УДК 004.728 : 519.87

А.А. Коваленко[1], Г.А. Кучук[2]

[1] *Харьковский национальный университет радиоэлектроники, Харьков*
[2] *Харьковский национальный университет Воздушных Сил имени И. Кожедуба, Харьков*

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ТРАФИКОМ МУЛЬТИСЕРВИСНОЙ СЕТИ НА ОСНОВЕ МЕТОДОВ ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ РЕШЕНИЙ

Рассмотрены алгоритмы последовательного улучшения решений для задачи дискретной оптимизации на коммутационных узлах мультисервисных сетей. Определены условия для выбора типа алгоритма в зависимости от характера решаемой задачи. Подробно проанализированы методы, которые базируются на таких алгоритмах: итеративные алгоритмы дискретного программирования; весовые локальные алгоритмы; стохастические и локально-стохастические алгоритмы дискретной оптимизации.

Ключевые слова: *коммутационный узел, мультисервисная сеть, алгоритм, метод, управление, оптимизация, трафик.*

Введение

Задачи оптимизации разного уровня сложности непрерывно возникают и должны оперативно решаться в современных высокоскоростных мультисервисных сетях. Преимущественно, решение таких задач реализуется коммутационными устройствами и сводится к определенному классу задач управления трафиком.

В работе [1] были представлены ре¬зультаты анализа комбинаторных алгри¬мов решения задач дискретной оптимизации. Также было показано, что для задач небольшой размерности, которые выполняются на коммутационных узлах мультисервисних сетей, во многих случаях преимущество имеют алгоритмы ограниченного перебора на решетках и на деревьях. В работах [2 - 11] рассматриваются алгорит¬мы дискретной оптимизации для различных задач большой размерности. Поскольку основным критерием при выборе алгоритмов для оптимизации трафика мультисервисных сетей является вычислительная сложность, то в работе [12] авторы проанализировали принципы выбора наименее затратных алгоритмов – изменения множества альтернатив задачи, в частности, были представлены ре¬зультаты анализа методов, которые базируются на алгоритмах предварительного расширения и последующего сужения множества допустимых решений, показаны их преимущества и недостатки, основным из которых является в большинстве случаев медленная сходимость при допустимой вычислительной сложности. Поэтому необходимо провести анализ группы алгоритмов, которые, по сравнению с рассмотренными, хотя и уступают по вычислительной сложности, однако существенно превосходят все рассмотренные ранее группы по скорости сходимости.

Целью данной статьи является проведение анализа алгоритмов последовательного улучшения решений с целью дальнейшей разработки методов для оптимального управления трафиком мультисервисной сети, основным условием при реализации которых являются ограничения на время поиска решения в коммутационных узлах современных высокоскоростных мультисервисных сетей при допустимой вычислительной сложности.

1. Общие принципы

Две ранее рассмотренные группы алгоритмов [1, 2] – это алгоритмы «чистого» дискретного программирования с хорошими показателями по вычислительной сложности. Однако, при условии наличия ограничений на время поиска решения в коммутационных узлах при допустимой вычислительной сложности, можно рассматривать гибридные алгоритмы, сочетающие подходы как дискретного, так и математического программирования. Такие алгоритмы, в большинстве случаев, позволяют достичь лишь локального экстремума и, как правило, носят название локальных алгоритмов дискретной оптимизации.

Основополагающим в понятии локального экстремума является понятие окрестности, которое, в свою очередь, требует наличия определенной метрики, ориентированной на дискретные множества, т.е. необходимо рассматривать метрическое пространство L. Элементами L являются изолированные точки, а для каждой пары точек $x, y \in L$ определяется расстояние (вещественное число $\rho(x, y)$), обладающее такими свойствами:

$$\begin{cases} \rho(x, y) \ge 0; \\ \rho(x, y) = \rho(y, x); \\ \rho(x, y) = 0 \Leftrightarrow x = y; \\ \rho(x, y) \le \rho(x, z) + \rho(z, y), \ \forall z \in L. \end{cases} \quad (1)$$

Таким условиям удовлетворяет большое количество метрик, например:

- пространство Z^n: – это n-мерное пространство целочисленных точек; расстояние между парой точек с координатами $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_n)$ определяется посредством метрик такого вида:

$$\rho(x,y)=\sum_{j=1}^{n}\left|x_j-y_j\right| \qquad (2)$$

или

$$\rho'(x,y)=\max_{1\le j\le n}\left\{\left|x_j-y_j\right|\right\}; \qquad (3)$$

- пространство B^n – это n-мерное пространство булевых векторов; метрика имеет вид

$$\rho(x,y)=\sum_{j=1}^{n}\left|x_j-y_j\right|,\ x_j y_j\in\{0,1\}; \qquad (4)$$

- пространство $M_A=\Delta=\Omega(A)$ – это пространство неупорядоченных выборок множества A ($\operatorname{card}A=h$); комбинаторные альтернативы $\tilde{x}\in M_A$ и $\tilde{y}\in M_A$ являются подмножествами множества A, а метрика пространства

$$\rho_M(\tilde{x},\tilde{y})=\tilde{x}\cup\tilde{y}\setminus\tilde{x}\cap\tilde{y}\,. \qquad (5)$$

Каждому множеству точек $\tilde{x}$ множества $\Omega(A)$ соответствует кортеж x в пространстве булевых векторов B^n, количество ненулевых компонент которого соответствует количеству компонент множества $\tilde{x}$, а паре множеств $\tilde{x},\tilde{y}\in\Omega(A)$ соответствует пара кортежей $\tilde{x},\tilde{y}\in B^n$, для которой расстояние Хэмминга $\rho(x,y)=\sum_{j=1}^{n}\left|x_j-y_j\right|$ совпадает с величиной метрики (5);

- пространство $N_A=\hat{\Delta}\subseteq\Omega(\Omega(A))$ – это пространство всех разбиений множества A на соответствующие подмножества, где расстояние между $\tilde{x}\in N_A$ и $\tilde{y}\in N_A$ определяется метрикой

$$\rho_N(\tilde{x},\tilde{y})=\max_{\tilde{\xi}\in\tilde{x}}\min_{\tilde{\eta}\in\tilde{y}}\rho_M\left(\tilde{\xi},\tilde{\eta}\right)+\max_{\tilde{\eta}\in\tilde{y}}\min_{\tilde{\xi}\in\tilde{x}}\rho_M\left(\tilde{\xi},\tilde{\eta}\right). \qquad (6)$$

Такая метрика может быть определена посредством нахождения максимальных расстояний между подмножествами разбиений $\tilde{x}$ и $\tilde{y}$ при различном порядке их рассмотрения; расстояния $\tilde{\xi}\in\tilde{x}$ и $\tilde{\eta}\in\tilde{y}$ могут быть определены, например, в соответствии с выражением (5) в пространстве M_A;

- пространство $P_A=\hat{\Delta}\subseteq A\times\ldots\times A=A^k$ – пространство всех k-перестановок в пределах множества кортежей A^k, а расстояние между точками пространства $\tilde{x}\in P_A$ и $\tilde{y}\in P_A$ определяется метрикой

$$\rho_p(\tilde{x},\tilde{y})=\overline{\overline{T^*(\tilde{x},\tilde{y})}}, \qquad (7)$$

где $T^*(\tilde{x},\tilde{y})$ – множество транспозиций с минимальным числом элементов, переводящее $\tilde{x}$ в $\tilde{y}$.

В дискретном пространстве L определим:

- окрестность $O_G(x,r)$ с центром в точке $x\in G$ и радиусом $r>0$ – это множество точек $y\in G$, удовлетворяющих условию $\rho(x,y)\le r$:

$$O_G(x,r)=\left\{y\in G\,\middle|\,\rho(x,y)\le r\right\}; \qquad (8)$$

- точка минимума функции f относительно окрестности радиуса r ($x=\arg\min_{y\in O_G(x,r)} f(y)$) – это точка $x\in G$, при условии, что окрестность $O_G(x,r)$ не является вырожденной и

$$\forall\left(y\in O_G(x,r)\right)\left(f(y)\ge f(x)\right); \qquad (9)$$

- точка локального максимума функции f – это точка $x\in G$, для которой существует радиус $r>0$, причем для него справедливо (9).

