

# Комбинаторная оптимизация при построении конфигураций систем

Марк Ш. Левин

*Институт проблем передачи информации РАН, Российская академия наук,  
Москва, Россия  
email: mslevin@acm.org*

Поступила в редколлегию 10.11.2008

**Аннотация**—Важность и актуальность задач построения конфигураций систем возрастает во многих прикладных областях (например, программирование, электроника, производственные системы, коммуникации, логистика, стратегии решения задач, модульное планирование, технология материалов и комбинаторная химия). Обычно рассматриваются следующие типы прикладных систем/объектов, для которых могут исследоваться задачи конфигурации/реконфигуации: (а) исходные данные/информация (или исходные материалы); (б) системы обработки (машины, компьютеры, алгоритмы); (в) результирующие решения (продукты, планы); (г) прикладные системы поддержки (например, сервисы, сети); (д) системы требований; (е) стандарты. В данной статье рассматриваются несколько задач построения конфигураций систем: (i) поиск (отбор) множества (или структуры) системных компонентов, (ii) поиск множества совместимых системных компонентов, (iii) размещение системных компонентов, (iv) реконфигурация как перепроектирование структуры системы, (v) многостадийное построение и перепроектирование конфигурации системы, (vi) построение (проектирование) или перепроектирование конфигурации системы для случая многопродуктовой системы (т.е., системы, включающей много частей-продуктов с общими компонентами), (vii) построение иерархий систем. Описаны задачи комбинаторной оптимизации (включая многокритериальные постановки): задача о блочном рюкзаке, задача о размещении, задача раскраски графа, задача морфологической клики (с совместимостью системных компонентов), задача многодольной клики и модификации, задачи покрывающих деревьев.

## 1. ВВЕДЕНИЕ

В последние годы важность задач построения системных конфигураций для многокомпонентных систем резко возросла во многих прикладных областях: (1) производственные и робототехнические системы ([31], [261], [282], [377]); (2) устройства, семейства устройств, платформы ([26], [54], [55], [84], [166], [193], [315], [321], [330]); (3) организации ([4], [16], [129], [264], [267], [351]); (4) электроника, вычислительные системы ([42], [259], [298], [350]); (5) программирование ([10], [37], [121], [163], [204], [205], [214], [221], [222], [245], [260], [263], [298], [302], [338], [385]) включая динамическую реконфигуацию систем, состоящих их компонентов (т.е., модульных систем) ([25], [34], [75], [136], [150], [164], [258], [368]); (6) алгоритмические системы и стратегии решения задач ([219], [245], [343], [357], [384], [385]); (7) системы моделей (при управлении моделями, композиции моделей) ([8], [24], [32], [67], [68], [137], [143], [168], [174], [206], [230], [277], [278], [291], [297], [329]); (8) информационные системы (информационные структуры, информация, системы знаний, процессы и др.) ([38], [80], [191], [214], [216]); (9) методы развития информационных систем (технология методов - method engineering) ([1], [49]); (10) коммуникации и телекоммуникационные системы ([5], [44], [237], [238], [239], [240], [260]); (11)

системы безопасности [241]; (12) системы тестирования [236]; (13) сервисы на основе Интернета ([27], [34], [35], [228], [252], [337], [343], [378], [379]); (14) маркетинговые каналы [280]; (15) машинное обучение (machine learning) ([268], [383]); (16) биоинформатика ([248], [326], [347]); (17) OLAP-системы (аналитическая обработка данных в многомерностных информационных системах) ([341], [366]); (18) распознавание сложных (составных) образов ([232], [303]); (19) строительство и архитектура ([138], [222], [234], [301]); (20) транспортные сети ([101] и др.); (21) технология материалов ([62], [370], [371]) и комбинаторная химия ([119], [362]) включая проектирование лекарств ([175], [207], [209]); (22) индивидуальное обучение на основе композиции частей учебных курсов ([186], [219], [223], [226]); (23) модульное планирование [222]; (24) системы распределения ресурсов (например, электрические сети, сети распределения тепла) [140]; (25) системы логистики ([39], [70], [195], [312], [332]). В области системного инжиниринга вопросы системной конфигурации направлены, в основном, на следующее: (i) системы сборки и композиции, (ii) управление конфигурированием/реконфигурированием, (iii) эволюция сложных систем, (iv) архитектуры систем, (v) технология требований к сложным системам.

В основном, следующие типы объектов могут рассматриваться в области построения конфигураций систем (или реконфигурирования): (а) исходные данные (или исходные материалы); (б) системы обработки (т.е., машины, компьютеры, алгоритмы, специалисты, сервисы); (в) результирующие решения (например, составные действия, планы, стратегии, продукты); (г) прикладные системы поддержки (например, сетевые системы, логистика, архитектурная среда, сервисы); (д) системы требований; (е) стандарты. С другой стороны, современные подходы к обработке и анализу данных (например, knowledge discovery and data mining) могут быть часто рассмотрены как конфигурирование информации ([38], [191]). Аналогично, процессы обучения и мышления можно также рассматривать как выявление и конфигурирование структур информации ([191] и др.). С неформальной точки зрения конфигурация системы может быть определена как объединение искусственных объектов (некая спроектированная система) из фиксированного набора заранее определенных компонентов, которые могут быть связаны в рамках предварительно определенных возможностей ([14], [180], [269], [270], [314], [335], [360], [387]). Рис. 1 иллюстрирует задачу построения конфигурации системы.

