

УДК:519.854.3

**ПРАКТИЧЕСКИЕ АСПЕКТЫ РЕАЛИЗАЦИИ
ЭФФЕКТИВНОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ
ОПТИМАЛЬНОЙ КОМПЛЕКТАЦИИ**

Ю.А. Мезенцев

*Новосибирский государственный технический университет
mesyan@yandex.ru*

Предложен численный метод решения общей задачи оптимальной комплектации, относящейся к классу NP-трудных задач дискретного программирования с булевыми переменными. Метод основан на процедурах линеаризации исходных условий, лагранжевой декомпозиции и итеративного параметрического анализа последовательности порождаемых подзадач. Показано, что алгоритм является эффективным и асимптотически точным по размерности задачи оптимальной комплектации.

Ключевые слова: целочисленное программирование, декомпозиция, параметрический анализ, эффективный алгоритм, задача оптимальной комплектации.

Введение

Настоящая статья является продолжением и логическим следствием работы [1], в которой исследованы свойства общей задачи оптимальной комплектации (ЗОК), относящейся к классу NP-трудных задач дискретного программирования с булевыми переменными, а также предложены процедуры линеаризации исходных условий, лагранжевой декомпозиции и итеративного параметрического анализа последовательности порождаемых подзадач. Там же изложены основные принципы, положенные в основу эффективного алгоритма решения задачи оптимальной комплектации, явившегося предметом настоящей статьи.

1. Формальная постановка задачи

Рассмотрим декомпозированную и линеаризованную ЗОК (ЛЗОК) [1].

$$Q_j^l = \sum_{v \in V_j^l} q_{j,v}^l x_{j,v}^l \rightarrow \max \quad \forall j \in J, l = \overline{1, L_j}, \quad (1.1)$$

$$\sum_{v \in V_j^l} p_{j,v}^l x_{j,v}^l \leq P_j^l \quad \forall j \in J, l = \overline{1, L_j}, \quad (1.2)$$

Работа выполнена при финансовой поддержке АВИЦП 8.536.2011 «Разработка интеллектуальных технологий, средств компьютерного моделирования и эффективных методов оптимизации как функционального наполнения информационно-аналитических систем поддержки принятия решений».

$$\underline{y}_j^l \leq \sum_{v \in V_j^l} x_{j,v}^l \leq \bar{y}_j^l \quad \forall j \in J, l = \overline{1, L_j}, \quad (1.3)$$

$$0 \leq x_{j,v}^l \leq 1 \quad \forall v \in V_j^l \quad j \in J, l = \overline{1, L_j}, \quad (1.4)$$

$$x_{j,v}^l \in \{0 \vee 1\} \quad \forall v \in V_j^l \quad j \in J, l = \overline{1, L_j}, \quad (1.5)$$

$$Q_{\text{об}} = \sup_l \left\{ \sum_{j \in J^l} Q_j^l(P_j^l) \right\} \rightarrow \max, \quad (1.6)$$

$$\sum_{j \in J^l} P_j^l \leq P_{\text{об}}, \quad (1.7)$$

$$P_j^l \geq 0, \quad j \in J, l = \overline{1, L_j}, \quad (1.8)$$

$$P_{\min} \leq P_{\text{об}} \leq P_{\max}. \quad (1.9)$$

Здесь $q_{j,v}^l$ – оценка качества комплекта v подсистемы j на шаге l ; $Q_j^l = Q_j$ – качество подсистемы j на шаге l ; $p_{j,v}^l$ – оценка затрат на комплект v подсистемы j на шаге l ; P_j^l – затраты на комплектацию подсистемы j на шаге l ; \widehat{J} – число выделяемых подсистем в комплектуемой системе.

$x_{j,v}^l$ – переменные выбора комплекта v подсистемы j на шаге l ,

$$x_{j,v}^l = \begin{cases} 1, & \text{если выбирается } v\text{-вариант комплектации} \\ & j\text{-й подсистемы на } l\text{-м шаге и его носитель,} \\ 0 & \text{в противном случае.} \end{cases}$$

V_j^l – множество альтернативных вариантов комплектации подсистемы j на шаге l ;

L_j – максимальное число интервалов ФПП подсистемы j .