Можно сделать вывод, что если в точке x не существует локального экстремума функции f относительно окрестности радиусом $r_1>0$, то в такой точке не существует и такого же экстремума (минимума или максимума) функции f относительно любой из окрестностей радиусом $r\ge r_1$.

Предположим, что $f(y)=f(x)+\delta(x,y)$ для всех точек $y\in O_G(x,r)$, а количество точек в окрестности равно $(n-1)$. Тогда вектор, характеризующий изменения f(x) при переходах между различными точками окрестности $O_G(x,r)$, будет иметь следующий вид:

$$\vec{\delta}^r(x)=\left(\delta\left(x,y^1\right),\delta\left(x,y^2\right),\ldots,\delta\left(x,y^n\right)\right). \qquad (10)$$

Компоненты вектора (10) являются аналогами производных направления для континуальных экстремальных задач.

Один из вариантов классификация типов алгоритмов последовательного улучшения решений представлен на рис. 1.

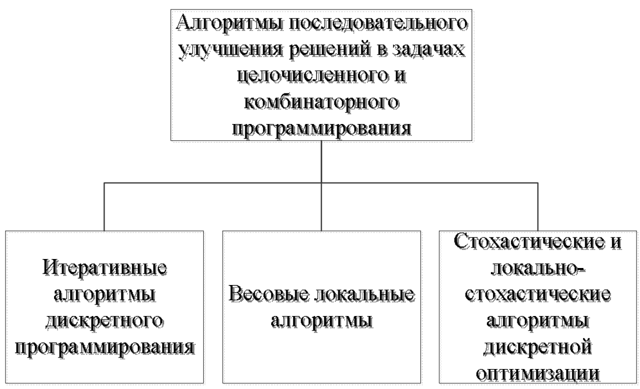

Рис. 1. Классификация типов алгоритмов последовательного улучшения решений

Далее рассмотрим особенности каждого из 3-х типов классификации применительно к решению задач оптимизации на коммутационных узлах мультисервисных сетей.

2. Итеративные алгоритмы дискретного программирования

Для примера, без ограничения общности, будем рассматривать задачу минимизации применительно к любым целочисленным и комбинаторным пространствам.

Первый шаг итеративных алгоритмов – это всегда выбор начального приближения x^0 путем решения релаксированной задачи с последующим округлением, либо на основании специальных подходов, регламентируемых свойствами задачи (относится к весовым локальным алгоритмам).

На каждом из последующих шагов алгоритма определяются компоненты $\delta\left(x^h, y^i\right)$ вектора спада в окрестности точки x^h с заданным радиусом, причем:

- окрестность может оказаться вырожденной и необходимо выбирать другую точку или увеличивать радиус окрестности;

- точка x^h может являться точкой локального минимума ($\delta\left(x^h, y^i\right) \geq 0$), при этом вычислительный процесс завершается (при достижении радиусом максимально возможного значения) либо начинается расчет вектора спада с увеличенным радиусом;

- если существуют точки $y^i \in O_G\left(x^h, r\right)$ для которых $\delta\left(x^h, y^i\right) < 0$, то осуществляется переход в одну из таких точек с последующим возвратом к началу шага.

Пример задачи булева программирования:

$$f(\vec{x}) = \sum_{j=1}^{n} C_j x_j \to \min,$$

$$\Delta_\beta = G =$$

$$= \left\{ \vec{x} \left| \sum_{j=1}^{n} a_{ij} x_j = b_i \left(i = \overline{1, L}\right), \sum_{j=1}^{n} a_{ij} x_j \leq b_i \left(i = \overline{L+1, n}\right) \right. \right\},$$

для которой требуется построить алгоритм локальной оптимизации с фиксированным радиусом r.

Такой алгоритм будет содержать два принципиальных шага, представленных ниже.

