

УДК 681.3.06:51

**В.С. Горин****ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ  
ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ КОМПОНОВКИ**

*Рассмотрены возможные способы решения задачи компоновки несвязанных подсхем в блоки с заданными контактными и модульными ограничениями. Приведены формальные постановки задачи и предложены точный и приближенные алгоритмы решения, основанные на вычислительной схеме метода динамического программирования.*

**Ключевые слова:** алгоритм, компоновка, формальная постановка задачи, динамическое программирование.

**Введение.** В результате работы последовательных алгоритмов компоновки в силу локального пошагового характера оптимизации и эвристических способов разрешения неоднозначных ситуаций возможно появление некоторого множества не связанных между собой подсхем, модульные и контактные емкости которых невелики. Обычно их пытаются включить во вновь формируемые блоки, что приводит, с одной стороны, к низкой плотности заполнения блоков, а с другой – к усложнению последующих задач размещения и трассировки.

Подобная задача может быть весьма актуальной в случае компоновки типовых элементов замены или выделения из общей схемы устройства функционально законченных узлов.

**Целью настоящей работы** являются формулировка возможных постановок задачи объединения небольших по объему подсхем в отдельные конструктивные блоки и реализация комбинаторных алгоритмов, позволяющих найти оптимальное или приближенное решение этой задачи.

**Формальная постановка задачи.** Обозначим через  $Q = \{q_1, q_2, \dots, q_m\}$  – множество не связанных между собой подсхем, модульные  $s_i$  и контактные  $p_i$  емкости которых удовлетворяют условиям:

$$s_i < \bar{s}, p_i < \bar{p}, i = 1, 2, \dots, m,$$

где  $\bar{s}$  и  $\bar{p}$  – соответственно модульное и контактное ограничения формируемых блоков.

Для определения минимального количества конструктивных блоков, реализующих эти подсхемы, можно воспользоваться точным алгоритмом, предложенным в [1]. При этом формальная постановка задачи будет выглядеть следующим образом:

$$\text{минимизировать } \sum_{j=1}^r \sum_{i=1}^m p_i \cdot x_{ij} \quad (1)$$

при ограничениях:

$$\sum_{i=1}^m p_i \cdot x_{ij} \leq \bar{p}, j = 1, 2, \dots, r; \quad (2)$$

$$\sum_{i=1}^m s_i \cdot x_{ij} \leq \bar{s}, j = 1, 2, \dots, r; \quad (3)$$

$$\sum_{j=1}^r x_{ij} = 1, \quad (4)$$

$$x_{ij} \in \{0; 1\}; i = 1, 2, \dots, m; j = 1, 2, \dots, r.$$

Здесь  $r$  – число конструктивных блоков, требуемых для реализации подсхем из множества  $Q$ , а  $x_{ij}$  – псевдобулевы переменные, определяющие принадлежность  $q_i$  – й подсхемы  $j$  – му блоку:

$$x_{ij} = \begin{cases} 1, & \text{если } q_i \text{ – я подсхема реализуется} \\ & j \text{ – м блоком;} \\ 0 & \text{– в противном случае.} \end{cases}$$

Наименьшее количество блоков  $r_{\min}$ , требуемых для реализации схемы, можно приблизительно оценить следующим образом:

$$r_{\min} = \text{round} \left( \sum_{i=1}^m s_i / \bar{s} \right),$$

где  $\text{round}(X)$  равно наименьшему целому, большему или равному  $X$ .

Задавая последовательно  $r = r_{\min}, r_{\min} + 1, r_{\min} + 2, \dots, r_{\max}$ , можно найти допустимые решения задачи. Решение, при котором  $r = \min_k(r_k)$ , будет оптимальным.

Из-за малой размерности данная задача значительно проще задачи [1], однако многократное повторение решения для различных значений  $r$

представляет существенную сложность.

Рассмотрим приближенное решение поставленной задачи, для чего на каждом шаге будем выделять из всего множества подсхем  $Q$  множество подсхем максимального объема [2].

