження на цей вибір, тоді з'являється можливість у побудові нових схем пошуку лексикографічного максимуму множини (3) для яких кількість точок послідовності $y^0 >^L z^1 >^L y^1 >^L \ldots$ буде значно меншою у порівнянні з відповідною кількістю, що отримується за загальною схемою точного алгоритму пошуку лексикографічного максимуму множини (3). Але слід зауважити, що при використанні додаткових обмежень на вибір індексу l_r умови теореми 1 [1] будуть порушуватись. Тобто гарантії того, що $\max^L \bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) = \max^L \bar{X}\left(f_0^{k-1}, y^{r-1}\right)$ не буде і можливо, що на черговому етапі r виявиться $z^r = \max^L \bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) <^L \max^L \bar{X}\left(f_0^{k-1}, y^{r-1}\right)$ але при цьому $x^k = \max^L \bar{X}\left(f_0^{k-1}, y^{r-1}\right) \in X^D$ і тоді $z^r <^L x^k$. Іншими словами буде пропущений допустимий розв'язок задачі для якого $f_0\left(x^k\right) > f_0^{k-1}$.

Нехай при відшуканні оптимального розв'язку задачі (1), (2) за алгоритмом лексикографічного пошуку $AStdLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$ для кожної координати $t \in \{1, 2, \ldots, n-1\}$ фіксується частотний розподіл значень цільової функції (1) які вважаються випадковими величинами та на основі цього визначається імовірнісний m_k - модальний розподіл значень цільової функції (1) за координатою t з функцією розподілу [1]:

$$\tilde{F}_{X,t}(f) = P(x_0(\omega) \leq f) = \\ \sum_{k=1}^{m_k} \alpha_k \left(\tilde{P}_{y_{k-1}} + \frac{\tilde{P}_{y_k} - \tilde{P}_{y_{k-1}}}{F_{X,t}^k(y_k) - F_{X,t}^k(y_{k-1})} \left(F_{X,t}^k(f) - F_{X,t}^k(y_{k-1}) \right) \right) \quad (4)$$

де α_k, $\tilde{P}_{y_k}$, $k = 1, 2, \ldots, m_k$ — параметри розподілу, $F_{X,t}^k(f)$, $k = 1, 2, \ldots, m_k$ — наперед визначені функції розподілу випадкової величини $x_0(\omega)$.

Функція розподілу (4) дозволяє визначати імовірність того, що після виконання певної кількості етапів за алгоритмом лексикографічного пошуку на наступному етапі пошуку за координатою $l_r = t$ випадкова величина значення цільової функції $x_0(\omega)$ не перевищить значення f.

Використовуючи функцію розподілу (4) побудуємо стохастичний алгоритм пошуку лексикографічного максимуму множини (3). Припустимо, що для кожного $t \in \{1, 2, \ldots, n-1\}$ визначені значення f_t^{lim} та ε_t, де f_t^{lim} — граничне значення цільової функції яке відділяє "гарні" значення від "поганих" значень, $0 \leq \varepsilon_t \leq 1$ — точність такого розділення. Загальна структура алгоритму стохастичного лексикографічного пошуку є такою самою як в алгоритмі $AStdLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$. Відмінність буде полягати лише у виборі індексу l_r та уточненні $\tilde{F}_{X,l_r}(f)$. Тому опишемо тільки загальну схему етапу r.

Алгоритм $APrLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$.
Етап $r, (r > 0)$.

Покладаємо $l_r = n-1$ та розпочинаємо цикл пошуку індексу l_r. Якщо $y_{l_r}^{r-1} = 0$, тоді $l_r = l_r - 1$ і переходимо до наступного кроку циклу l_r. Якщо $y_{l_r}^{r-1} > 0$, обчислюємо значення імовірності $p_{l_r} = 1 - \tilde{F}_{X,l_r}\left(f_{l_r}^{\lim}\right)$ і якщо $p_{l_r} < \varepsilon_{l_r}$, тоді $l_r = l_r - 1$ і на цьому поточний крок циклу завершуємо. Якщо $p_{l_r} \geq \varepsilon_{l_r}$ та $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) = \emptyset$, тоді $l_r = l_r - 1$ і теж переходимо до наступного кроку циклу. Якщо ж $p_{l_r} \geq \varepsilon_{l_r}$ та $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right) \neq \emptyset$, тоді фіксуємо значення l_r та виходимо з циклу. Якщо $l_r \leq s$, тоді цикл пошуку індексу l_r завершується. В цьому разі множина $\left\{x \in X^D \middle| f_0(x) > f_0^{k-1}, \bar{x} \leq^L x \leq^L y^{r-1}\right\}$ — імовірно порожня і таким чином лексикографічним максимумом множини (3) імовірно буде точка x^{k-1}.

