УДК 004.414.2.021:519.853.62

А.В. ПЕТРОВ

Харьковский университет Воздушных Сил им. Ивана Кожедуба, Украина

ПРИМЕНЕНИЕ ВЕКТОРА СПАДА ПРИ ИНСТАЛЯЦИИ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО КОМПЛЕКСА В СРЕДЕ ГЕТЕРОГЕННОЙ МУЛЬТИСЕРВИСНОЙ СЕТИ

Анализируются методы решения задач линейного булева программирования. Приведены численные оценки машинного эксперимента исследования алгоритмов булева программирования. На основании результатов машинного эксперимента предложен метод с наиболее эффективным алгоритмом реализации задачи на ЭВМ, позволяющий учесть специфику рассматриваемой задачи. Получено квазиоптимальное решение задачи оптимизации структуры гетерогенной мультисервисной сети. Представлен метод инсталляции специализированного программного комплекса в среде гетерогенной мультисервисной сети.

Ключевые слова: *гетерогенный, мультисервисные сети, метод вектора спада.*

Введение

Постановка проблемы. В настоящее время при создании единой автоматизированной системы управления (ЕАСУ) войсками одной из актуальных задач является задача распределения программ специализированных программных комплексов (СПК) по узлам гетерогенной сети [1], так как для Вооруженных Сил Украины в настоящий момент наиболее характерна разнородность и неравнозначность аппаратно-программных средств ЕАСУ, а требования оперативности, достоверности, непрерывности и полноты информации имеют первостепенное значение. Нормирование структуры сети неразрывно связано с обеспечением функционирования СПК, обеспечивающих решение задач управления войсками.

Анализ литературы [2 – 5] показал, что в существующих подходах к решению задачи распределения задач СПК по узлам сети не учитывается ее гетерогенность. Кроме того, в СПК ЕАСУ основным показателем функционирования СПК является своевременность доставки, требующим в условиях гетерогенности аппаратно-программных средств поддержки СПК. Поэтому необходимо решить задачу минимизации суммарных затрат сетевого ресурса, при этом достаточно эффективными являются математические методы нахождения решения, которые в отличии от методов имитационного моделирования являются менее ресурсоемкими. Данная задача сформулирована в [6].

Целью данной статьи является разработка метода инсталяции специализированного программного комплекса в среде мультисервисной гетерогенной сети ЕАСУ, базирующегося на решении задачи оптимального распределения задач СПК по узлам сети, сформулированной в [6].

1. Выбор и обоснование метода решения

Сформулированная в статье [6] задача минимизации затрат сетевого ресурса приведена к линейному стандартному виду задачи булевого программирования [7]:

найти $$\min \sum_{l=1}^{l_0} C_l Z_l\,, \qquad (1)$$

если $$\sum_{l=1}^{l_0} d'_{k'l} Z_l = l_{k'}\,,\ \forall k' \in K'\,; \qquad (2)$$

$$\sum_{l=1}^{l_0} d_{kl} Z_l \le l_k\,,\ \forall k \in K\,; \qquad (3)$$

$$Z_l \in \{0,1\}\,,\ \forall l \in L\,; \qquad (4)$$

$$K = \left\{k \middle| k = \overline{1,k_0}\right\}; \qquad (5)$$

$$K' = \left\{k' \middle| k' = \overline{1,m+1}\right\}; \qquad (6)$$

$$L = \left\{l \middle| l = \overline{1,l_0}\right\}, \qquad (7)$$

где $$Z_l = \begin{cases} x_{ij},\ \text{где } l = (i-1)*n + j; \\ y_s,\ \text{где } l = mn + s; \end{cases} \quad l = \overline{1,l_0}\,;$$

$$C_l = \begin{cases} a_{ij} + \sum\limits_{k=1}^{n} (p_{ik} + q_{ik}) h_{jk}\,; & \text{где } l = (i-1)*n + j; \\ \sum\limits_{i,i',j,j'} c_{ii'} h_{jj'},\ \text{для } l = mn + s; \end{cases}$$

$$d'_{k'l} = \begin{cases} 0,\ \text{если } (k' \le m) \& (l \le (k'-1)n) \& (l > k'n); \\ 1,\ \text{если } (k' \le m) \& ((k'-1)n + 1 \le l \le k'n); \\ b_{ij},\ \text{если } (k' = m+1) \& (l = (i-1)n + j); \\ 0,\ \text{если } (k' = m+1) \& (l > mn); \end{cases}$$