Обозначим через  $Q^j = \{q_1, q_2, \dots, q_{m_j}\}$  множество подсхем, не распределенных в ранее сформированные конструктивные блоки к моменту формирования  $B^j$ -го блока. Введем булевы переменные  $x_i$  ( $i = 1, 2, \dots, m_j$ ), определяющие принадлежность  $q_i$ -й подсхемы  $B^j$ -му блоку. Тогда указанная выше задача может быть сведена к задаче линейного целочисленного программирования с булевыми переменными или «многомерной задаче о ранце» [3] и сформулирована следующим образом:

найти максимум целевой функции

$$\sum_{i=1}^{m_j} s_i \cdot x_i \quad (5)$$

при ограничениях

$$\sum_{i=1}^{m_j} p_i \cdot x_i \leq \bar{p}; \quad (6)$$

$$\sum_{i=1}^{m_j} s_i \cdot x_i \leq \bar{s}; \quad (7)$$

$$x_i \in \{0; 1\}, \quad i = 1, 2, \dots, m_j. \quad (8)$$

Здесь все параметры  $\bar{s}$ ,  $\bar{p}$ ,  $s_i$  и  $p_i$  принимают целочисленные значения.

При неэффективности ограничения (7) данную задачу можно представить как известную задачу «об одномерном ранце» и воспользоваться для ее решения методом динамического программирования [4].

Будем считать, что текущее состояние процесса формирования  $B^j$ -го блока задачи (5), (6), (8) определяется количеством свободных контактов  $z_k$  ( $k = 0, 1, 2, \dots, m_j$ ), и перед  $i$ -м шагом оно определяется следующим образом:

$$z_{i-1} = \bar{p} - \sum_{k=1}^{i-1} p_k \cdot x_k. \quad (9)$$

Тогда  $z_0$  будет начальным состоянием, которому соответствует величина  $\bar{p}$  – контактное ограничение формируемого блока.

Управление на  $i$ -м шаге означает присваивание двоичной переменной  $x_i$  значения 1 или 0, что соответствует либо включению  $q_i$ -й подсхемы в  $B^j$ -й блок, либо нет. Причем допустимость управления, устанавливающего  $x_i = 1$ , определяется условием:

$$z_i = z_{i-1} - p_i \cdot x_i = \bar{p} - \sum_{k=1}^i p_k \cdot x_k \geq 0. \quad (10)$$

Выполнение (или невыполнение) условия (10) позволяет получить приращение целевой функции на  $i$ -м шаге:

$$f_i(z_i) = s_i \cdot x_i. \quad (11)$$

Просуммировав приращения целевой функции по всем  $i = 1, 2, \dots, m_j$ , получим оценку допустимого решения задачи.

Поставим в соответствие множеству состояний процесса формирования  $B^j$ -го блока целочисленную переменную  $z$  ( $0 \leq z \leq \bar{p}$ ) и рассмотрим оптимальное решение задачи (5), (6), (8) в зависимости от переменной  $z$ :

$$f_k(z) = \max \sum_{i=1}^k s_i \cdot x_i, \quad i \leq k \leq m_j, \quad (12)$$

где 
$$\sum_{i=1}^k p_i \cdot x_i \leq z, \quad (13)$$

$$x_i \in \{0; 1\}, \quad i = 1, 2, \dots, m_j.$$

Пусть для каждого значения  $z$  уже определены значения переменных  $x_1(z), x_2(z), \dots, x_{k-1}(z)$ , т.е. решена следующая задача:

$$f_{k-1}(z) = \sum_{i=1}^{k-1} s_i \cdot x_i \rightarrow \max, \quad (14)$$

при 
$$\sum_{i=1}^{k-1} p_i \cdot x_i \leq z, \quad (15)$$

$$x_i \in \{0; 1\}, \quad i = 1, 2, \dots, (k-1). \quad (16)$$

Тогда значение переменной  $x_k(z)$  определяется из условия

$$f_k(z) = \max_{x_k \in \{0; 1\}} [f_{k-1}(z - p_k \cdot x_k) + s_k \cdot x_k], \quad (17)$$

при этом

$$z - p_k \cdot x_k \geq 0, \quad k = 1, 2, \dots, m_j. \quad (18)$$

Очевидно,  $f_k(z) = 0$  при  $z < \min_i(p_i) = p_{\min}$  ( $i = 1, 2, \dots, m_i$ ).