В соответствии с разработанными положениями [1] задача (1.1)–(1.9) линеаризована и разбита на подзадачи. (1.6)–(1.9) являются координирующей задачей. Условия (1.1)–(1.4) определяют совокупность локальных подзадач со снятыми требованиями целочисленности (1.5).

Приведем алгоритм поиска оптимальных решений ЛЗОК, основанный на лагранжевой декомпозиции [1, 2, 3].

2. Декомпозиционный алгоритм решения задач оптимальной комплектации

Пусть $\lambda_j(P_j)$ – функция предельной полезности (ФПП) затрат P_j подзадач ЛП (1.1)–(1.4) и $\lambda(P_{06})$ – ФПП затрат на комплектацию системы в целом [1, 2, 4]. Предварим описание искомого алгоритма описанием его компоненты – процедуры построения ФПП $\lambda_j(P_j)$ для произвольной подсистемы j , $j \in J$, $l = \overline{1, L_j}$.

Алгоритм A_1

1. Ввод исходных данных (положим для подсистемы j величины P_j^{\min} , P_j^{\max} , $c_{j,v}^l$, $q_{j,v}^l$, $v \in V_j^l$ известными), определяем малую константу $\varepsilon > 0$. Рассматриваем в качестве неизвестного параметра величину P_j с начальным значением P_j^{\min} ($P_j := P_j^{\min}$), присваиваем значение номера шага алгоритма $l := 0$.

2. Увеличиваем номер шага $l := l + 1$.

3. Посредством любого координатного алгоритма находим решение подзадачи (1.1)–(1.4). Запоминаем решение $x_{j,v}^l$ и двойственную оценку $\lambda_j^l(P_j^l)$ (элемент ФПП на шаге l).

4. Находим интервал устойчивости параметра P_j , на котором найденное решение остается оптимальным. Правую и левую границы интервала обозначим соответственно как P_j^l и P_j^{l+1} , ($P_j^1 \equiv P_j^{\min}$).

5. Проверка на окончание алгоритма. Если $\lambda_j^l(P_j^l) = 0$, ФПП подсистемы j построена, $P_j^{\max} := P_j^l$, переход к п. 7. Если нет, следующий пункт.

6. Увеличиваем значение параметра в правой части ограничений (1.2) и (1.3) $P_j := P_j^{l+1} + \varepsilon$, выходя за правую границу текущего интервала устойчивости. Переход к п. 2.

7. Конец алгоритма.

В результате работы A_1 формируется ФПП затрат на комплектацию подсистем j , соответствующие параметрические решения и функция качества комплектации подсистемы j . Пример такой функции представлен в табл. 2.1.

Здесь $\lambda_j(P_j)$ – невозрастающая функция, т. е. $\lambda_j^{\max} \geq \dots \geq \lambda_j^l \geq \dots > 0$ и λ_j^l – константы, $l = \overline{1, L_j}$; $x_{j,v}^l(P_j)$ – параметрические решения на интервале l ; $Q_j^l(P_j)$ – значения критерия (1.1) на интервале l , $l = \overline{1, L}$.

Таблица 2.1

ФПП затрат и функция качества комплектации подсистемы

P_j	$P_j^{\min} < P_j \leq P_j^2$...	$P_j^l < P_j \leq P_j^{l+1}$...	$P_j > P_j^{\max}$
$\lambda_j(P_j)$	λ_j^{\max}	...	λ_j^l	...	0
$x_{j,v}^l$	$x_{j,v}^1(P_j)$...	$x_{j,v}^l(P_j)$...	$x_{j,v}^L(P_j)$
$Q_j(P_j)$	$Q_j^1(P_j)$...	$Q_j^l(P_j)$...	$Q_j^L(P_j)$

Алгоритм A_1 позволяет за полиномиальное число шагов построить локальные ФПП всех подсистем комплектуемой системы. Тем самым реализуется декомпозиционная схема решения ЗОК, завершает которую алгоритм построения общей ФПП затрат на комплектацию и оптимизации комплектаций по критерию качества, т. е. решения задачи (1.6)–(1.9).

Алгоритм A_2

1. Ввод исходных данных (положим величины P_j^{\min} , P_j^{\max} , $P_{об}$, $c_{j,v}^l$, $q_{j,v}^l$ $\forall j \in J^l$, $l = \overline{1, L}$, $v \in V_j^l$, \underline{y}_j^l , \bar{y}_j^l известными).