Шаг 0. Осуществляется выбор начального приближения $x^0 \in G$.

Шаг h + 1:

- осуществляется определение компонент вектора спада $\delta\left(\vec{x}^h, \vec{y}\right)$ в окрестности $O_G\left(\vec{x}^h, r\right)$ точки $\vec{x}^h$ радиуса r. Если $J = \{1, \ldots, n\}$ – множество индексов компонент векторов $\vec{x}$ и $\vec{y}$, а $U_r = \{u\}$ – множество его подмножеств с мощностью меньшей r, тогда каждому из подмножеств можно поставить в соответствие точку в окрестности $O_G\left(\vec{x}^h, r\right)$: вектор $\vec{y}$ с элементами $y_j = x_j^h$ при $j \notin u$ и $y_j = 1 - x_j^h$ при $j \in u$ и соответствующее значение компоненты вектора спада $\delta\left(\vec{x}^h, \vec{y}\right) = \sum_{j \in u_0} C_j - \sum_{j \in u \setminus u_0} C_j$, $u_0 \subseteq u$, подмножество индексов множества $x_j^h = 0$;

- выполняя перебор указанных точек, наряду с вычислением значений компонент вектора спада и проверкой на допустимость в терминах ограничений, выбирается точка $\vec{y}^* \in G$, для которой выполняется $\delta\left(\vec{x}^h, \vec{y}\right) < 0$, и она переобозначается как $\vec{y}^* = \vec{x}^{h+1}$, после чего выполняется шаг h + 2;

- в случае необнаружения такой точки окрестность считается вырожденной и осуществляется переход к другой начальной точке $\vec{x}^0$ либо производится расширение окрестности. Условиями окончания алгоритма являются

$$h + 1 > 1 \quad \text{и} \quad \forall\left(\vec{y} \in O_G(\vec{x}, r)\right)\left(\delta\left(\vec{x}^h, \vec{y}\right) \geq 0\right).$$

В задачах k-значного целочисленного программирования, итеративные алгоритмы строятся по аналогии с рассмотренным примером, главным отличием является то, что компоненты вектора $\vec{y} \in O_G\left(\vec{x}^h, r\right)$ более чем на единицу могут отличаться от компонент $\vec{x}^h$, однако, общее изменение всех компонент, в соответствии с метрикой $\sum_{j=1}^{h} \left|x_j - y_j\right|$, должно быть менее r.

В комбинаторных задачах итеративные алгоритмы позволяют решать такие задачи без приведения к задачам целочисленного программирования. При построении алгоритмов, за основу берется схема, аналогичная рассмотренному примеру, с тем отличием, что на каждом шаге h рассмотрению подлежит множество комбинаторных альтернатив $\tilde{y}$, принадлежащих окрестности $Q\left(\tilde{x}^h, r\right)$ комбинаторной альтернативы $\tilde{x}^h$, и удаленных от $\tilde{x}^h$, согласно метрике $\rho\left(\tilde{x}^h, \tilde{y}\right)$, на расстояние, не превышающее r.

3. Весовые локальные алгоритмы

Весовые локальные алгоритмы применяются в задачах булева программирования и позволяют выполнять адаптацию к исходным условиям задачи, в частности, область поиска экстремума определяется с учетом конкретных числовых параметров задачи. Как правило, процесс такой адаптации подразумевает последовательное назначение единиц компонента вектора $\vec{x}$ в соответствии с весами, зависящими, в свою очередь, от исходных данных.

Пример применения алгоритмов для задачи:

$$\sum_{j=1}^{n} C_j x_j \to \max_{\vec{x} \in \Delta_\beta};$$

$$\Delta_\beta = \left\{ \vec{x} \left| \sum_{j=1}^{n} a_{ij}\ x_j \le b_i \left(i=\overline{1,m}\right), x_j \in \{0,1\}, \left(j=\overline{1,n}\right) \right.\right\};$$

$$C_j \ge 0,\ a_{ij} \ge 0,\ b_i > 0\ \left(i=\overline{1,m};\ j=\overline{1,n}\right).$$