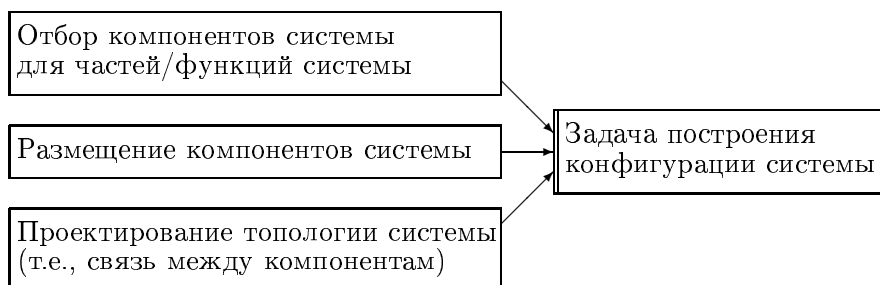


Рис. 1. Иллюстрация задачи построения конфигурации системы

Таким образом можно выделить два типа задач построения конфигураций систем:

**I.** Отбор и/или размещение (вход: множество элементов/компонентов, которое разделено на подмножества): 1.1. отбор представителей (из подмножеств); 1.2. размещение элементов по заданным позициям; 1.3. отбор представителей и их размещение по позициям.

**II.** Построение иерархии системы (т.е., структуры на множестве элементов в виде дерева, леса, иерархии; дополнительные вершины-элементы могут также быть использованы; вход: множество элементов/компонентов системы, дополнительные элементы, свойства требуемой структуры).

Иллюстрация для первой упрощенной задачи (т.е., задача 1.1) представлена на Рис. 2. Задача 1.3 изображена на Рис. 3.

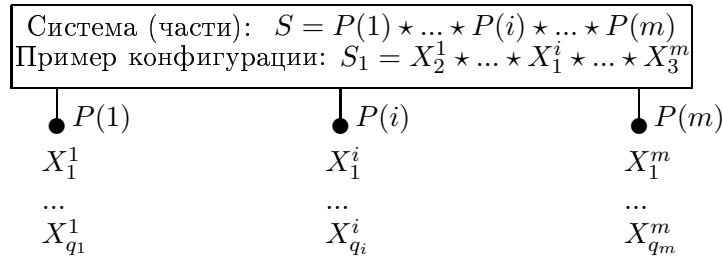


Рис. 2. Построение конфигурации системы (отбор)

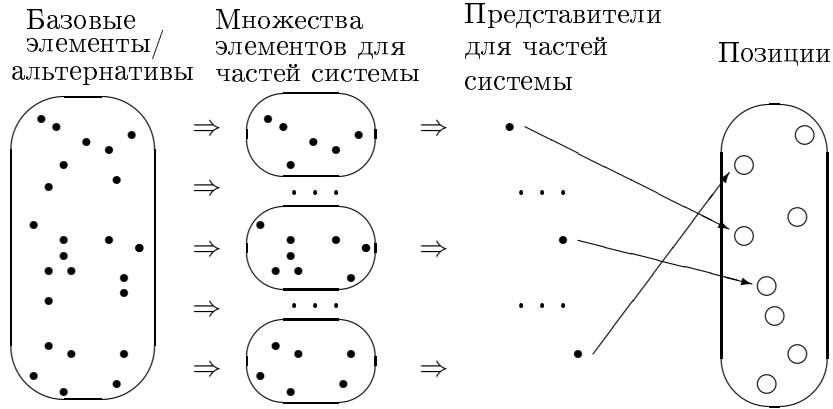


Рис. 3. Иллюстрация отбора&размещения

Таблица 1. Основные задачи построения конфигураций систем, подходы к решению

Основные задачи	Комбинаторные схемы решения
1.Отбор компонентов	1.Задача о представителях [155] 2.Задача о блочном рюкзаке ([133], [190], [214], [237], [256], [379]) 3.Задача кратчайшего пути ([10], [133], [379]) 4.Кластеризация в многодольном графе ([71], [251]), многодольная клика ([94], [247]) 5.Морфологический анализ ([180], [308], [387]) 6.Модифицированный морфологический анализ ([14], [105], [128]) 7.Иерархическое морфологическое проектирование (на основе морфологической клики) ([219], [222] и др.)
2.Размещение компонентов системы	1.Задача о назначении ([133]), многокритериальная задача о назначении ([91], [319]) 2.Квадратичная задача о назначении (QAP) ([63], [289]) 3.Обобщенная задача о назначении, обобщенная квадратичная задача о назначении ([133], [286], [313]) 4.Иерархическое морфологическое проектирование (на основе морфологической клики) ([219], [222] и др.)
3.Построение иерархии системы	1.Иерархическая кластеризация ([171] и др.) 1.1.Агломеративный алгоритм ([171], [173]) 1.2.Иерархический метод с пересечением кластеров [225] 2.Построение 'оптимальной' иерархии ([145], [267], [351])

В основном, следующие подходы (модели) используются при построении конфигураций систем: (1) задача кратчайшего пути ([10], [379]); (2) версии задачи о блочном рюкзаке [214] и многокритериальная задача о блочном рюкзаке ([237], [298], [378], [379]); (3) методы многокритериального принятия решений [44]; (4) традиционные методы морфологического анализа и его модификации ([14], [26], [105], [128], [180], [301], [308], [387]); (5) кластеризация многодольного

графа ([71], [94], [247], [251]); (6) метод Иерархического Морфологического Многокритериального Проектирования (ИММП) на основе задачи морфологической клики ([219], [221], [222]); (7) параметрическое проектирование ([342], [343]); (8) эвристики для задачи идентификации множества компонентов ([37], [38]); (9) эволюционные подходы (генетические алгоритмы и т.п.) ([101], [310] и др.); (10) многоагентные подходы ([55], [56]); (11) подходы на основе размытых множеств ([332] и др.); (12) методы на основе ограничений ([46], [51], [122], [254], [310], [336]), включая составную задачу о выполнимости (composite constraint satisfaction problem) ([314], [335]); (13) подходы, основанные на онтологиях ([39], [80], [82]); (14) методы искусственного интеллекта ([23], [33], [135], [174], [206], [259], [329], [335], [338], [360]); (15) подходы на основе проектных грамматик (например, междисциплинарные грамматики, включающие производственные правила и оптимизацию) ([7], [51], [138], [322], [323], [324]). В тоже время, задачи назначения/размещения ([5], [93], [133], [289]) и задачи построения 'оптимальных' иерархий ([145], [267], [351]) также могут рассматриваться как подходы к построению конфигураций систем. Дополнительно представляется целесообразным указать близкое направление *управление конфигурациями*, которое применяется для программных комплексов [83] и компьютерных сетей ([89], [90]). При этом используются специальные методы, например: парадигма версий (versioning paradigms) [83], модели машин состояний (автоматы) [90], семантические системные изменения (semantic system changes) [309].