2. Для всех подсистем j осуществляется построение ФПП затрат на комплектацию $\lambda_j(P_j)$ $\forall j \in J^l$, $l = \overline{1, L}$ посредством алгоритма A_1 .

3. Определяем минимальные и максимальные затраты на комплектацию системы $P^{\min} = \sum_{j \in J^L} P_j^{\min} \underline{y}_j$ и $P^{\max} = \sum_{j \in J^L} P_j^{\max} \bar{y}_j$.

4. Строим общую ФПП $\lambda(P)$ системы в интервале $P^{\min} \leq P \leq P^{\max}$ посредством объединения и упорядочения по убыванию значений $\lambda_j(P_j)$ частных ФПП. Этот результат достигается объединением $\lambda_j(P_j)$ и применением любого алгоритма сортировки [5].

5. На основе общей ФПП определяем оптимальные комплектации $x_j^*(P)$ и $Q(P)$ в каждой седловой точке (точке разрыва ФПП).

6. Для произвольно задаваемого значения правой части ресурсного ограничения $P_{об}$ из допустимого диапазона P находим замыкающий интервал $\lambda(P_{об})$, т. е. интервал аргумента ФПП, содержащий $P_{об}$, но не содержащий разрывов, соответствующих седловым точкам функции Лагранжа. P_B и P_H – его верхняя и нижняя границы соответственно, т. е. $P_H \leq P_{об} \leq P_B$.

7. Определяем подсистему j , которой соответствует интервал $[P_H, P_B]$. Находим интервал $[P_{H,j}^s, P_{B,j}^s]$ (s – его порядковый номер) и значение затрат $P_{об,j}$ в

пересчете для подсистемы j . На интервале $[p_{н,j}^s, p_{в,j}^s]$ находим ближайшее слева к $P_{об,j}$ значение координаты P_j^* дискретной функции качества $Q_j(P_j)$. Найденные координаты P_j^* и $Q_j(P_j^*)$ полностью идентифицируют приближенное решение задачи оптимальной комплектации.

7. Фиксируем найденное решение $x(P_{об})$ и $Q(P_{об})$. Конец алгоритма.

Представленный алгоритм позволяет находить точные решения при значениях P , определяющих седловые точки ФПП $\lambda(P)$ [1, 2, 3]. В других случаях оптимальность решений не гарантируется, однако точность алгоритма всегда можно оценить. Ниже приведены подобные оценки и представлены результаты экспериментального исследования быстродействия и точности алгоритма A_2 . Результаты работы алгоритма $\lambda(P)$, $x(P)$, $Q(P)$ представимы в виде таблиц, аналогичных табл. 2.1. Графическое отображение ФПП ($\lambda(P)$) и $Q(P)$ для условного примера ЛЗОК, содержащей пять подзадач (подсистем) от 15 до 22 вариантов комплектации каждой, представлено на рис. 2.1 и 2.2 соответственно. График на рис. 2.2 демонстрирует тот факт, что $Q(P)$ является кусочно-линейной выпуклой вверх функцией как для отдельных подсистем, так и в целом для любой ЗОК. Цифры над интервалами постоянной предельной полезности $\lambda(P)$ (рис. 2.1) отображают номера подсистем, которым соответствуют значения $\lambda(P)$ и $Q(P)$.