Якщо індекс l_r визначений, тоді знайдемо точку $z^r = \max^L \bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right)$. З визначення множини $\bar{X}_{l_r}\left(f_0^{k-1}, y^{r-1}\right)$ випливає, що завжди $f_0(z^r) > f_0^{k-1}$ та $z^r <^L y^{r-1}$ але не обов'язково z^r — допустима точка. Тому, якщо $z^r \in X^D$, тоді отримаємо $x^k = z^r$, уточнимо параметри функції розподілу $\tilde{F}_{X,l_r}(f)$ і на цьому k-й крок завершується.

Якщо $z^r \notin X^D$, тоді визначимо $y^r = \max^L \bar{X}^D(z^r)$. Отримавши значення $f_0(y^r)$ уточнюємо параметри функції розподілу $\tilde{F}_{X,l_r}(f)$. З визначення множини $\bar{X}_{p_r}^D(z^r)$ випливає, що $y^r \in X^D$, $y^r <^L z^r$ але не обов'язково $f_0(y^r) > f_0^{k-1}$. Тому, якщо $f_0(y^r) > f_0^{k-1}$, тоді $x^k = y^r$ і на цьому k-й крок завершується. Якщо ні, тоді переходимо до наступного етапу алгоритму.

Скінченність алгоритму $APrLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$ також випливає з побудови на кожному кроці лексикографічно спадної послідовності точок яка обмежена знизу точкою $\bar{x}$.

Конструктивно загальна схема етапу r алгоритму $APrLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$ представлена на рис. 1.

На кожному етапі r алгоритму $APrLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$ розпочинається аналіз множини $\bar{X}_t^D(y^{r-1})$, де $l_r = t$, який полягає у відповіді на запитання: чи містить множина $\bar{X}_t^D(y^{r-1})$ точки значення цільової функції в яких більше ніж f_0^{k-1}. Або чи є множина $\bar{X}_t^D(y^{r-1}) \cap \bar{X}_t\left(f_0^{k-1}, y^{r-1}\right)$ не порожньою. Однозначно дати відповідь на це запитання можливо лише після повного перегляду множини $\bar{X}_t^D(y^{r-1})$, наприклад, як це робиться в алгоритмі $AStdLexMax\left(\bar{X}^D(\hat{x}_0, \tilde{x}, s)\right)$. Тому на початку аналізу імовірність того, що множина $\bar{X}_t^D(y^{r-1}) \cap \bar{X}_t\left(f_0^{k-1}, y^{r-1}\right)$ — порожня чи не порожня є однаковою.

Визначимо подію B яка полягає у тому, що множина $\bar{X}_t^D(y^{r-1}) \cap \bar{X}_t\left(f_0^{k-1}, y^{r-1}\right)$

```
APrLexMax_Etap(r > 0, f_0^{k-1} = f_0(x^{k-1}), y^{r-1} ∈ X^D){
l_r = n − 1;
while(l_r > s){
  if(y_{l_r}^{r-1} > 0){
    p_{l_r} = 1 − F̃_{X,l_r}(f_{l_r}^{lim});
    if(p_{l_r} ≥ ε_{l_r} ∧ X̄_{l_r}(f_0^{k-1}, y^{r-1}) ≠ ∅)
      z^r = max^L X̄_{l_r}(f_0^{k-1}, y^{r-1});
      break;
    }
  }
  l_r = l_r − 1;
}
if(l_r ≤ s) return (−1);
if(z^r ∈ X^D){
  Update(F̃_{X,l_r}, f_0(z^r));
  x^k = z^r; return 0;
}
y^r = max^L X̄_{l_r}^D(z^r);
Update(F̃_{X,l_r}, f_0(y^r));
if(f_0(y^r) > f_0^{k-1}){ x^k = y^r; return 0; }
return 1;
}
```

Рис. 1. Конструктивна схема етапу r алгоритму $APrLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$