© А.В. Петров

$$d'_{k'l} = \begin{cases} 0,\ \text{если}\ (k \le m+2)\&(l \le (k-1)n)\&(l > kn); \\ t_{ij},\ \text{если}\ (k' \le m)\&((k-1)n+1 \le l \le kn); \\ f_{ij},\ \text{если}\ (k = m+1)\&(l \le mn); \\ \sum_{j'=1}^{n} g_{jj'},\ \text{если}\ (k = m+2)\&(l \le mn); \\ \alpha_{ii'jj'},\ \text{если}\ k>m; \end{cases}$$

$$l_{k'} = \begin{cases} 1,\ \text{если}\ k' \le m; \\ m,\ \text{если}\ k'=m+1; \end{cases} \quad l_k = \begin{cases} \tau_k,\ \text{если}\ k \le m; \\ \beta_k,\ \text{если}\ k>m, \end{cases}$$

$\alpha_{ii'jj'}$ – коэффициенты дополнительных ограничений; β_k – свободный член дополнительных ограничений; p_{ik} – количество передаваемых i-й задачей на k-й узел сети информационных единиц прикладного уровня, полученных после ее решения; $c_{ii'}$ – число информационных единиц прикладного уровня, передаваемых от задачи i к задаче i'; a_{ij} – процессорное время, затраченное на выполнение i-ой задачи в j-м узле; h_{jk} – затраты вычислительного ресурса на передачу информационной единицы прикладного уровня с узла j на узел k; q_{ik} – объемы вводимой информации с узлов;

$$x_{ij} = \begin{cases} 1,\ \text{если i-ая задача выполняется в j-м узле сети;} \\ 0,\ \text{в противном случае.} \end{cases}$$

Задача (1) – (7) является оптимизационной задачей булевого программирования.

Следствием наличия значительных трудностей и специфических особенностей в решении булевых линейных задач оптимизации большой размерности является большое количество методов и алгоритмов. При их разработке привлекался весьма разнообразный математический аппарат – от математической логики до теории динамических систем.

В работе [8] проанализированы группы методов решения задачи линейного булевого программирования:

– методы отсечения;

– метод ветвей и границ;

– методы последовательного анализа и отсева вариантов;

– аддитивные методы;

– приближенные методы локальной оптимизации (например, метод вектора спада, Н-метод, метод Бендера);

– лексикографические методы.

На рис. 1 показаны пять основных направлений развития методов и соответствующих им алгоритмов.

Первые два из них связаны с выявлением условий, при которых решение задач булева программирования может полностью или почти полностью опираться на использование алгоритмов непрерывного программирования и теории оптимального управления.

Ввиду того, что эти возможности существуют лишь для сравнительно небольшого класса задач, и практически нереализуемы для рассматриваемой постановки, большой интерес представляют три остальных направления собственно методов и алгоритмов булева программирования. Третье и четвертое направления, охватывающие методы и алгоритмы последовательного улучшения решений, - прямые методы дискретной оптимизации, пятое направление - непрямые методы собственно дискретной оптимизации. Так же как и для обычных задач математического программирования, характерная особенность непрямых методов состоит в том, что при их применении осуществляется редукция исходной задачи с использованием условий достижений экстремума и других условий.

Для рассматриваемой линейной оптимизационной задачи (1) – (7) при выборе метода решения учитывались два основных фактора, которые наиболее важны с практической точки зрения:

- время решения задачи с использованием представленного алгоритма на ЭВМ;

- объем памяти, используемый при реализации алгоритма.

В настоящее время для оценки эффективности алгоритмов используется два различных подхода:

– проведение вычислительного эксперимента с применением ЭВМ;

– применение теории сложности алгоритмов.

Учитывая, что для решения задачи (1) – (7) достаточно найти наиболее эффективный алгоритм, с точки зрения реализации его на ЭВМ, ориентированный на конкретную, вычислительную технику и то, что результаты исследований теории сложности алгоритмов дают представление о поведении задач в наихудшем случае, тогда как для вычислительной техники важно их поведение в среднем, был выбран первый подход.