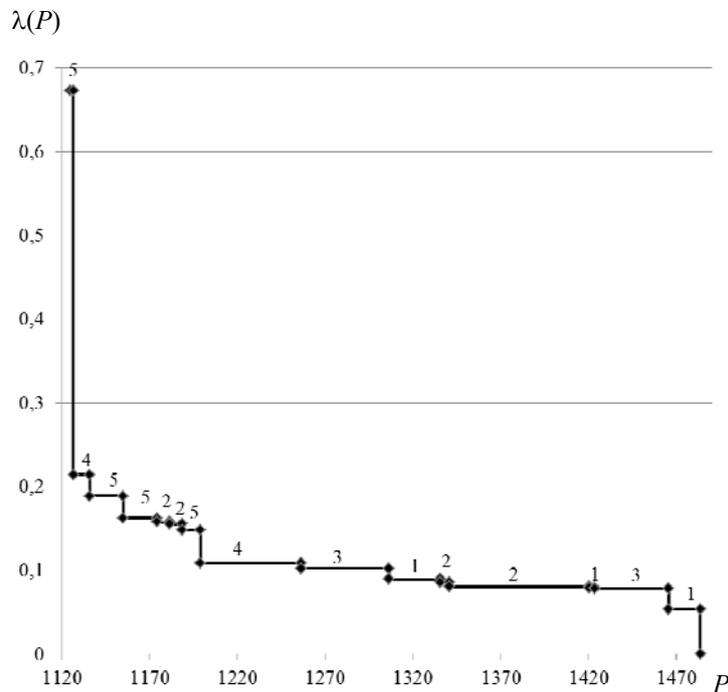


Рис. 2.1. Пример ФПП ЛЗОК

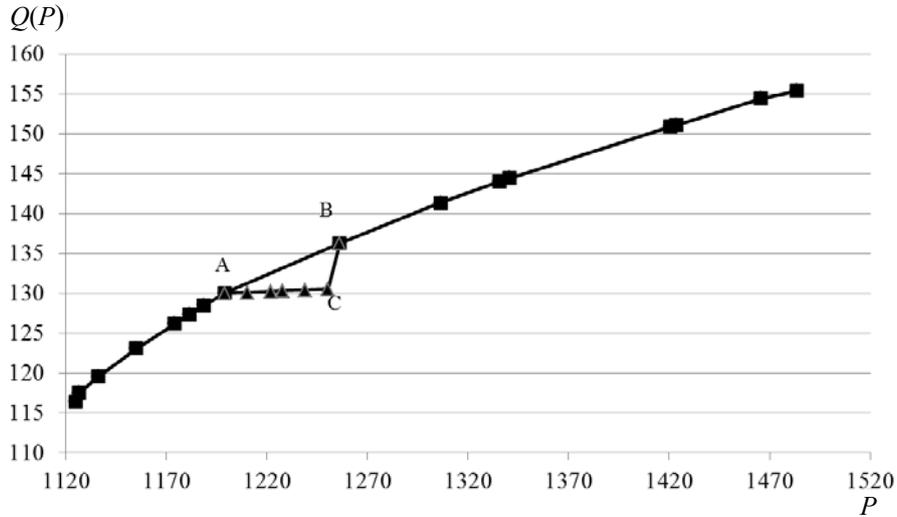


Рис. 2.2. Соотношение качество – цена (зависимость $Q(P)$)

3. Свойства алгоритма и его программной реализации

Оценим основные свойства декомпозиционного алгоритма решения ЛЗОК, в частности его быстродействие и близость синтезируемых вариантов комплектаций к оптимальным. Теоретические выкладки подкрепим анализом результатов вычислительных экспериментов с программной реализацией приведенных выше алгоритмов решения ЛЗОК.

Оценим максимально возможную теоретически ошибку алгоритма при поиске приближенного решения ЛЗОК. Измерять ошибку естественно в терминах критерия эффективности ЗОК Q .

Очевидно, что максимально возможная ошибка алгоритма δ реализуется при самой неблагоприятной ситуации: замещающий комплект замыкающей подсистемы (см. п. 7 алгоритма A_2) имеет оценку качества, бесконечно мало превышающую оценку левой границы (седловой точки), и оценку затрат, максимально превышающую оценку левой границы. Геометрическая интерпретация подобной ситуации отражена на рис. 2.2, где точки A и B соответствуют левой и правой границам интервала устойчивости $\lambda(P)$ и $Q(P)$ для заданного значения P в точке C. Максимальная возможная ошибка алгоритма на интервале $[P_A, P_B]$ в данном случае составит величину $\delta_{AB} = Q(P_B) - Q(P_A) - \varepsilon$. Здесь P_A и P_B – значения затрат в точках A и B соответственно, δ_{AB} – максимальная абсолютная возможная ошибка на интервале $[P_A, P_B]$, $\varepsilon > 0$ – малая величина.