— не порожня . При цьому імовірність здійснення події $P\left(B\right)=\frac{1}{2}$. Зрозуміло, також, що $P\left(\bar{B}\right)=\frac{1}{2}$, де $\bar{B}$ — подія яка протилежна до події B. Нехай подія C полягає у тому, що на черговому етапі алгоритму $APrLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$ використання індексу $l_r=t$ — заборонено. Іншими словами, аналіз множини $\bar{X}_t^D\left(y^{r-1}\right)$ здійснюватись не буде. Оскільки заборона або дозвіл аналізу множини $\bar{X}_t^D\left(y^{r-1}\right)$ відбувається на підставі виконання нерівності $1-\tilde{F}_{X,t}\left(f_t^{\lim}\right)=p_t\geq\varepsilon_t$, тоді подія C здійсниться коли $\tilde{F}_{X,t}\left(f_t^{\lim}\right)=\bar{p}_t>1-\varepsilon_t$. Значення $1-\varepsilon_t$ є точною нижньою гранню при якому подія C виникає. Тому можна вважати, що імовірність здійснення події С дорівнює $1-\varepsilon_t$, $P\left(C\right)=1-\varepsilon_t$ та $P\left(\bar{C}\right)=\varepsilon_t$, де $\bar{C}$ — протилежна до C подія. Події B та C ніяк не впливають одна на одну, тобто вони є незалежними. Визначимо подію A, яка полягає в тому, що при забороні подальшого аналізу множини $\bar{X}_t^D\left(y^{r-1}\right)$ вона все ж містить точки в яких значення цільової функції більше ніж f_0^{k-1}, тобто події B та C здійсняться одночасно. Тоді $P\left(A\right)=P\left(B\right)P\left(C\right)=\dfrac{1-\varepsilon_t}{2}$, крім того $P\left(\bar{A}\right)=$

$P\left(\bar{B}\right)P\left(C\right)+P\left(B\right)P\left(\bar{C}\right)+P\left(\bar{B}\right)P\left(\bar{C}\right)=\dfrac{1+\varepsilon_t}{2}$. Не важко помітити, що при $\varepsilon_t \to 1$ $P\left(A\right)\to 0$, тобто при збільшенні ε_t імовірність здійснення події C зростає і коли $\varepsilon_t = 1$ аналіз усіх множин $\bar{X}_t^D\left(y^{r-1}\right)$ буде заборонений. При $\varepsilon_t \to 0$ $P\left(A\right)\to \dfrac{1}{2}$, тобто при зменшенні ε_t імовірність здійснення події C зменшується і коли $\varepsilon_t = 0$ аналіз усіх множин $\bar{X}_t^D\left(y^{r-1}\right)$ буде дозволений, що приводить до алгоритму $AStdLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$.

Визначимо випадкову величину W_r, яка рівна 1, якщо на етапі r подія A здійснилася і 0 – в протилежному випадку. Тоді імовірність того, що після виконання n_t етапів аналіз множин $\bar{X}_t^D\left(y^{r-1}\right)$ було m_t разів помилково заборонено визначається як:

$$P\left(\sum_{r=1}^{n_t} W_r = m_t\right) = C_{n_t}^{m_t}\frac{\left(1-\varepsilon_t\right)^{m_t}}{2^{m_t}}\frac{\left(1+\varepsilon_t\right)^{n_t-m_t}}{2^{n_t-m_t}} = \frac{C_{n_t}^{m_t}}{2^{n_t}}\left(1-\varepsilon_t\right)^{m_t}\left(1+\varepsilon_t\right)^{n_t-m_t}, \quad (5)$$

де $C_{n_t}^{m_t}$ — біноміальні коефіцієнти.

Теорема 1. *Якщо* $2m_t < n_t$ *та* $\varepsilon_t^* = \dfrac{n_t-2m_t}{n_t}$, *тоді імовірність* (5) *набуває максимального значення і при цьому* $\lim\limits_{n_t\to+\infty} P\left(\sum\limits_{r=1}^{n_t} W_r = m_t\right) = \dfrac{m_t^{m_t-1}}{\left(m_t-1\right)!}e^{-m_t}$.