Результат машинного эксперимента, выполненного по алгоритмам различных методов с различными вариантами исходных данных, приведены в табл. 1 (номера используемых методов: № 1 – метод последовательных отсечений Гомори; № 2 – метод ветвей и границ; № 3 – метод вектора спада; № 4 – Р-метод расширения частичных решений; № 5 – метод последовательного анализа вариантов Михалевича; № 6 – аддитивный метод Балаша; № 7 – метод построения последовательности планов; m – число постоянно функционирующих задач в сети; n – число узлов вычислительной сети).

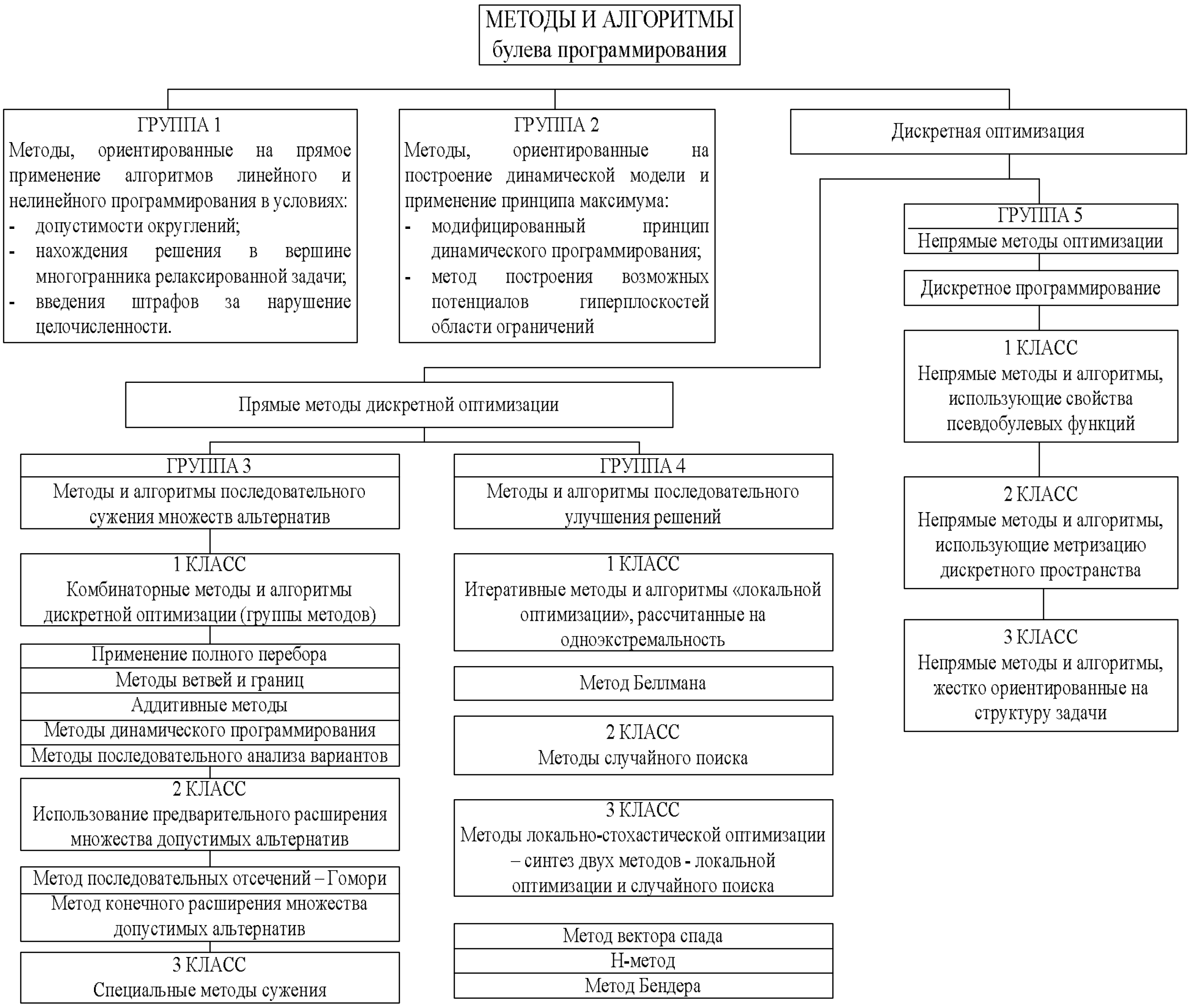

Рис. 1. Классификация методов и алгоритмов решения задачи булева программирования