Если j – номер замыкающей подсистемы, s – порядковый номер замыкающего интервала ФПП j -й подсистемы, то в общем случае δ вычисляется исходя из соотношений:

$$q_j^s(t) = a_{j,0}^s + a_{j,1}^s p_j^s(t), \tag{3.1}$$

где

$$a_{j,0}^s = \frac{q_{B,j}^s p_{H,j}^s - q_{H,j}^s p_{B,j}^s}{p_{H,j}^s - p_{B,j}^s}, \quad a_{j,1}^s = -\frac{q_{\epsilon,j}^s - q_{H,j}^s}{p_{H,j}^s - p_{\epsilon,j}^s}, \quad p_{H,j}^s \leq p_j^s(t) \leq p_{\epsilon,j}^s,$$

$q_{H,j}^s, q_{B,j}^s, p_{H,j}^s, p_{B,j}^s$ – соответственно значения функции качества и затрат на нижней и верхней границах интервала s подсистемы j . Выражение (3.1) определяет линейную границу $Q(P)$ на каждом из интервалов; $p_j^s(t)$ – линейная аппроксимация дискретной функции затрат на комплектацию для подсистемы j ; $q_j^s(t)$ – предельное значение качества для подсистемы j . Тогда максимальная ошибка алгоритма на интервале s для подсистемы j составит: $\delta_j^s = q_j^s(P_{об,j}) - Q_j(P_j^*)$.

И в целом абсолютная ошибка алгоритма: $\delta = \max_{j,s} \{\delta_j^s\}$.

Более информативным показателем является относительная ошибка Δ (предельный процент отклонения целевой функции на приближенном решении от оптимума). Поскольку ошибка может возникнуть только при определении комплектации замыкающей подсистемы, Δ естественно определить следующим образом:

$$\Delta = \delta_j^s \cdot 100 / \left(Q_{j-1}(P_{B,j-1}) + q_j^s(P_{об,j}) - Q_j(P_j^*) \right),$$

где $Q_{j-1}(P_{B,j-1})$ – значение функции качества на верхней границе интервала, непосредственно предшествующего замыкающему; $q_j^s(P_{об,j})$ – предельное значение качества для замыкающего интервала при заданных затратах.

Для рассмотренного примера получаем: $\delta = 5,625646$; $\Delta = 4,557253$.

Замечаем, что при самой неблагоприятной ситуации ошибка не превысила 5 % от возможного максимума. Фактическое же отклонение от оптимума для данного примера составляет 1,195 %.

Результаты тестирования программной реализации алгоритма отображены в ниже следующей таблице.

Таблица 3.1

Результаты тестирования алгоритма

Группа	Число тестов	Число подсистем	max число переменных п.с.	Всего булевых переменных	δ_{\max}	Δ_{\max} (%)	Время решения (с)
1	10	10	123	1000	0,096337	0,025226	0,000001
2	10	10	1230	10 000	0,236492	0,038604	0,000191
3	10	10	12300	100 000	0,024639	0,003960	0,187240

В табл. 3.1. приведены результаты решения 30 тестовых примеров, сгенерированных программой-генератором исходных данных ЛЗОК (1.1)–(1.9). Было сгенерировано три группы тестов разной размерности (по 10 тестов в группе с числом булевых переменных 1000, 10 000 и 100 000 соответственно).

Колонка «max число переменных п.с.» отображает максимальное число переменных выбора альтернативных вариантов комплектации одной из подсистем. Общее же число таких вариантов определяется количеством булевых переменных ЛЗОК (следующая графа).

В колонках « δ_{\max} » и « Δ_{\max} » отображены максимальные возможные абсолютные и относительные отклонения по всем десяти тестам каждой группы.

Из таблицы видно, что возможные отклонения от теоретических оптимумов крайне незначительны и уже при 1000 переменных пренебрежимо малы. С ростом числа булевых переменных растет и точность алгоритма A_2 , что подтверждает вывод об асимптотической точности последнего. Данный вывод хорошо иллюстрируется графиком $Q(P)$ для одного из тестов первой группы (рис. 3.1). Маркеры на рисунке соответствуют седловым точкам функции Лагранжа. С ростом размерности число седловых точек пропорционально возрастает. Поэтому возможное отклонение решений, определяемых алгоритмом A_2 , от оптимальных стремится к нулю. С учетом же быстродействия алгоритма (см. табл. 3.1) следует и вовсе парадоксальный вывод: чем выше размерность рассматриваемых задач оптимальной комплектации, тем точнее и быстрее (поскольку не требуют дополнительных усилий на уточнение результата) они решаются.