Введем $a_{ij}^{*} = a_{ij}/b_i$, тогда $\sum_{j=1}^{n} a_{ij}^{*} x_j \le 1$. Рассчитаем веса $\alpha_j \left(j=\overline{1,n}\right)$ по одному из критериев:

$$\alpha_j = C_j,\ \alpha_j = \frac{C_j}{\max\limits_i a_{ij}^{*}},\ \alpha_j = \frac{C_j}{\max\limits_i a_{ij}^{*} - \min\limits_i a_{ij}^{*}},$$
$$\alpha_j = C_j \Big/ \sum_{i=1}^{m} a_{ij}^{*},\ \alpha_j = C_j \Big/ \left(\max_i a_{ij}^{*}\right) \sum_{i=1}^{m} a_{ij}^{*}. \qquad (11)$$

В результате перестановки индексов в соответствии с полученными весами в порядке невозрастания определяется последовательность $\left\langle j_1^0,\ldots,j_n^0 \right\rangle$ и находится решение $\vec{x} = \left\langle x_1,\ldots,x_n \right\rangle$ посредством определения компонент $x_j = x_{j_s^0}$, учитывая порядок приоритетов и выполнение таких ограничений:

$$x_{j_s^0} = \begin{cases} 1, \text{ если } \sum\limits_{k<s} a_{ij_k^0} + a_{ij_s^0} \le b_i\ \left(i=\overline{1,m}\right), \\ 0, \text{ иначе.} \end{cases} \qquad (12)$$

Зачастую, алгоритм дополняется процедурой, позволяющей определять последовательность $\left\langle j_1^{*},\ldots,j_n^{*} \right\rangle$ в соответствии с упорядочением $C_1 \ge \ldots \ge C_n$ и находить

$$K = \min\left\{ j_h^{*} \left| x_{j_h^{*}} = 0 \right.\right\},\ \ell = \max\left\{ j_h^{*} \left| x_{j_h^{*}} = 1 \right.\right\}.$$

4. Стохастические и локально-стохастические алгоритмы дискретной оптимизации

Основная область применения такого типа алгоритмов ограничивается булевым и комбинаторным программированием. Алгоритмы базируются на использовании стохастических процедур назначения единиц с заданной вероятностью и генерации случайных перестановок.

Процедура назначения единиц с заданной вероятностью р для компонент вектора $\vec{x} = \left(x_1,\ldots,x_n\right)$ с использованием генерируемой последовательности случайных величин $\xi_i \left(i=1,2,\ldots\right)$, равномерно распределенных на интервале [0, 1], имеет вид:

$$x_j = \begin{cases} 1, \text{ если } \xi_j \le p, \\ 0, \text{ если } \xi_j > p. \end{cases} \qquad (13)$$

Процедура генерации случайных перестановок в исходной последовательности $J_0 = \left\langle j_1^0,\ldots,j_n^0 \right\rangle$ строится так: итоговая перестановка J_{n-1} формируется как последний член последовательности перестановок J_0, …, J_{n-1}, каждая из которых определяется случайной транспозицией пары элементов в предыдущей перестановке.

Шаг 1 – введение исходной последовательности $J_0 = \left\langle j_1^0,\ldots,j_n^0 \right\rangle$, которая является перестановкой в множестве {1, …, n}, и числа i = n; последовательность представляется в виде $J_0 = J_{n-i}$.

Шаг 2 – вычисление значения $\eta = 1 - \xi^{\alpha}$, где $\xi \in (0,1)$ – сгенерированное случайное число, $\alpha \in \{1,2,\ldots\}$ – натуральное число.

Шаг 3 – выделение целой части числа $K = \ll \eta i + 1 \gg$.

Шаг 4 – замена местами элементов последовательности J_{n-i}, стоящих на местах i и k, для получения последовательности J_{n-i+1}.

Шаг 5 – замена i на (i - 1) и, если полученная величина больше 1, осуществление перехода к шагу 2, иначе случайная перестановка $J = J_{n-1}$ считается полученной.