Таблица 1

Результаты машинного эксперимента исследования алгоритмов булева программирования

Размерность задач $(m \times n)$	Время счета 1500 задач для алгоритмов (с)							Число точно решенных задач алгоритмом №3
	№1	№2	№3	№4	№5	№6	№7	
20x10	37,2	34,2	14,4	24,6	26,4	25,8	23,4	1158
50x10	151,2	135,6	57	92,4	107,4	118,8	96,6	1160
80x20	270	222	96	156	186	174	162	1228
100x20	366	330	168	234	258	270	246	1389
143x36	1206	1074	642	1110	1086	804	1008	1455
185x54	3672	3198	1410	2226	2538	2796	2508	1462
243x84	6936	6024	2724	4494	5106	4782	4512	1478

Анализ данных результатов позволил сделать следующие выводы относительно возможностей исследуемых методов решения задачи булева программирования большой размерности:

– при увеличении размерности задачи процент локальных решений, найденных методом вектора спада, которые отличны от глобального, не увеличивается, а даже имеет тенденцию в сторону уменьшения; – при увеличении размерности задачи метод вектора спада позволяет получать достаточно хорошие результаты по времени решения при сравнительно небольших объемах вычислений по сравнению с другими исследуемыми методами;

– при использовании метода вектора спада возможно прервать процесс вычислений, имея близкое к оптимальному решение, а при возможности его улучшить.

Используем предложенный критерий и результаты проведенного анализа для выбора метода решения оптимизационной задачи [2].

Данная задача относится к задачам оптимизации большой размерности. Поэтому, учитывая вышесказанное, для решения рассматриваемой задачи был выбран метод локальной оптимизации – метод вектора спада, реализация которого применительно к (1) – (7) описана ниже.

2. Применение метода вектора спада при инсталляции специализированного программного комплекса в среде мультисервисной гетерогенной сети

Метод вектора спада относится к приближенным методам локальной оптимизации задач дискретного программирования. Метод позволяет, путем сужения локальных открытых окрестностей начального приближения, по направлению уменьшения значений специально формируемого вектора, найти оптимальное решение.

Пусть M – метрическое пространство с некоторой метрикой $\rho(x, y)$, определенной для любых двух точек x и y из этого пространства.

Если r – какое-либо положительное число, а x – фиксированная точка метрического пространства M, то множество всех точек $y \in M$, для которых $\rho(x, y) < r$, образует открытую окрестность с центром x и радиусом r.

Множество всех точек $y \in M$, для которых $\rho(x, y) \le r$, образует замкнутый шар радиусом $r > 0$ с центром в точке $x \in M$.

Если множество M дискретно, то для любого открытого шара в метрическом пространстве M найдется замкнутый шар с тем же центром и полностью совпадающий с ним. В связи с этим для дискретных множеств, которые будем рассматривать в дальнейшем, перейдем от открытых окрестностей к окрестностям, связанным с понятием замкнутого шара. Тогда окрестностью точки x будет замкнутый шар радиуса $r > 0$ с центром $x \in M$, обозначаемый символом $O_M(x, r)$.

Итак, пусть M – дискретное множество, а $f(x)$ – произвольная действительная функция, определенная на M. Тогда точка $x \in M$ – точка локального минимума функции f относительно окрестности радиуса r, если для всех точек $y \in O_M(x, r)$ выполнено неравенство

$$f(x) \le f(y) \text{ и } O_M(x, r) \setminus \{x\} \ne \varnothing.$$

Если некоторая точка $x \in M$ не является локальной точкой минимума функции f относительно окрестности радиуса $r_1 > 0$, то она не будет таковой и относительно окрестности любого радиуса r_2, удовлетворяющего неравенству $r_1 \le r_2$.

Рассматриваемая задача булева программирования состоит в определении локального решения, выбираемого из точек локальных минимумов функции f, заданной на M, относительно окрестности $r > 0$.