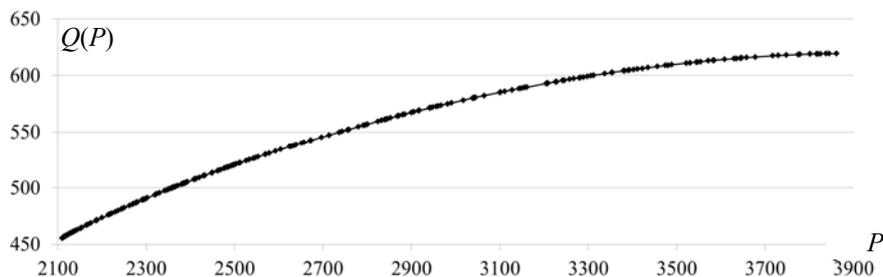


Рис. 3.1. Соотношение качество – цена ЛЗОК (1000 булевых переменных)

Разработанный программный комплекс нацелен на решение практических задач оптимальной комплектации в режиме реального времени и служит дополнением к созданному ранее инструментарию дискретной оптимизации [6, 7, 8, 9].

Заключение

Представленная работа фактически снимает проблему размерности для одного из ключевых классов задач оптимального проектирования. Несмотря на NP-сложность задач дискретной оптимизации, к которым сводятся рассмотренные задачи оптимальной комплектации, удалось построить эффективный алгоритм, позволяющий находить фактически точные их решения. Осуществлена программная реализация разработанного алгоритма, позволившая экспериментально подтвердить его полезные свойства.

ЛИТЕРАТУРА

- [1] Мезенцев Ю.А. Эффективный алгоритм решения одного класса задач целочисленного программирования / Ю.А. Мезенцев // Доклады Академии наук высшей школы РФ. – Новосибирск: Изд-во НГТУ, 2012. – № 2(19). – С. 42–53.
- [2] Мезенцев Ю.А. Декомпозиционный метод решения одного класса задач оптимального проектирования / Ю.А. Мезенцев // Научный вестник НГТУ. – Новосибирск, Изд-во НГТУ, 2006. – № 3(24). – С. 67–100.
- [3] Лэсдон Л.С. Оптимизация больших систем / Л.С. Лэсдон. – М.: Наука, 1975. – 432 с.
- [4] Бахтин А.Е. Математическое моделирование в экономике / А.Е. Бахтин. – Новосибирск: НГАЭиУ, 1995. – 196 с.
- [5] Кнут Д. Искусство программирования для ЭВМ, том 3: Сортировка и поиск / Д. Кнут. – М.: Мир, 1978. – 812 с.
- [6] Мезенцев Ю.А. Эффективный алгоритм целочисленного программирования / Ю.А. Мезенцев // Научный вестник НГТУ. – 2009. – № 2(35). – С. 91–114.
- [7] Мезенцев Ю.А. Метод бинарных отсечений и ветвлений целочисленного программирования / Ю.А. Мезенцев // Доклады Академии наук высшей школы РФ. – Новосибирск: Изд-во НГТУ, 2011. – № 1(16). – С. 12–25.
- [8] Мезенцев Ю.А. Реализация алгоритма решения специальных задач полуопределенного программирования с использованием IBM ILOG CPLEX / Ю.А. Мезенцев, П.С. Павлов // Научный вестник НГТУ. – Новосибирск: Изд-во НГТУ, 2011. – № 4(45). – С. 25–34.
- [9] Мезенцев Ю.А. К программной реализации декомпозиционного алгоритма решения одного класса задач дискретной оптимизации с полуопределенной релаксацией / Ю.А. Мезенцев, П.С. Павлов // Информационные технологии. – М.: Новые технологии, 2012. – № 2. – С. 54–59.

Mesentsev Y.A.

PRACTICAL ASPECTS OF THE IMPLEMENTATION OF AN EFFECTIVE ALGORITHM FOR SOLVING OPTIMAL COMPLETE SET PROBLEMS

A numerical method for solving general problem of optimal complete set related to the class of NP-hard problems of discrete programming with Boolean variables has been proposed. The method is based on procedures of linearization of the initial conditions, Lagrangian decomposition and iterative parametric analysis of sequences generated subproblems. It has been shown that the algorithm is effective and asymptotically accurate according to dimension of optimal complete set problems.

Keywords: Integer programming; Decomposition; Parametric analysis; Effective algorithm; Optimal complete set problem.

Статья поступила 13 июля 2012 г.