Доведення. Визначимо значення ε_t при якому функція $h\left(e_t\right) = \left(1-\varepsilon_t\right)^{m_t}\times\left(1+\varepsilon_t\right)^{n_t-m_t}$ набуває максимального значення. Для цього знайдемо першу похідну функції $h\left(e_t\right)$, $\dfrac{dh}{d\varepsilon_t} = \left(1-\varepsilon_t\right)^{m_t-1}\times\left(1+\varepsilon_t\right)^{n_t-m_t-1}\times\left(n_t-2m_t-\varepsilon_t n_t\right)$ та розв‘яжемо рівняння $\dfrac{dh}{d\varepsilon_t} = 0$. Отримаємо три критичні точки $\varepsilon_t^* = \pm 1$ і $\varepsilon_t^* = \dfrac{n_t-2m_t}{n_t}$. Значення $\varepsilon_t^* = -1$ відкидаємо, оскільки $0 \le \varepsilon_t \le 1$ і воно не може бути від‘ємним. Хоча значення $\varepsilon_t^* = 1$ є точкою локального мінімуму функції $h\left(e_t\right)$ його також відкидаємо, оскільки імовірність здійснення події C $P\left(C\right) = 1-\varepsilon_t = 0$, тобто подія C — неможлива, а отже алгоритм $APrLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$ буде повністю проходити за схемою алгоритму $AStdLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$. Покажемо, що $\varepsilon_t^* = \dfrac{n_t-2m_t}{n_t}$ — точка максимуму функції $h\left(e_t\right)$. Знайдемо

$$\frac{d^2h}{d\varepsilon_t^2} = \left(1-\varepsilon_t\right)^{m_t-2}\times\left(1+\varepsilon_t\right)^{n_t-m_t-2}\times \times\left(\left(n_t^2+n_t\right)\varepsilon_t^2 - 2n_t\left(n_t-2m_t\right)\varepsilon_t + \left(n_t-2m_t\right)^2 - n_t\right).$$

Розв’язок нерівності $\dfrac{d^2h}{d\varepsilon_t^2} < 0$ визначає інтервал (6) на якому функція $\dfrac{d^2h}{d\varepsilon_t^2}\left(e_t\right)$

від‘ємна.

$$\frac{n_t-2m_t}{n_t+1}-\frac{\sqrt{n_t\left(n_t+4m_t\left(n_t-m_t\right)\right)}}{n_t^2+n_t}\leq\varepsilon_t\leq$$
$$\leq\frac{n_t-2m_t}{n_t+1}+\frac{\sqrt{n_t\left(n_t+4m_t\left(n_t-m_t\right)\right)}}{n_t^2+n_t}\quad(6)$$

Покажемо, що $\varepsilon_t^*=\dfrac{n_t-2m_t}{n_t}$ належить інтервалу (6). Враховуючи, що $2m_t<n_t$, отримаємо $\varepsilon_t^*-\dfrac{n_t-2m_t}{n_t+1}=\dfrac{n_t-2m_t}{n_t\left(n_t+1\right)}>0$. Отже ліва частина нерівності (6) задовольняється. Для того щоб показати, що задовольняється також і права частина нерівності (6) достатньо показати, що при $2m_t<n_t$ виконується нерівність $n_t-2m_t\leq\sqrt{n_t\left(n_t+4m_t\left(n_t-m_t\right)\right)}$. Дійсно $\left(n_t-2m_t\right)^2=n_t^2-4n_tm_t+4m_t^2\leq n_t\left(n_t+4m_t\left(n_t-m_t\right)\right)\Rightarrow 4m_t\left(m_t-n_t\right)\leq 4m_tn_t\left(n_t-m_t\right)\Rightarrow\left(m_t-n_t\right)\leq 0\leq n_t\left(n_t-m_t\right)$. Це доводить, що $\varepsilon_t^*=\dfrac{n_t-2m_t}{n_t}$ — точка максимуму функції $h\left(e_t\right)$, отже, при цьому значенні імовірність (5) набуває максимального значення.