Векторная функция $\Delta_M^r(x)$, определенная на M является *вектором спада функции* f относительно окрестности радиуса r, если выполняются условия:

1) значение функции Δ_M^r в каждой точке $x \in M$ является l-мерным $(l = l(x, y))$ вектором с компонентами $\Delta_1, ..., \Delta_l$, которые являются действительными числами;

2) точка $x \in M$ является точкой локального минимума функции f тогда и только тогда, когда $\Delta_i \ge 0$ при всех допустимых значениях индекса i;

3) если $x \in M$ не является точкой локального минимума функции f относительно $O_M(x, r)$, то с помощью вектора спада можно определить точку x', принадлежащую окрестности $O_M(x, r)$, такую, что $f(x') < f(x)$.

В соответствии с этим вектор $\Delta_M^r(x)$ позволяет для каждой точки $x \in M$ найти направление уменьшения (спада) значений функции f в окрестности $O_M(x, r)$. Идея использования вектора спада $\Delta_M^r(x)$ для отыскания такого направления лежит в основе применяемого метода. Для задачи [2] компоненты вектора спада можно вычислить проще, чем значения функции f в точках окрестности $O_M(x, r)$. Кроме того, поиск точки x' такой, что $f(x') < f(x)$ или выявление того факта, что x является точкой локального минимума функции f относительно $O_M(x, r)$, предполагают рассмотрение лишь части компонент вектора $\Delta_M^r(x, r)$.

В соответствии с вышеописанной схемой вектора спада можно представить общую последовательность действий в виде алгоритма (рис. 2).