В данной статье описывается набор статических прикладных задач построения конфигураций систем, указанных выше (Таблица 1). Для этих задач используются ряд моделей комбинаторной оптимизации (поддерживающие модели): задача кратчайшего пути, задача о блочном рюкзаке, клика в многодольном графе или морфологическая клика, задача размещения, построение покрывающих структур и др. (Таблица 1, Рис. 4). Отметим, для автора задача морфологической клики занимает центральное место. В работе впервые предлагается интегрированная точка зрения применительно к задачам построения конфигураций систем на основе комбинаторной оптимизации. Предварительный материал для данной статьи был представлен на международной конференции о сложных системах ICCS-2007 [227].

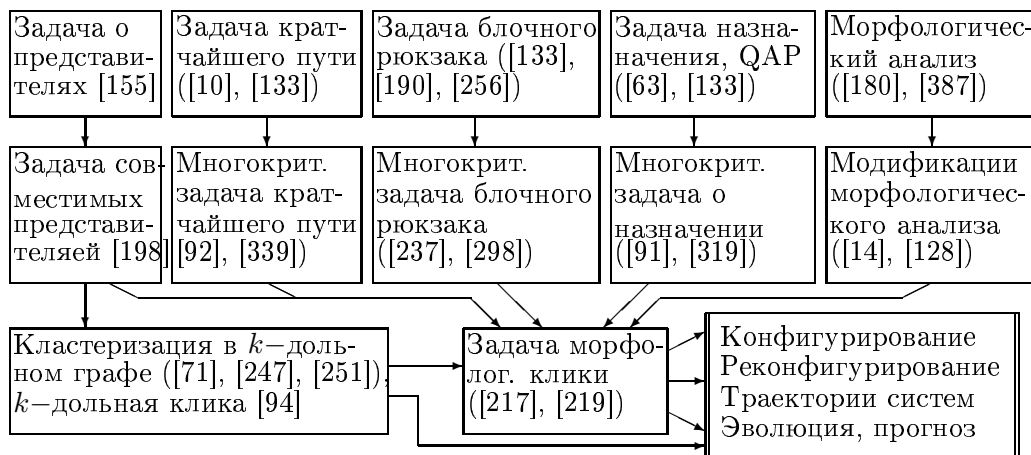


Рис. 4. Базовые задачи комбинаторной оптимизации в построении конфигураций систем (отбор, композиция, назначение/размещение)

## 2. БАЗОВЫЕ КОМБИНАТОРНЫЕ МОДЕЛИ И СХЕМЫ

### 2.1. Задача о представителях

Задача о представителях описана в комбинаторике [155]. Пусть имеется множество элементов  $A$  и подмножества этого множества:  $A = \bigcup_{i=1}^m A_i$ ,  $A_i = \{a_{ij}, j = \overline{1, q_i}\}$ . Задача имеет вид (Рис. 5):

Построить множество из представителей каждого исходного подмножества  $A_i$ .

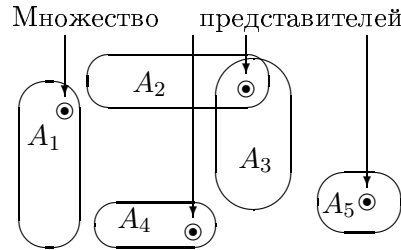


Рис. 5. Задача представителей

Очевидно, что можно рассматривать оптимизационные и многокритериальные постановки задачи о представителях (минимизация числа представителей и др.). В [198] задача о представителях исследуется с учетом бинарных отношений совместимости на элементах (в общем случае эта задача является NP-трудной).

Пусть множество представителей имеет вид:  $X = \{x_\theta\} \subseteq A$ ,  $\theta = \overline{1, k}$ ,  $|X| = k \leq m$ . В случае базовой оптимизационной постановки каждый элемент  $a_{ij}$  имеет некоторую полезность (неотрицательное число)  $c_{ij} = c(a_{ij})$  и процесс отбора представителей может основываться на следующей аддитивной целевой функции (максимизировать или минимизировать):  $f(X) = \sum_{x_\theta \in X} c(x_\theta)$ . Отметим интеграция индивидуальных полезностей может основываться на подобной аддитивной функции. В многокритериальном случае каждый элемент  $a_{ij}$  имеет полезность в виде вектора размерности  $r$ :  $\overline{c}_{ij} = (c_{ij}^1, \dots, c_{ij}^\xi, \dots, c_{ij}^r)$  и отбор представителей нацелен на поиск Парето-эффективных решений, получаемых на основе векторной целевой функции:  $\overline{f}(X) = (\sum_{x_\theta \in X} c^1(x_\theta), \dots, \sum_{x_\theta \in X} c^\xi(x_\theta), \dots, \sum_{x_\theta \in X} c^r(x_\theta))$ .

Отметим, что множество представителей называется *трансверсаль*. С общей точки зрения можно рассматривать задачи о покрытиях и задачи поиска трансверсалей, которые возникают в таких областях, как логика, искусственный интеллект, проектирование систем на базе Интернета ([112], [186]). Трансверсаль, состоящая из минимального числа элементов, называется минимальной ([111], [112], [317]). Разработано много алгоритмов для задачи вычисления минимальной трансверсали (и для некоторых более общих задач) ([108], [111], [161], [176], [189], [192]).