Рассмотрим класс алгоритмов управляемого случайного поиска, позволяющий производить учет результатов, получаемых в процессе поиска, и на их основе вносить изменения в параметры процесса поиска, на примере задачи:

$$\sum_{j=1}^{n} C_j x_j \to \max_{\vec{x} \in \Delta_\beta},$$

$$\Delta_\beta = \left\{ \vec{x} \left| \begin{array}{l} \sum\limits_{j=1}^{n} a_{ij}\ x_j \le b_i \left(i=\overline{1,m}\right),\ x_j \in \{0,1\}, \\ \left(j=\overline{1,n}\right), C_j \ge 0, a_{ij} \ge 0, b_i > 0 \end{array} \right.\right\}.$$

Смысл каждого итеративного шага заключается в реализации поиска допустимого решения $\vec{x}^{[k]}$, кроме того, удовлетворяющему дополнительному условию $\sum_{j=1}^{n} C_j x_j \ge b_0^{[k]}$, причем $b_0^{[k]} = b_{0p}^{[k-1]} + \Delta b_0^{[k]}$, где $b_{0p}^{[k-1]} = \sum_{j=1}^{n} C_j x_j^{[k-1]}$ – лучшее значение целевой функции с предыдущего шага, $\Delta b_0^{[k]}$ – допустимая погрешность.

Процесс поиска $\vec{x}^{[k]}$ на каждом итеративном шаге требует введения произвольного начального вектора $\vec{x}^{k0}$ получаемого, например, посредством случайного назначения единиц. Последовательность векторов $\vec{x}^{k1}, \vec{x}^{k2}, \ldots, \vec{x}^{kr}$ может быть определена в результате выполнения итеративных шагов случайного поиска, использующих вектора с предыдущего шага. Для этого вычисляются следующие невязки:

$$\delta_0^{k,r+1} = \max\left\{\left(b_0^{[k]} - \sum_{j=1}^{n} C_j x_j^{kr}\right)\Big/ b_0^{[k]}\right\},$$

$$\delta_i^{k,r+1} = \max\left\{\left(\sum_{j=1}^{n} a_{ij} x_j^{kr} - b_i\right)\Big/ b_i, 0\right\}\left(i = \overline{1,m}\right).$$

Суть управления поиском сводится к изменению вероятности назначения единиц от 0 до p^0 на основе невязок.

Выводы

В статье представлены результаты анализа алгоритмов последовательного улучшения решений с целью дальнейшей разработки методов для оптимального управления трафиком мультисервисной сети, основным условием при реализации которых являются ограничения на время поиска решения в коммутационных узлах современных высокоскоростных мультисервисных сетей при допустимой вычислительной сложности. Определены условия для выбора типа алгоритма в зависимости от характера решаемой задачи. Приведены примеры реализации алгоритмов для конкретных задач булева и комбинаторного программирования.

Направление дальнейших исследований – разработка метода автоматического выбора алгоритма оптимизации при выполнении задач в коммутационных узлах современных высокоскоростных мультисервисных сетей.

Список литературы

1. Коваленко, А.А. Выбор комбинаторного алгоритма оптимизации при управлении трафиком мультисервисной сети [Текст] / А.А. Коваленко, А.А. Можаев, Г.А. Кучук // Системи обробки інформації: збірник наукових праць. – Х.: ХУ ПС, 2015. – Вип. 10 (135). – С. 97 – 102.

2. Коваленко, А.А. Подходы к синтезу информационной структуры системы управления объектом критического применения [Текст] / А.А. Коваленко // Системи обробки інформації. – Х.: ХУ ПС, 2014. – Вип. 1 (117). – С. 180 – 184.

3. Кучук, Г.А. Модель процесса эволюции топологической структуры компьютерной сети системы управления объектом критического применения [Текст] / Г.А. Кучук, А.А. Коваленко, А.А. Янковский // Системи обробки інформації: збірник наукових праць. – Х.: ХУ ПС, 2014. – Вип. 7 (123). – С. 93 – 96.