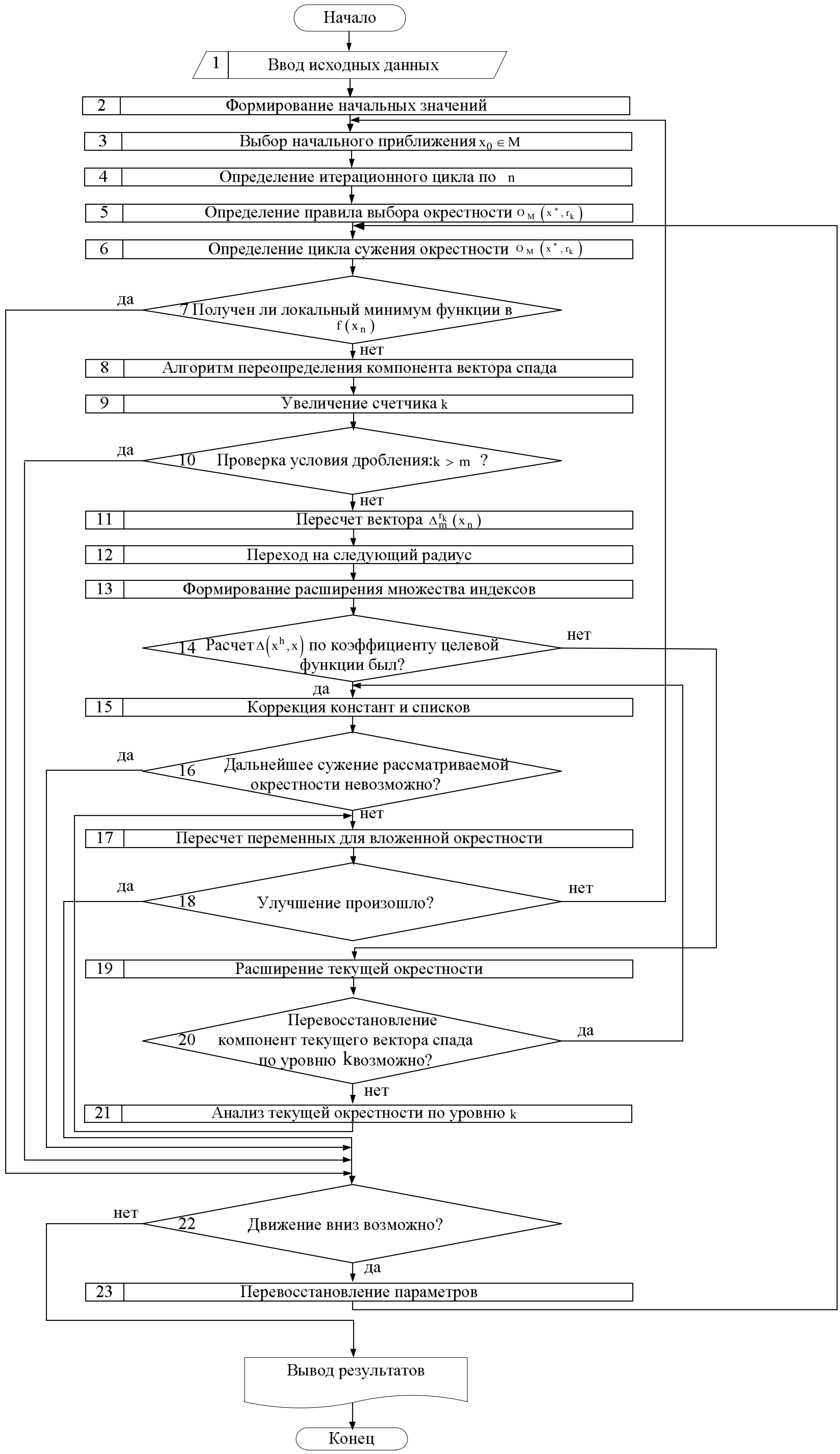

Рис. 2. Блок-схема алгоритма оптимизации

Первый шаг. Случайно или, исходя из учета особенностей решаемой задачи, выбираем некоторое начальное приближение $x^0 \in M$ и задаем величину радиуса $r > 0$.

Второй шаг. Задаем некоторую последовательность радиусов $\{r_1, r_2, ..., r_m\}$, удовлетворяющую соотношениям

$$0 < r_1 < r_2 <, ..., < r_m = r,\ m \geq 1.$$

Третий шаг. Полагаем $n = 0$.

Четвертый шаг. На каждой $(n+1)$)-й итерации алгоритма выполняем следующие действия:

4.1 Полагаем $k = 1$.

4.2 Рассматриваем окрестность $O_M\left(x^*, r_k\right)$.

4.3 По значениям компонент вектора спада $\Delta_M^{r_k}(x_n)$ определяем, является ли $f(x_n)$ локальным минимумом функции $f(x)$ относительно $O_M^{r_k}(x_n, r_k)$. Если $f(x_n)$ – локальный минимум, то при $k < m$, заменив k на $k+1$, переходим к п. 4.2, а при $k = m$ – к шагу 5. В противном случае переходим к п.4.4.

4.4 В окрестности $O_M(x_n, r_k)$ среди точек x с меньшим значением целевой функции, чем $f(x_n)$, выбрать одну, которая обеспечивает минимум функции $f(x)$. Обозначив эту точку x_{n+1} и заменив n на $n+1$, возвращаемся к п.4.1 четвертого шага данного алгоритма.

Пятый шаг. Завершение работы алгоритма – точка x_n является искомой точкой локального минимума $f(x)$ относительно окрестности радиуса r.

Списание обшей схемы метода необходимо дополнить рассмотрением следующего возможного случая: пусть локальное решение $\tilde{x}$ рассматриваемой задачи дискретного программирования не удовлетворяет некоторому заданному дополнительному критерию (например, критерию, устанавливающему максимально допустимое отклонение локального минимума от глобального). Для такого случая можно предложить несколько вариантов поиска других приближенных решений.

Во-первых, выбрав точку $\tilde{x}$ как новое приближение и увеличив первоначально заданную величину радиуса r, следует продолжить вычислительный процесс по алгоритму до получения других приближенных решений задач, которые, очевидно, будут ближе к глобальному (по значению целевой функции), чем $\tilde{x}$. Однако при больших значениях радиуса r процесс вычислений может существенно усложниться.

Во-вторых, исходя из точки $\tilde{x}$ и увеличив величину радиуса r, можно осуществить лишь один шаг алгоритма для получения точки x', удовлетворяющий неравенству $f(x') < f(\tilde{x})$; затем вернуться к первоначальной величине радиуса и продолжить вычислительный процесс по алгоритму.

В-третьих, не изменяя величины r, можно продолжить вычисления, исходя из некоторого другого начального приближения $x'_0 \in M$ и получить в результате локальное решение x', в общем случае, отличное от $\tilde{x}$.

Применим вышеописанную схему вектора спада для нахождения оптимального решения рассматриваемой задачи (1) – (7).