## 2.2. Задача кратчайшего пути

Задача кратчайшего пути была рассмотрена применительно к построению конфигурации систем в [10]. Близкая задача в виде версии морфологической клики с упрощенной структурой совместимости (только для системных компонентов, соответствующим различным 'соседним' частям системы) описана в ([215], [219]). Базовая задача кратчайшего пути (*точка-точка*) имеет следующий вид. Дан направленный связный взвешенный граф  $G = (A, E)$  ( $A$  - множество вершин/узлов,  $E$  - множество дуг), имеется неотрицательный вес  $w(e) \forall e \in E$ , две вершины выделены как исходная  $a \in A$  и целевая  $b \in A$ . Задача формулируется так:

*Найти кратчайший путь из  $a$  в  $b$ , где общая длина пути  $\langle a, b \rangle$  равна сумме весов дуг, входящих в путь.*

Стандартный алгоритм для указанной задачи был разработан Дийкстрой ([100], [133]), вычислительная сложность алгоритма равна  $O(|E| + |A| \log |A|)$  [124]. Формулировка многокритериальной модели для задачи кратчайшего пути была предложена Винске [348], в данной модели несколько параметров связаны с каждой дугой. При этом проводится поиск Парето-эффективных (недоминируемых) путей. Алгоритм для этой задачи был предложен Мартиномсом [257]. Обзор различных многокритериальных постановок задачи кратчайшего пути содержится в ([92], [339]). Многокритериальная задача о кратчайшем пути является NP-трудной

(даже в случае двух критериев) ([133] и др.). В основном следующие алгоритмы используются для многокритериальной задаче о кратчайшем пути: (i) многокритериальный алгоритм Дейкстры (multicriteria Dijkstra's algorithm - label setting algorithm), (ii) многокритериальный алгоритм Форда-Беллмана (multicriteria Ford-Bellman's algorithm), (iii) динамическое программирование, (iv) приближенные алгоритмы, (v) генетические алгоритмы, (vi) эвристики.

В случае построения конфигураций систем задача кратчайшего пути обычно рассматривается на многодольном графе ([10], [215], [236]). При этом каждая доля соответствует части системы, части системы (т.е., доли графа) упорядочены линейно, ребра/дуги связывают вершины (т.е., альтернативы) различных 'соседних' частей графа, результирующий путь должен включать представителей из каждой части графа (Рис. 6).

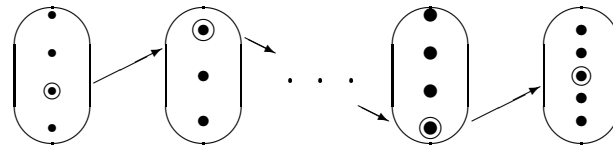


Рис. 6. Кратчайший путь в многодольном графе

В рассмотренном типе задачи кратчайшего пути веса ребер/дуг часто являются векторами или более сложными характеристиками [236]). Такой подход является полезным в случае проектирования последовательных (или параллельно-последовательных) систем (например, стратегии решения, программные комплексы) ([10], [215], [219], [236]).

### 2.3. Задача о рюкзаке и задача о блочном рюкзаке

Задача о рюкзаке широко используется для построения конфигураций систем. Базовая постановка имеет вид ([133], [190], [256]):  $\max \sum_{i=1}^m c_i x_i \quad s.t. \quad \sum_{i=1}^m a_i x_i \leq b, \quad x_i = 0 \cup 1, \quad i = \overline{1, m}$ ; где  $x_i = 1$  если отобран элемент  $i$ ,  $c_i$  - важность ("полезность") элемента  $i$ ,  $a_i$  - вес (или требуемый ресурс). Часто коэффициенты предполагаются неотрицательными. Эта задача является NP-трудной ([133], [256]) и может решаться на основе следующих методов: (i) переборные методы (например, метод ветвей и границ, динамическое программирование), (ii) приближенные полиномиальные схемы с ограниченной ошибкой по целевой функции ([190], [256]), (iii) эвристики. Для многокритериального варианта постановки необходимо проводить поиск Парето-эффективных решений. При этом аналогичные методы могут быть использованы. В задаче о блочном рюкзаке элементы разделены на группы и требуется отбирать элементы в каждой группе с учетом общего ограничения по ресурсу (или ограничений):

$$\max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \quad \sum_{i=1}^m \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b, \quad \sum_{j=1}^{q_i} x_{ij} = 1, \quad i = \overline{1, m}, \quad x_{ij} = 0 \cup 1.$$

В случае многокритериального описания  $\{c_{i,j}\} \quad \forall(i,j)$  (т.е., многокритериальной задачи о блочном рюкзаке), используется следующая векторная целевая функция  $( f^1, \dots, f^\xi, \dots, f^r )$  [237]:

$$( \max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij}^1 x_{ij}, \dots, \max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij}^\xi x_{ij}, \dots, \max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij}^r x_{ij} ).$$

Очевидно, что при этом необходимо проводить поиск Парето-эффективных (по указанной векторной функции) решений. В этом случае могут использоваться следующие схемы решения ([237] и др.): (i) переборные методы (например, метод ветвей и границ, динамическое программирование), (ii) эвристика на основе многокритериального ранжирования элементов и последующее пошаговое заполнение рюкзака, (iii) многокритериальное ранжирование элементов

для получения порядковых оценок их качества с последующим использованием приближенной схемы (как для обычного рюкзака) на основе дискретного пространства качества решения (решетка, описана далее).

#### 2.4. Морфологическое проектирование

##### Морфологический анализ и его модификации

Во-первых, следует указать основные версии морфологического анализа: (1) базовая версия морфологического анализа, предложенная Цвикки ([26], [43], [180], [387]); (2) модификация морфологического анализа в виде поиска допустимых комбинаций из совместимых компонентов (один представитель из каждого морфологического класса, т.е., множества альтернатив для части/компонента системы) с учетом наибольшей 'близости' искомой комбинации к некой идеальной комбинации из лучших для каждого морфологического класса элементов ([14], [105]); (3) модификация морфологического анализа следующего вида: (а) поиск всех допустимых (по совместимости) комбинаций элементов (один представитель из каждого морфологического класса), (б) оценивание найденных комбинаций элементов по набору критериев и выбор Парето-эффективных решений ([113], [128]); (4) интеграция морфологического анализа и кластеризации [304].