Підставимо значення $\varepsilon_t^*=\dfrac{n_t-2m_t}{n_t}$ в (5). Враховуючи, що $1-\varepsilon_t^*=\dfrac{2m_t}{n_t}$ і $1+\varepsilon_t^*=\dfrac{2\left(n_t-m_t\right)}{n_t}$, перепишемо (5) наступним чином:

$$C_{n_t}^{m_t}\frac{m_t^{m_t}}{n_t^{m_t}}\frac{\left(n_t-m_t\right)^{n_t-m_t}}{n_t^{n_t-m_t}}=C_{n_t}^{m_t}\frac{m_t^{m_t}}{n_t^{m_t}}\left(1+\frac{\left(-m_t\right)}{n_t}\right)^{n_t}\left(1-\frac{m_t}{n_t}\right)^{-m_t}=$$
$$=\frac{\prod\limits_{i=1}^{m_t-1}\left(n_t-i\right)}{\left(m_t-1\right)!}\left(\frac{m_t}{n_t}\right)^{m_t-1}\left(1+\frac{\left(-m_t\right)}{n_t}\right)^{n_t}\left(1-\frac{m_t}{n_t}\right)^{-m_t}=$$
$$=\prod_{i=1}^{m_t-1}\left(1-\frac{i}{n_t}\right)\frac{m_t^{m_t-1}}{\left(m_t-1\right)!}\left(1+\frac{\left(-m_t\right)}{n_t}\right)^{n_t}\left(1-\frac{m_t}{n_t}\right)^{-m_t}.$$

Перейшовши до границі при $n_t\to+\infty$, отримаємо $\lim\limits_{n_t\to+\infty}P\left(\sum\limits_{r=1}^{n_t}W_r=m_t\right)=\dfrac{m_t^{m_t-1}}{\left(m_t-1\right)!}e^{-m_t}$. Доведення завершено.

Зауважимо, що якщо $2m_t=n_t$, тоді $\varepsilon_t^*=\dfrac{n_t-2m_t}{n_t}=0$ та імовірність здійснення події C $P\left(C\right)=1-\varepsilon_t^*=1$. При цьому не буде здійснено жодного повного етапу за алгоритмом $APrLexMax\left(\bar{X}^D\left(\hat{x}_0,\tilde{x},s\right)\right)$, тобто завжди буде $n_t=1$. Якщо $2m_t>n_t$, тоді $\varepsilon_t^*=\dfrac{n_t-2m_t}{n_t}<0$. Це означає, що на проміжку $0\leq\varepsilon_t\leq 1$ функція $h\left(e_t\right)$ має лише одну критичну точку $\varepsilon_t^*=1$ яка є точкою локального мінімуму.

Нехай $\bar{W}_r$ — випадкова величина яка рівна 1, якщо на етапі r подія A не здійснилася і 0 — в протилежному випадку. Визначимо імовірність

$$P\left(\sum_{r=1}^{n_t}\bar{W}_r=m_t\right)=\frac{C_{n_t}^{m_t}}{2^{n_t}}\left(1+\varepsilon_t\right)^{m_t}\left(1-\varepsilon_t\right)^{n_t-m_t}\quad(7)$$

того, що після виконання n_t етапів жодна із m_t відкинутих множин $\bar{X}_t^D\left(y^{r-1}\right)$ не буде містити розв'язку який за цільовою функцією кращий ніж x^{k-1}. Має місце наступна теорема:

Теорема 2. *Якщо* $2m_t > n_t$ *та* $\varepsilon_t^* = \dfrac{2m_t - n_t}{n_t}$*, тоді імовірність* (7) *набуває максимального значення і при цьому* $\lim\limits_{n_t \to +\infty} P\left(\sum\limits_{r=1}^{n_t} \bar{W}_r = m_t\right) = \dfrac{m_t^{m_t-1}}{(m_t-1)!}e^{-m_t}$.

Доведення. Доведення теореми здійснюється аналогічно до того як це робилося для теореми 1.

Не важко показати, що:

Наслідок 1. *На проміжку* $0 \le \varepsilon_t \le 1$ *значення імовірностей* (5) *та* (7) *монотонно спадають від* $\dfrac{C_{n_t}^{m_t}}{2^{n_t}}$ *до* 0 *при* $2m_t < n_t$ *та* $2m_t > n_t$ *відповідно.*