В качестве дискретного множества M рассмотрим подмножество B^n всех точек метрического пространства целых чисел, удовлетворяющих условию

$$x_j \in \{0,1\} \quad \forall j = \overline{1,n}.$$

Это подмножество является метрическим пространством с метрикой "расстояние":

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

При реализации общей схемы вектора спада для решения поставленной задачи во втором пункте четвертого шага алгоритма рассмотрим окрестность $O_{B^n}\left(x^h, r_k\right)$.

При этом координаты произвольной точки $x = (x_1, ..., x_n) \in B^n$ можно выразить через координаты точки x^h следующим образом:

$$x_j = \begin{cases} 1 - x_j^h, \text{ при } j \in W, \\ x_j^h, \text{ при } j \in \{1, n\} \setminus W, \end{cases}$$

где $W \subseteq \left\{\overline{1,n}\right\}$, то есть является подмножеством множества индексов.

При выполнении третьего и четвертого пунктов четвертого шага компоненты вектора спада будем вычислять по следующей формуле:

$$\Delta\left(x^h, x\right) = \sum_{j \in W^0} c_j - \sum_{j \in W \setminus W^0} c_j,$$

где c_j – коэффициент целевой функции задачи; W^0 – множество значений индекса $j \in W$, для которых

$$x^h = O\left(W^0 \subseteq W\right).$$

В [9] доказана теорема о сходимости алгоритма вектора спада в применении к канонической задаче булева программирования.

При любом выборе начального приближения из всех точек рассматриваемого булева пространства с метрикой M, удовлетворяющего ограничениям решаемой задачи и радиуса $r > 0$, последовательность приближений $\{x_n\}$, определяемая по предложенной выше модифицированной схеме, сходится к локальному решению задачи, в каноническом виде за конечное число шагов, не превышающее следующей величины:

$$\alpha = \beta\left(f\left(x^0\right) - c\right),$$

где $$c = \begin{cases} 0, \text{ если } c_j \geq 0 \text{ для } \forall j \in \overline{1,n}; \\ \sum\limits_{j \in J^-} c_j, \text{ в противном случае,} \end{cases}$$

J^- – множество тех значений индекса $j \in \overline{1,n}$, для которых $c_j < 0$, β – наименьшее общее кратное чисел β_j, взятых из следующего представления для c_j, $\left(j \in \overline{1,n}\right)$:

$$c_j = \frac{\alpha_j}{\beta_j}, \ \alpha_j, \beta_j – \text{целые числа, } \beta \neq 0.$$

Представленное выше положение сходимости алгоритма вектора спада для решения задач булева программирования основывается на конечности булева пространства B^n, вследствие чего функция, определенная на подмножестве $G \subset B^n$, удовлетворяет условиям теоремы, доказанной в [9].

Для ускорения вычислений по представленному алгоритму процесс можно частично распараллелить:

а) вычислительный процесс по алгоритму можно начинать одновременно на разных процессорах, исходя из различных начальных приближений, а затем выбрать лучшее из полученных решений;

б) каждый $(h+1)$-й шаг алгоритма можно выполнять одновременно на разных процессорах, исходя из одной и той же точки x^h, но работая с различными величинами (от r_1 до r_m) радиусов рассматриваемых окрестностей; окончанием каждого шага можно считать получение, хотя бы на одном из процессоров точки x^{h+1}, согласно п.4.4, алгоритма реализации метода вектора спада.

Выводы

В статье предложен и обоснован подход к решению задачи булева программирования большой размерности. Он позволил выбрать метод, имеющий наиболее эффективный алгоритм реализации конкретной задачи на ЭВМ учитывающий факторы:

– допустимые границы получения квазиоптимального решения;

– интервалы изменения коэффициентов целевой функции и ограничений;

– верхнюю границу астрономического времени нахождения квазиоптимального решения на ПЭВМ.

Получено квазиоптимальное решение задачи оптимизации структуры гетерогенной мультисервисной сети. По сравнению с общепринятыми точными методами решения данной задачи большой размерности достигнут выигрыш по астрономическому времени при аналогичных начальных условиях.

Предложен метод инсталляции специализированного программного комплекса в среде гетерогенной мультисервисной сети. При использовании данного метода затраты вычислительного ресурса на обработку задач минимальны, а загрузка оконечных узлов сети наиболее близка к равномерной.