С целью учета ограничения (7) при реализации алгоритма решения задачи следует исключить такие значения  $x_k$ , при которых нарушается условие  $f_k(z) \leq \bar{s}$ .

Соотношение (17) позволяет последовательно вычислить значения всех функций  $f_k(z)$  для  $p_{\min} \leq z \leq \bar{p}$ .

Определив функции  $f_k(z)$ , несложно вычислить требуемые компоненты оптимального плана  $X^* = \{x_1^*, x_2^*, \dots, x_{m_j}^*\}$  задачи (5) – (8).

Для решения поставленной задачи можно

применить численный метод [3], использующий выражение (17) и не требующий предварительного определения всех допустимых наборов подсхем, общие модульная и контактная емкости которых не превышают заданных. Однако учитывая, что переменные  $x_i$  ( $i=1,2,\dots,m_j$ ) могут принимать значения только 0 или 1, можно предложить более простую схему решения [5]. Для этого дадим другую постановку задачи.

Обозначим через

$$\varphi_k[r(k)] = \max \sum_{i=1}^k s_i \cdot x_i, \quad (19)$$

$$\begin{aligned} \varphi_0[r(0)] &= 0, \\ k &= 1, 2, \dots, m_j, \end{aligned} \quad (20)$$

где

$$r(k) = \sum_{i=1}^k p_i \cdot x_i, \quad (21)$$

$$\begin{aligned} r(0) &= 0, \\ k &= 1, 2, \dots, m_j. \end{aligned}$$

Отсюда

$$\begin{aligned} r(k) - r(k-1) &= p_k \cdot x_k; \\ r(0) &= 0; \\ k &= 1, 2, \dots, m_j. \end{aligned} \quad (22)$$

Рекуррентное соотношение (22) можно рассматривать как уравнение движения некоторой дискретной управляемой динамической системы, положение которой задается переменной  $r(k)$ , а переменная  $x_k$  определяет управляющее воздействие на систему. В дискретный момент времени  $k$  система скачком переходит из одного состояния в другое, а величина скачка определяется значением переменной  $x_k$ .

Сформулируем задачу (5) – (8) следующим образом: определить оптимальное управление  $X^* = \{x_1^*, x_2^*, \dots, x_{m_j}^*\}$ , обеспечивающее перевод динамической системы (22) из положения  $r(0)$  в конечное положение

$$r(m_j) \leq \bar{p} \quad (23)$$

и доставляющее максимум целевой функции (19) при условиях (7), (8).

Несложно показать, что рекуррентное соотношение (17), соответствующее вычислительной схеме метода, для данной постановки задачи будет иметь вид:

$$\begin{aligned} \varphi_k[r(k)] &= \\ &= \max \left\{ \varphi_{k-1}[r(k) - p_k \cdot x_k] + s_k \cdot x_k \right\}. \end{aligned} \quad (24)$$

Решение задачи будем осуществлять в два

этапа. На первом этапе построим все возможные допустимые состояния системы, а на втором будем искать оптимальный план задачи  $X^*$ .

Рассмотрим шаг  $k=1$ .

Согласно (21) и (22) переменная  $r(1)$  может принимать только два значения:

$$r(1) = \begin{cases} 0 & \text{при } x_1 = 0, \\ p_1 & \text{при } x_1 = 1. \end{cases}$$

Целевая функция при этом тоже может иметь только два значения (19), (20), (21):

$$\begin{aligned} \varphi_1^{(1)}[r^{(1)}(1)] &= 0; \\ \varphi_1^{(2)}[r^{(2)}(1)] &= s_1. \end{aligned}$$

Запомним значения целевой функции и соответствующие им координаты векторов  $\overline{r(1)}$  и  $\overline{x_1}$ , перейдем к шагу  $k=2$ .

Координаты вектора состояний  $\overline{r(2)}$  определяются согласно (22) следующим образом:

$$\begin{aligned} r^{(1)}(2) &= 0; \\ r^{(2)}(2) &= p_1; \\ r^{(3)}(2) &= p_2 \text{ и} \\ r^{(4)}(2) &= p_1 + p_2, \text{ если } p_1 + p_2 \leq \bar{p} \text{ и } s_1 + s_2 \leq \bar{s}. \end{aligned}$$

С помощью выражения (24) с учетом (7) найдем значения  $\varphi_2[\overline{r(2)}]$ ,  $\overline{r(2)}$  и  $\overline{x_2}$ .