##### Иерархическое Морфологическое Многокритериальное Проектирование

Иерархическое Морфологическое Многокритериальное Проектирование (ИММП), основанное на задаче морфологической клики, обобщает задачу о блочном рюкзаке (посредством учета совместимости элементов) и морфологический анализ (посредством порядкового оценивания элементов и их совместимости). С общей точки зрения, ИММП реализует движение от традиционной парадигмы принятия решений Г. Саймона (многокритериальный выбор/отбор альтернатив) [331] к новой парадигме принятия решений в виде комбинаторного синтеза составных альтернатив ([219], [222]). Первое описание ИММП было опубликовано в [217], предварительная версия с порядковыми элементами (т.е., альтернативами) морфологических классов была представлена в [215], предварительная версия иерархического морфологического проектирования модульных систем (на основе ряда задач о блочном рюкзаке) была опубликована в [214].

Краткое типовое описание ИММП имеет следующий вид ([219], [221], [222], [224]). Рассматриваемая система (модульная, декомпозируемая) состоит из компонентов (частей) и их взаимосвязи (IC) или совместимости. При этом делаются предположения: (а) система имеет древовидную структуру (в общем случае это *морфологическое дерево* [224]); (б) составная оценка качества системы интегрирует качество компонентов (подсистем, частей) и качество IC (совместимости) между подсистемами; (в) критерии оценки системы и ее компонентов являются монотонными; (г) качество компонентов и IC оценивается на основе скоординированных порядковых шкал. Используются следующие обозначения: (1) DAs - проектные альтернативы для вершин древовидной модели системы; (2)  $r = \overline{1, k}$  - приоритеты DAs, 1 соответствует наилучшему качеству; (3) порядковая оценка совместимости (IC) для пары DAs  $w = \overline{0, l}$  ( $l$  соответствует наилучшему качеству). ИММП включает четыре основные фазы:

*Фаза 1.* Построение древовидной модели системы,

*Фаза 2.* Генерация проектных альтернатив DAs для всяких вершин модели системы.

*Фаза 3.* Иерархический отбор (т.е., многокритериальное ранжирование для получения приоритетов) и композиция DAs в составные DAs соответствующего более высокого иерархического уровня модели системы (задача морфологической клики).

*Фаза 4.* Анализ и улучшение результирующих с составных решений (т.е., DAs для системы в целом).

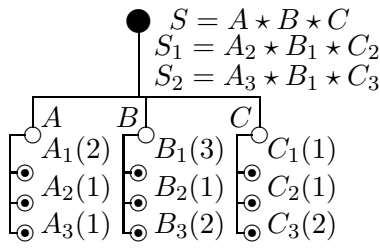


Рис. 7. Пример композиции

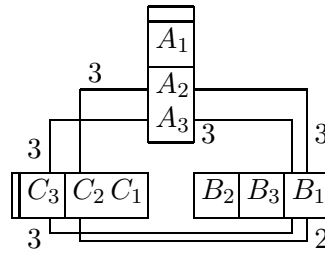


Рис. 8. Концентрическое представление

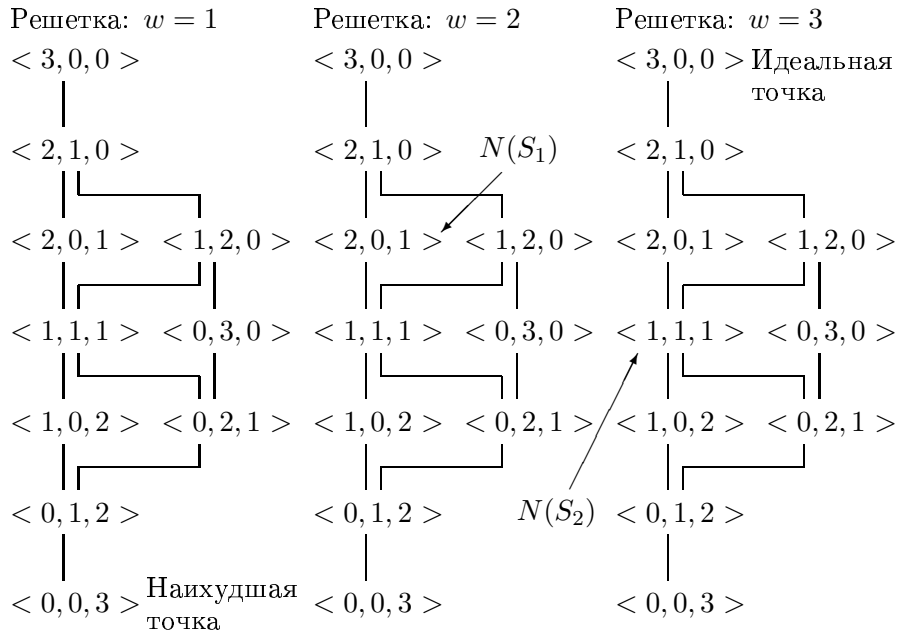


Рис. 9. Дискретное пространство качество системы

Базовая версия задачи морфологической клики имеет следующий вид. Рассмотрим  $S$  как систему, состоящую из  $m$  частей (компонентов):  $P(1), \dots, P(i), \dots, P(m)$ . Множество проектных альтернатив генерируется для каждой указанной части системы. Задача имеет вид:

Найти составную проектную альтернативу  $S = S(1) * \dots * S(i) * \dots * S(m)$  из DAs (один представитель в виде проектной альтернативы  $S(i)$  для каждой части системы/компонента  $P(i), i = \overline{1, m}$ ) с ненулевыми совместимостью IC между проектными альтернативами.