Литература

1. Біла книга 2011. Збройні Сили України [Текст]. – К.: Міністерство оборони України, 2012. – 22 с.

2. Топорков, В.В. Модели распределенных вычислений [Текст] / В.В. Топорков. – М.: ФИЗМАТЛИТ, 2004. – 320 с.

3. *Гофф, М.К. Сетевые распределенные вычисления: Достижения и проблемы [Текст] / М.К. Гофф. – М.: КУДИЦ-Образ, 2005. – 320 с.*

4. Таненбаум, Э. Распределенные системы. Принципы и парадигмы [Текст] / Э. Таненбаум, М. ван Стеен. – СПб.: Питер, 2003. – 877 с.

5. Андрианов, С.Н. Параллельные и распределенные вычисления [Текст] / С.Н. Андрианов, А.Б. Дегтярев. – СПб.: Изд-во Санкт-Петербургского университета, 2007. – 61 с.

6. Кучук, Г.А. Лианеризация задачи минимизации суммарных затрат сетевого ресурса гетерогенной вычислительной сети [Текст] / Г.А. Кучук, А.В. Петров, С.М. Балакирева // Системи управління, навігації та зв'язку. – К.: ЦНДІ НіУ, 2012. – №2 (22). – С. 111 – 114.

7. Кучук, Г.А. Математическая модель функционирования специализированного программного комплекса в среде гетерогенной мультисервисной сети [Текст] / Г.А. Кучук, А.В. Петров, Р.В. Королев // Системи обробки інформації. – Х.: ХУПС, 2012. – №3 (101). – С. 191 – 199.

8. Жолобов, Д.А. Введение в математическое программирование [Текст] / Д.А. Жолобов. – М.: МИФИ, 2008. – 376 с.

9. Bazaraa, M.S. Linear Programming and Network Flows, fourth edition [Text] / M.S. Bazaraa, J.J. Jarvis, H.D. Sherali. – John Wiley & Sons, Inc. – 2010. – 748 p.

Поступила в редакцию 22.08.2012

Рецензент: д-р техн. наук, с.н.с., профессор кафедры конструкции и технического обслуживания летательных аппаратов и двигателей Е.А. Украинец, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков, Украина.

ЗАСТОСУВАННЯ МЕТОДУ ВЕКТОРУ СПАДУ ПІД ЧАС ІНСТАЛЯЦІЇ СПЕЦІАЛІЗОВАНОГО ПРОГРАМНОГО КОМПЛЕКСУ В СЕРЕДОВИЩІ ГЕТЕРОГЕННОЇ МУЛЬТИСЕРВІСНОЇ МЕРЕЖІ

О.В. Петров

Аналізуються методи розв'язання задач лінійного булева програмування. Наведені числові оцінки машинного експерименту дослідження алгоритмів булева програмування. Спираючись на результати машинного експерименту запропоновано метод із найбільш ефективним алгоритмом реалізації задачі на ЕОМ, який дозволяє врахувати специфіку розглядаємої задачі. Отримано квазіоптимальне рішення задачі оптимізації структури гетерогенної мультисервісної мережі. Наведено метод інсталяції спеціалізованого програмного комплексу в середовищі гетерогенної мультисервісної мережі.

Ключові слова: гетерогенні мультисервісні мережі, спеціалізований програмний комплекс, булеве програмування, метод вектору спаду.

APPLICATION OF VECTOR OF SLUMP AT INSTALATION OF THE SPECIALIZED PROGRAMMATIC COMPLEX IN ENVIRONMENT OF HETEROGENEOUS COMPUTER MULTISERVICE NETWORK

O.V. Petrov

The methods of decision of tasks are analyzed linear boolean programming. Numeral estimations over of machine experiment of research of algorithms are brought boolean programming. On the basis of results of machine experiment a method offers with the most effective algorithm of realization of task on COMPUTER. The decision of task of optimization of heterogeneous multiservice network structure is got. The method of installation of the specialized programmatic complex is presented in the environment of heterogeneous computer multiservice network.

Key words: heterogeneous computer multiservice networks, specialized programmatic complex, boolean programming, method of vector of slump.

Петров Алексей Валерьевич – научный сотрудник научно-исследовательского отдела научного центра Воздушных Сил, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков, Украина.