Рассмотрим  $t$ -й шаг вычислений.

По соотношению (22) найдем допустимые состояния системы  $\overline{r(t)}$  и определим значения  $\varphi_t[\overline{r(t)}]$ . При этом:

$$\begin{aligned} \varphi_t^{(1)}[r^{(1)}(t)] &= 0, \quad \overline{x_t} = \vec{0}; \\ \varphi_t^{(i)}[r^{(i)}(t)] &= \max \left\{ \varphi_{t-1}^{(i)}[r^{(i)}(t) - p_t \cdot x_t] + \right. \\ &\quad \left. + s_t \cdot x_t \right\}, \quad 2 \leq t \leq m_j. \end{aligned} \quad (25)$$

Рассмотрим результаты последнего шага этапа построения функций  $\varphi_k[\overline{r(k)}]$ .

Среди всех значений  $\varphi_{m_j}[\overline{r(m_j)}]$  найдем

$$\varphi_{m_j}^*[r^*(m_j)] = \max_i \left\{ \varphi_{m_j}^{(i)}[r^{(i)}(m_j)] \right\},$$

который достигается при наименьшем значении  $r^*(m_j)$ .

Очевидно,  $\varphi_{m_j}^*[r^*(m_j)]$  – значение целевой функции (5), соответствующее оптимальному решению. В это состояние система попадает с

предпоследнего шага под действием управления  $x_{m_j}^* \in X^*$ . Осуществляя «попятное» движение, определяем

$$\begin{aligned} \varphi_{m_{j-1}}^* [r^*(m_j) - p_{m_j} \cdot x_{m_j}^*] &= \\ &= \varphi_{m_j}^* [r^*(m_j)] - s_{m_j} \cdot x_{m_j}^* \end{aligned}$$

и соответственно  $x_{m_{j-1}}^*$ . Продолжая движение далее, дойдем до  $\varphi_0[r(0)] = 0$  и определим тем самым все компоненты оптимального плана  $X^*$  задачи.

**Алгоритм**

- П.1.  $j := 1$ .
- П.2.  $k := 1$ .
- П.3. Заполняем таблицы, соответствующие значениям координат векторов состояний  $\overline{r(k)}$ , управлений  $\overline{x_k}$  и функций  $\varphi_k[\overline{r(k)}]$ , полагая  $\overline{x_k} = (0; 1)$ :

$$\begin{aligned} r^{i(1)}(k) &= r^i(k-1); \\ r^{i(2)}(k) &= r^i(k-1) + p_k, \text{ если } r^{i(2)}(k) \leq \overline{p}. \end{aligned}$$

Одновременно для каждой координаты вектора  $\overline{r(k)}$  вычисляем значение  $\varphi_k^{(i)}[\overline{r(k)}]$  и оставляем только те управления, которые доставляют максимум этой функции.

- П.4. Проверяем:
  - если  $k = m_j$ , то перейти к п.5,
  - иначе  $k := k + 1$ ; перейти к п.3.

- П.5.  $l := 0$ .
- П.6. Если  $m_j - l = 0$ , то перейти к п.9.
- П.7. Определяем значение функции

$$\varphi_{m_{j-1}}^* [r^*(m_j - l)] = \max_i \left\{ \varphi_{m_{j-1}-l}^i [r^i(m_j - l)] \right\}$$

и соответствующие ему  $x_{m_{j-1}}^*$  и  $r^*(m_j - l)$ .

- П.8.  $l := l + 1$ . Перейти к п.6.
- П.9. Исключаем из множества  $Q^j = \{q_1, q_2, \dots, q_{m_j}\}$  подмножество подсхем, вошедших в сформированный блок. Получаем новое множество

$$Q^{j+1} = \{q_1, q_2, \dots, q_{m_{j-1}}\}.$$

- П.10.  $j := j + 1$ ;  $m_j := m_{j-1}$ .
- П.11. Если  $Q^j \neq \emptyset$ , перейти к п.2.
- П.12. Конец.