Дискретное пространство качества (совершенства) системы рассматривается на основе следующего вектора:  $N(S) = (w(S); n(S))$ , где  $w(S)$  - минимальное значение парной совместимости между отобранными проектными альтернативами DAs, которые соответствуют различным системным частям (т.е.,  $\forall P(j_1)$  и  $P(j_2), 1 \leq j_1 \neq j_2 \leq m$ ) в  $S$ ,  $n(S) = (n_1, \dots, n_r, \dots, n_k)$ , где  $n_r$  - число DAs  $r$ -го уровня качества в  $S$ . В результате ищутся составные решения, которые недоминируются по  $N(S)$ . Приведенная задача является NP-трудной. Ясно, что компонент совместимости в векторе  $N(S)$  может быть рассмотрен также на основе частично-упорядоченного множества (т.е., решетки) (как и  $n(S)$ ) [222]. В этом случае дискретное пространство качества системы будет также представлять собой аналогичную решетку. Рис. 7, 8 и 9 иллюстрируют задачу композиции. В примере получаются следующие составные решения:  $S_1 = A_2 * B_1 * C_2, N(S_1) = (2; 2, 0, 1); S_2 = A_3 * B_1 * C_3, N(S_2) = (3; 1, 1, 1)$ . Процесс решения



рассматриваемой задачи может базироваться на следующих стратегиях [219]: (1) переборный метод, (2) динамическое программирование.

### 2.5. Кластеризация многодольного графа и взвешенная клика в многодольном графе

Использование задачи морфологической клики (т.е., ИММП) для поиска и синтеза информации по многим источникам описано в ряде публикаций автора ([218], [219]). С другой стороны близкие исследования в указанной прикладной области (т.е., поиск информации по многим источникам) проводятся в последние годы в виде кластеризации в многодольном графе ([71], [251]), например, материал фирмы Yahoo: "clustering search engine" [251].

Описание такого подхода имеет следующий вид [251]. Пусть  $G = (A, E)$  будет многодольным взвешенным графом, где  $A$  - множество вершин (документов),  $E$  - множество ребер. Исследуемый граф рассматривается как объединение  $m$  попарно различных множеств вершин:  $A = \bigcup_{i_1=1}^m A_{i_1}$ ,  $|A_{i_1} \cap A_{i_2}| = 0 \quad \forall i_1 \neq i_2, \quad i_1, i_2 = \overline{1, m}$ . Вес ребра  $(v, u) \in E$  ( $v \in A_{i_1}, u \in A_{i_2}, i_1 \neq i_2$ ) соответствует уровню схожести между двумя документами (т.е., вершинами). Здесь веса всех ребер, соединяющих вершины внутри одной и той же доли (т.е.,  $i_1 = i_2$ ), равны нулю. Концептуальное описание задачи имеет вид ([251], [346]):

*Найти множество представителей (один или несколько) из каждой доли  $(A_{i_1}, i_1 = \overline{1, m})$  так, что общая схожесть (как некая аддитивная функция попарной схожести) в отобранном множестве представителей максимизируется.*

Точная формулировка задачи представлена в [251] и соответствующий полиномиальный алгоритм предложен в [346].

Одновременно также рассматривается задача максимальной клики в многодольном графе (т.е., многодольная клика) ([94], [293]). В [94] описаны три версии для задач с взвешенными ребрами и узлами (вершинами). При этом показано, что все задачи с взвешенными ребрами являются NP-сложными. Следует отметить, что задача кластеризации многодольного графа и задача клики в многодольном графе очень близки и рассматриваются со следующих точек зрения: (а) проектирование алгоритмов ([71], [94], [116], [142], [247], [251], [293], [347]); (б) приложения в области автоматически обучающихся систем (machine learning, supervised learning problems) ([268], [383]); (в) приложения в интеграции информации [251]; (г) приложения в биоинформатике ([248], [326], [347]).

### 2.6. Задачи о назначении/размещении

#### Базовая задача о назначении

Задачи о назначении/размещении широко применяются во многих областях ([63], [93], [133], [286], [289], [313]). Простейшая задача о назначении включает неотрицательную матрицу соответствия  $\Upsilon = ||c_{ij}||$  ( $i = \overline{1, n}, j = \overline{1, n}$ ) где  $c_{ij}$  - польза ('полезность', 'выгода') назначения элемента  $i$  на позицию  $j$ . Задача имеет вид [133]:

*Найти назначение  $\pi = (\pi(1), \dots, \pi(n))$  элементов  $i$  ( $i = \overline{1, n}$ ) на позиции  $\pi(i)$ , которое соответствует общей эффективности  $\sum_{i=1}^n c_{i\pi(i)} \rightarrow \max$ .*

Рис. 10 иллюстрирует задачу о назначении/размещении. Очевидно, что позиции могут рассматриваться в пространстве/области размерности  $m$ .

Алгебраическая постановка задачи имеет вид:

$$\max \sum_{i=1}^m \sum_{j=1}^n c_{i,j} x_{i,j} \quad \text{s.t.} \quad \sum_{i=1}^m x_{i,j} \leq 1, j = \overline{1, n}; \quad \sum_{j=1}^n x_{i,j} \leq 1, i = \overline{1, m}; \quad x_{i,j} = 0 \cup 1, i = \overline{1, m}, j = \overline{1, n}.$$

Здесь  $x_{i,j} = 1$ , если элемент  $i$  назначен на позицию  $j$ ,  $c_{i,j}$  - важность ('полезность') указанного назначения. Эта задача решается полиномиально, например, венгерским методом ([208] и

др.). Следует отметить, что указанная задача соответствует задаче сопоставления (matching) на двух-дольном графе (bipartite graph) ([133] и др.).