4. Кучук, Г.А. Концептуальний підхід до синтезу структури інформаційно-телекомунікаційної мережі [Текст] / Г.А. Кучук, І.В. Рубан, О.П. Давікоза // Системи обробки інформації : збірник наукових праць. – Х.: ХУ ПС, 2013. – Вип. 7 (114). – С. 106 – 112.

5. Кучук, Г.А. Синтез стратифікованої інформаційної структури інтеграційної компоненти гетерогенної складової Єдиної АСУ Збройними Силами України [Текст] / Г.А. Кучук, О.П. Давікоза // Наука і техніка Повітряних Сил Збройних Сил України : науково-технічний журнал. – Х.: ХУ ПС, 2013. – № 3(12). – С. 154-158.

6. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы [Текст] / В.Г. Олифер, Н.А. Олифер; 4-е изд. – СПб.: Питер, 2010. – 944 с.

7. Кучук, Г.А. Управление ресурсами инфотелекоммуникаций [Текст] / Г.А. Кучук, Р.П. Гахов, А.А. Пашнев. – М.: Физматлит, 2006. – 220 с.

8. Кучук, Г.А. Інформаційні технології управління інтегральними потоками даних в інформаційно-телекомунікаційних мережах систем критичного призначення [Текст] / Г.А. Кучук. – Х.: ХУ ПС, 2013. – 264 с.

9. Зайцев, М.Г. Методы оптимизации управления и принятия решений [Текст] / М.Г. Зайцев, С.Е. Варюхин. – М. : ДЕЛО, АНХ, 2008. – 664 с.

10. Коваленко, А.А. Подходы к синтезу технической структуры компьютерной системы, образующей систему управления объектом критического применения [Текст] / А.А. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Х.: ХУ ПС, 2014. – Вип. 1(38). – С. 116 – 119.

11. Коваленко, А.А. Подходы к оптимизации распределения задач управления по компонентам компьютерной системы, образующей систему управления объектом критического применения [Текст] / А.А. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України: науково-технічний журнал. – 2014. – № 2(15). – С. 158 – 160.

12. Кучук, Г.А. Використання методів зміни простору рішень для оптимізації управління трафіком мультисервісних мереж [Текст] / Г.А. Кучук, А.А. Коваленко // Системи обробки інформації: збірник наукових праць. – Х.: ХУ ПС, 2016. – Вип. 5 (142). – С. 128 – 132.

Надійшла до редколегії 11.07.2016

Рецензент: д-р техн. наук, проф. О.О. Можаєв, Національний технічний університет «ХПІ», Харків.

ОПТИМАЛЬНЕ УПРАВЛІННЯ ТРАФІКОМ МУЛЬТИСЕРВІСНОЇ МЕРЕЖІ НА ОСНОВІ МЕТОДІВ ПОСЛІДОВНОГО ПОКРАЩЕННЯ РІШЕНЬ

А.А. Коваленко, Г.А. Кучук

Розглянуто алгоритми послідовного покращення рішень для задачі дискретної оптимізації на комутаційних вузлах мультисервісних мереж. Визначено умови для вибору типу алгоритму залежно від характеру задачі, що вирішується. Детально проаналізовано методи, які базуються на таких алгоритмах: ітеративні алгоритми дискретного програмування; вагові локальні алгоритми; стохастичні і локально-стохастичні алгоритми дискретної оптимізації.

Ключові слова: *комутаційний вузол, мультисервісна мережа, алгоритм, метод, управління, оптимізація, трафік.*

MULTISERVICE NETWORK TRAFFIC OPTIMAL CONTROL ON THE BASIS OF METHODS OF SEQUENTIAL REFINEMENT OF A SOLUTION

A.A. Kovalenko, G.A. Kuchuk

The algorithms of sequential refinement of a solution are considered for the problem of discrete optimization on the switching hosts in multiservice networks. Conditions for selection of certain algorithm type depending on character of the problem are defined. Methods, which are based on such algorithms, are analyzed in details: iterative algorithms of discrete programming; weight local algorithms; stochastic and locally-stochastic algorithms of discrete optimization.

Keywords: *interconnect knot, multiservice network, algorithm, method, control, optimization, traffic.*