Припустимо, що максимальна кількість заборон не перевищує $\dfrac{n_t}{2}$ та ε_t обирається за умови виникнення найгіршої ситуації, тобто $\varepsilon_t = \dfrac{n_t - 2m_t}{n_t}$. Перш за все нас цікавить випадок коли імовірність помилок при забороні подальшого аналізу множин $\bar{X}_t^D\left(y^{r-1}\right)$ є мінімальною. Якщо $m_t = 1$, тоді отримаємо $P\left(\sum\limits_{r=1}^{n_t} W_r = 1\right)$ — імовірність того, що після виконання n_t етапів алгоритму $APrLexMax\left(\bar{X}^D\left(\hat{x}_0, \tilde{x}, s\right)\right)$ аналіз множини $\bar{X}_t^D\left(y^{r-1}\right)$ було один раз помилково заборонено. Зрозуміло, що якби цієї заборони не сталося, тоді був би знайдений лексикографічний максимум множини $\left\{x \in X^D \middle| f_0(x) > f_0^{k-1}, \bar{x} \le^L x \le^L y^{r-1}\right\}$. Якщо подальший аналіз множин $\bar{X}_t^D\left(y^{r-1}\right)$ буде заборонено, тоді може виникнути така ситуація, що хоча множина $\left\{x \in X^D \middle| f_0(x) > f_0^{k-1}, \bar{x} \le^L x \le^L y^{r-1}\right\}$ — не порожня, тобто лексикографічний максимум множини існує але він буде пропущений. Отже, при $m_t = 1$ маємо $\varepsilon_t = \dfrac{n_t - 2}{n_t}$, $P\left(\sum\limits_{r=1}^{n_t} W_r = 1\right) = \left(\dfrac{n_t - 1}{n_t}\right)^{n_t - 1}$ та $\lim\limits_{n_t \to +\infty} P\left(\sum\limits_{r=1}^{n_t} W_r = 1\right) = e^{-1}$. Але при $m_t = 1$, $1 - \varepsilon_t = \dfrac{2}{n_t}$ і при $n_t \to +\infty$, $1 - \varepsilon_t \to 0$. Це означає, що аналіз переважної більшості множин $\bar{X}_t^D\left(y^{r-1}\right)$ буде дозволятися. З іншого боку, якщо $2m_t = n_t - 1$, тоді $\varepsilon_t = \dfrac{n_t - n_t + 1}{n_t} = \dfrac{1}{n_t}$ і $1 - \varepsilon_t = \dfrac{n_t - 1}{n_t}$ та при $n_t \to +\infty$, $1 - \varepsilon_t \to 1$. А це означає, що аналіз переважної більшості множин $\bar{X}_t^D\left(y^{r-1}\right)$ буде заборонятися. Щоб дійти до певного компромісу при дозволі або забороні подальшого аналізу множин $\bar{X}_t^D\left(y^{r-1}\right)$ при виборі ε_t значення m_t слід обирати так, щоб кількість заборон на дозволів була збалансованою в тому сенсі, щоби при задовільній швидкості роботи алгоритму $APrLexMax\left(\bar{X}^D\left(\hat{x}_0, \tilde{x}, s\right)\right)$ кількість помилкових заборон була би не високою.

Список використаної літератури

1. *Чупов С.В.* Структурні та стохастичні властивості алгоритму лексикографічного пошуку розв'язку задачі дискретної оптимізації// Компьютерная математика. – К.: Ин-т кибернетики имени В.М. Глушкова НАН Украины, 2016.– № 1.– С. 155–164.
2. *Чупов С.В.* Модифікації алгоритму пошуку лексикографічного максимуму множини// Наук. вісник Ужгород. ун-ту. Сер. матем. і інформ. – 2015.– Вип. № 2(27). – С. 168–173.

3. *Чупов С.В.* Стохастичний алгоритм лексикографічного пошуку розв'язку булевої задачі про ранець// Праці VII міжнародної школи-семінару "Теорія прийняття рішень"(29 вересня-4 жовтня 2014 р. м. Ужгород). – Ужгород: Ужгородський нац. ун-т, 2014. – С. 263-264.
4. *Сергієнко І.В., Шило В.П.* Задачі дискретної оптимізації. Проблеми, методи розв'язання, дослідження. – К.: Наукова думка, 2003. – 261 с.
5. *Червак Ю.Ю., Гренджа В.І., Чупов С.В.* Лексикографічна оптимізація в дискретному програмуванні// Міжнар. конф. "Питання оптимізації обчислень"(Київ, 6-9 жовтня 1997р.). – Праці/ НАН України. – Ін-т кібернетики ім. В.М. Глушкова, 1997. – С. 317–319.

Одержано 10.03.2016