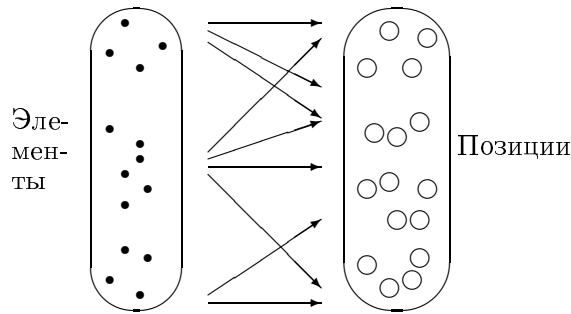


Рис. 10. Задача о назначении/размещении

Более сложной известной моделью является задача квадратичного назначения *quadratic assignment problem (QAP)*, которая включает учет взаимосвязей между различными группами (каждая группа соответствует определенной позиции) ([63], [219], [289]). Пусть неотрицательное значение  $d(i, j_1, k, j_2)$  будет соответствовать совместимости между элементом  $j_1$  в группе  $J_i$  и элементом  $j_2$  в группе  $J_k$ . Указанное значение прибавляется к целевой функции. Постановка задачи *QAP* (NP-трудная) имеет вид:

$$\begin{aligned} \max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{i,j} x_{i,j} + \sum_{l < k} \sum_{j_1=1}^{q_l} \sum_{j_2=1}^{q_k} d(l, j_1, k, j_2) x_{l,j_1} x_{k,j_2}, \quad l = \overline{1, m}, \quad k = \overline{1, m}; \\ \text{s.t.} \quad \sum_{i=1}^m \sum_{j=1}^{q_i} a_{i,j} x_{i,j} \leq b, \quad \sum_{j=1}^{q_i} x_{i,j} \leq 1, \quad j = \overline{1, m}; \quad x_{i,j} = 0 \cup 1, \quad i = \overline{1, m}, \quad j = \overline{1, q_i}. \end{aligned}$$

Для этой задачи используются следующие схемы решения: (1) методы ветвей и границ, (2) эвристики, например, генетические алгоритмы ([340] и др.), поиск с запретами (tabu search) ([334] и др.).

В рамках данного исследования представляется важным использовать *многокритериальную задачу о назначении*:  $c_{i,j} \Rightarrow \overline{c_{i,j}} = (c_{i,j}^1, \dots, c_{i,j}^\xi, \dots, c_{i,j}^r)$  ([91], [212], [219], [319]). Таким образом векторная целевая функция (например, в простейшем случае) имеет вид:

$$(\sum_{i=1}^m \sum_{j=1}^{q_i} c_{i,j}^1 x_{i,j}, \dots, \sum_{i=1}^m \sum_{j=1}^{q_i} c_{i,j}^\xi x_{i,j}, \dots, \sum_{i=1}^m \sum_{j=1}^{q_i} c_{i,j}^r x_{i,j}).$$

Ясно, что здесь надо искать Парето-эффективные решения. При этом могут использоваться несколько схем решения, например:

- (1) переборные методы;
- (2) интерактивные процедуры ([212] и др.);
- (3) различные эвристики, включая: (а) двух-стадийная эвристика: (i) многокритериальное ранжирование полезности назначений (т.е., векторов  $\{\overline{c_{i,j}}\}$ ), (ii) эвристика для полученной упрощенной задачи о назначении; (б) двух-стадийная эвристика: (i) генерация множества решений как аппроксимация множества Парето-эффективных точек (например, на основе решения нескольких задач о назначении с одной целевой функцией), (ii) выявление наилучшего окончательного решения или решений.

Аналогично можно рассмотреть трансформацию (чтобы перейти к многокритериальным постановкам) для квадратичной задачи о назначении и для обобщенной задачи о назначении.

### Расширенная задача размещения

В [219], предложен более общий случай задачи размещения со следующими расширениями: (а) дополнительные правила по размещению элементов по позициям (т.е., дополнительная

часть на основе искусственного интеллекта), (б) дополнительные бинарные отношения на множествах элементов и на множествах позиций. Подобная расширенная задача может решаться на основе ИММП [219]. Данный подход иллюстрируется примером, представляющим собой модификацию примера из [219]. Рассматриваются следующие концепты: (1) исследовательские проекты; (2) персонал (т.е., сотрудники); (3) помещения; (4) отношения на указанных концептах. Имеется два исследовательских проекта: (а) большой проект  $R_1$  (лидер, менеджер, три специалиста, секретарь); (б) проект  $R_2$  (лидер, два специалиста). На Рис. 11, Рис. 12 и в Таблице 2 приведены: (а) структуры проектов; (б) план помещений ( $A, B, C, X, Y$ ) и их близость (два уровня близости: пунктирная линия соответствует слабой близости); (в) описание сотрудников. Таблица 2 содержит также допустимые размещения сотрудников по помещениям (указано в квадратных скобках [...]).

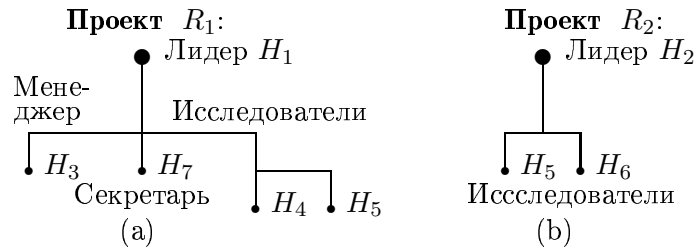


Рис. 11. Структура проектов

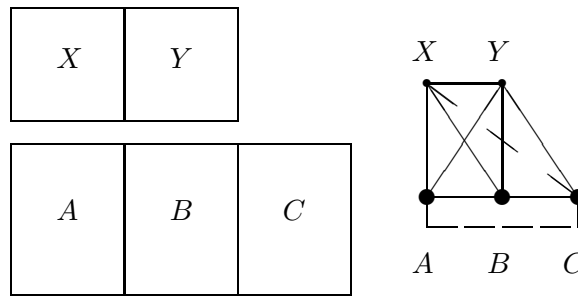


Рис. 12. План помещений, граф их 'близости'

Дополнительно рассматриваются следующие правила:

**Правило 1.** Лидер проекта должен быть размещен близко к сотрудникам, участвующим в его проекте (т.е., исследователи, менеджер, секретарь).

**Правило 2.** Лидер проекта должен быть размещен один в большом или малом помещении.

**Правило 3.** Менеджер проекта должен быть размещен близко к лидеру группы и секретарю.

**Правило 4.** Менеджер проекта может быть размещен в малом помещении (один) или в большом помещении (вдвоем).

**Правило 5.** Сотрудники, участвующие в одном проекте, должны быть размещены в одном и том же помещении или в помещениях, расположенных близко друг к другу.