**Экспериментальная часть.** Рассмотрим пример, иллюстрирующий предложенный метод компоновки не связанных между собой подсхем.

Пусть задано множество  $Q^1 = \{1, 2, \dots, 8\}$ , состоящее из 8 подсхем, модульные  $s_i$  и контактные  $p_i$  емкости которых заданы таблицей 1.

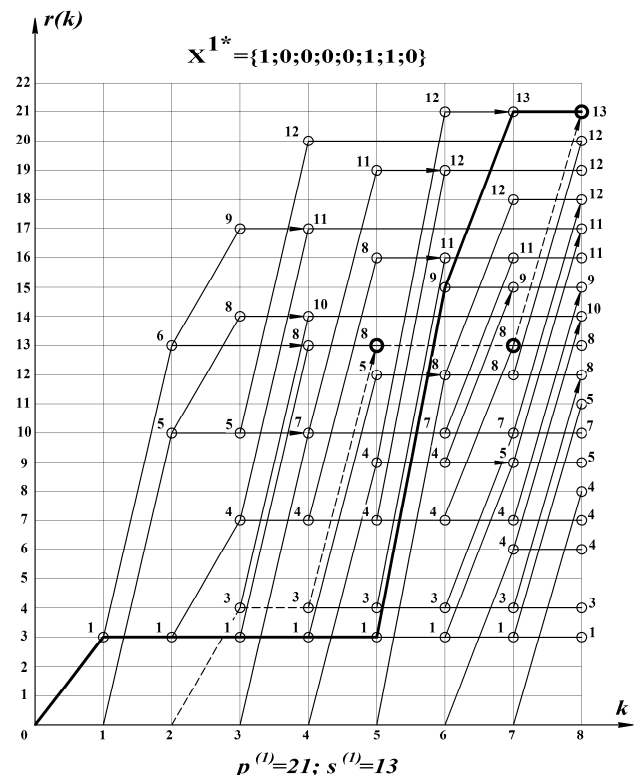
**Таблица 1– Модульные и контактные емкости подсхем из множества  $Q^1$**

$q_i$	1	2	3	4	5	6	7	8
$s_i$	1	5	3	7	4	8	4	4
$p_i$	3	10	4	10	9	12	6	8

Ограничения на формируемые блоки:

$$\overline{p} = 21; \overline{s} = 13.$$

Процесс решения примера изображен графически на рисунках 1 и 2. Горизонтальные стрелки на рисунках соответствуют «поглощению» неперспективных решений более перспективными, а наклонные стрелки – прерыванию неперспективных путей поиска решения.



**Рисунок 1 – Формирование первого блока**

Значения функций  $\varphi_k[\overline{r(k)}]$  показаны над соответствующими координатами вектора состояний  $\overline{r(k)}$ , а процесс поиска оптимальных планов  $X^{j*}$  решений выделен жирными линиями.

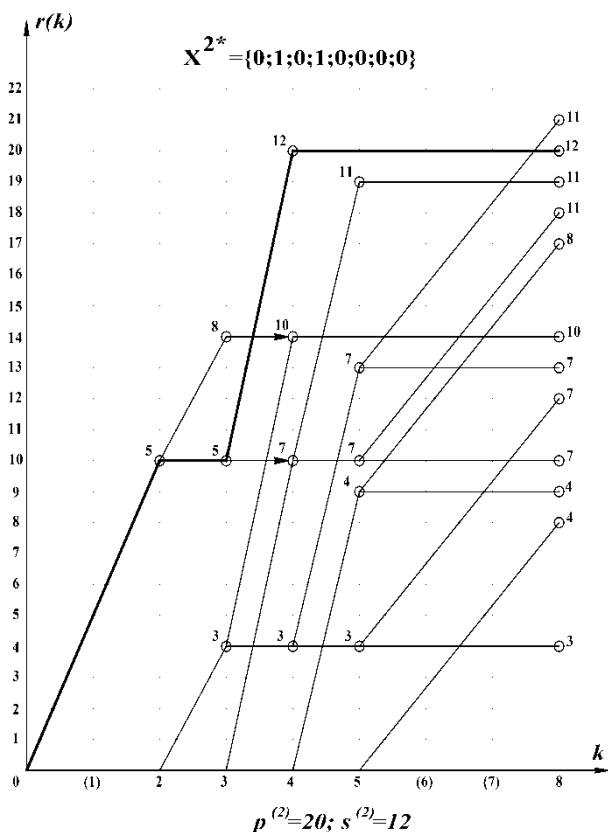


Рисунок 2 – Формирование второго блока

В результате двукратного решения задачи получены два конструктивных блока, имеющих следующий состав и параметры:

$$B^{(1)} = \{1, 6, 7\}; p^{(1)} = 21, s^{(1)} = 13;$$

$$B^{(2)} = \{2, 4\}; p^{(2)} = 20, s^{(2)} = 12.$$

Очевидно,

$$B^{(3)} = Q \setminus (B^{(1)} \cup B^{(2)}) = \{3, 5, 8\}; p^{(3)} = 21, s^{(3)} = 11.$$

В работе [3] отмечено, что для задачи об одномерном ранце с целочисленными значениями параметров все целочисленные решения, в том числе оптимальные, можно найти за один просчет.

Действительно, на рисунке 1 помимо первого конструктивного блока можно отыскать второй с оценкой 12, содержащий вторую и четвертую подсхемы, а также третий, прорисованный штриховой линией. Но из этого же рисунка видно, что некоторые вершины маршрута, соответствующие третьему блоку (они помечены широкими кольцевыми поясками), имеют оценки, отличные от фактических оценок для третьего блока. Следовательно, при решении задачи путем однократного просчета необходимо сохранять в памяти оценки для всех сочетаний элементов, т.е. нельзя сокращать перебор.

Кроме того, несложно показать, что предложенный алгоритм при поиске одного решения имеет линейные оценки трудоемкости  $2\bar{p} \cdot m_j$  и

требуемой памяти  $2(\bar{p} + 1) \cdot m_j$  [3], что обуславливает возможность решения задач практической размерности.

Замечу также, что схема поиска решения на рисунке 2 несколько отличается от предложенного алгоритма (в п.9 подсхемы, вошедшие в найденное решение, не исключаются из множества  $Q^j$  – им условно присваивают нулевые модульную и контактную емкости, а в п.10 мощность множества  $Q^j$  не изменяют). Это сделано для удобства ручного просчета и наглядности процесса формирования оптимального решения.

**Заключение.** В работе сформулированы возможные математические постановки и предложен точный и приближенные алгоритмы решения задачи компоновки блоков из небольших подсхем, образующихся при решении задачи компоновки узлов из функционально-логических и принципиальных схем электронной аппаратуры большой размерности в минимальное количество узлов с заданными модульным и контактным ограничениями или максимальным числом функционально законченных узлов одинаковых типов. Следует отметить, что в множество подсхем  $Q^j$  можно включать также скомпонованные в процессе работы САПР узлы, модульная и контактная емкости которых меньше допустимых.

Рекурсивность метода динамического программирования позволяет реализовать рассмотренный алгоритм в виде рекурсивной процедуры, что делает программу весьма компактной и эффективной.

Приведенный пример, а также проверка работы алгоритма на достаточно большом наборе реальных схем показали его хорошую эффективность.

**Библиографический список**

1. Гайфуллин Э.Ш., Егоров А.Г., Юсим Г.В. Оптимальное разбиение функционально-логических схем с учетом контактных ограничений. – Электронная техника. Серия 3. Микроэлектроника, 1975. Вып. 2. С. 56-58.
2. Горин В.С., Чикин В.А., Юсим Г.В. Подсистема компоновки ТЭЗов. Л.: Изв. ЛЭТИ, 1980. Вып. 266. С. 89-92.
3. Сигал И.Х., Иванова И.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: учеб. пособие. – М.: ФИЗМАТЛИТ, 2002. – 240 с.
4. Таха, Хемди А. Введение в исследование операций, 7 издание: пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 912 с.
5. Мухачева М.А., Рубинштейн Г.Ш. Математическое программирование. – Новосибирск: «Наука», 1987. – 274 с.