**Правило 6.** Секретарь может быть размещен в большом помещении (вдвоем) или в малом помещении (вдвоем).

**Правило 7.** Курящие и некурящие сотрудники не могут быть размещены в одном помещении.

**Правило 8.** Друзья должны быть размещены в одном и том же помещении или в близко расположенных помещениях.

Указанные правила были использованы при определении возможного размещения сотрудников в помещениях (Таблица 2).

Таблица 2. Описание персонала

Но.	Роль	Проект	Курение	Помещения	Дружба
$H_1$	Лидер большого проекта	$R_1$	Да	$A[1], B[1], C[1]$	$H_2, H_3$
$H_2$	Лидер проекта	$R_2$	Нет	$A[1], B[1], C[1]$	$H_1$
$H_3$	Менеджер большого проекта	$R_1$	Да	$A[2], B[2], C[2], X[1], Y[1]$	$H_1, H_7$
$H_4$	Исследователь	$R_1$	Нет	$A[2], B[2], C[2], X[1], Y[1]$	
$H_5$	Исследователь	$R_1, R_2$	Да	$A[2], B[2], C[2]$	$H_7$
$H_6$	Исследователь	$R_2$	Да	$A[2], B[2], C[2]$	$H_5$
$H_7$	Секретарь	$R_1$	Да	$A[2], B[2], C[2], X[2], Y[2]$	$H_3, H_5$

Таблица 3. Дополнительность

	$H_4$	$H_5$	$H_6$	$H_7$
$H_3$	(010)/0	(110)/1	(100)/0	(111)/2
$H_4$		(010)/0	(000)/0	(010)/0
$H_5$			(111)/2	(100)/0
$H_6$				

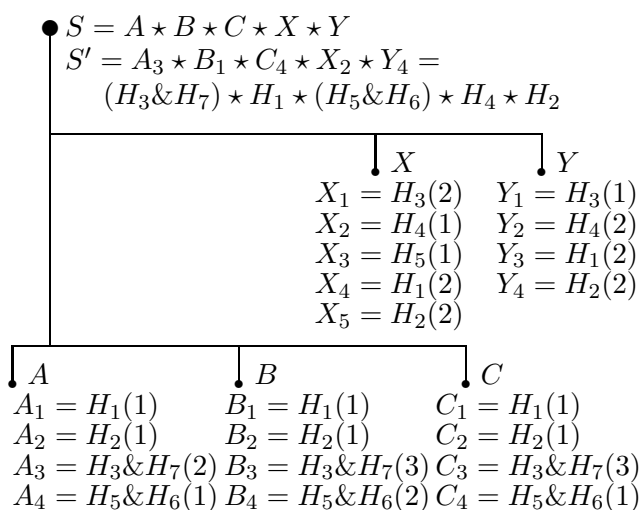


Рис. 13. Структура составных решений

Также рассматривается бинарное отношение 'дополнительности' на множестве сотрудников, т.е., возможности размещаться в одном помещении (с учетом дружбы, курения, участия в одном проекте). Таблица 3 содержит векторные оценки по этому бинарному отношению и результирующую порядковую оценку:

1. Векторная оценка включает следующие бинарные компоненты: (1) курение (1 - оба курят или не курят, 0 - иначе); (2) совместная работа (1 - участие в одном проекте, 0 - иначе); (3) дружба (1 - дружба, 0 - иначе).

2. Для результирующей оценки 'дополнительности' используется следующая порядковая шкала: [0, 1, 2], где 0 - несовместимость, 1 - совместимость, 2 - хорошая совместимость.

В результате получены следующие хорошие агрегированные пары сотрудников как альтернативы (DAs) совместного размещения:  $H_{37} = H_3 \& H_7$ ,  $H_{56} = H_5 \& H_6$ . Отметим, что здесь использована специальная модель: клика с полезностью и с учетом дополнительности отбираемых элементов (подробно рассмотрено в [219]).

Специальные оценки двух типов использованы для описания отношений между персоналом и помещениями: (1) соответствие персонала помещению; (2) совместимость локальных решений. Локальное решение состоит в размещении сотрудника в помещении. В каждом из рассмотренных случаев используются следующее: (а) вербальные описания и правила; (б)

многокритериальное оценивание; (в) результирующая порядковая оценка. Результирующая структура составного решения представлена на Рис. 13.

Таблица 4. Два отношения

	$H_2$	$H_3$	$H_4$	$H_5$	$H_6$	$H_7$
$H_1$	f	p,f	p	p	-	p
$H_2$		-	-	p	p	p
$H_3$			p	p	-	p,f
$H_4$				p	-	p
$H_5$					f	p,f
$H_6$						-

Таблица 5. Совместимость

	$B_1$	$B_2$	$B_3$	$B_4$
$A_1$	$/\Delta$	$(0, 2)/2$	$(4, 2)/4$	$(2, 0)/2$
$A_2$	$(0, 2)/2$	$/\Delta$	$/\diamond$	$(4, 0)/3$
$A_3$	$(4, 2)/4$	$/\diamond$	$/\Delta$	$(4, 4)/5$
$A_4$	$(2, 0)/2$	$(4, 0)/3$	$(4, 4)/5$	$/\Delta$

Таблица 6. Совместимость

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
$A_1$	$(2, 2)/3$	$(2, 0)/2$	$(2, 0)/2$	$/\Delta$	$/\diamond$	$(2, 2)/3$	$(2, 0)/2$	$/\Delta$	$/\diamond$
$A_2$	$(0, 2)/2$	$/\diamond$	$(2, 0)/2$	$/\diamond$	$/\Delta$	$(0, 2)/2$	$/\diamond$	$/\diamond$	$/\Delta$
$A_3$	$/\Delta$	$(4, 0)/3$	$(4, 4)/5$	$(2, 1)/2$	$(0, 0)/0$	$/\Delta$	$(4, 0)/3$	$(2, 1)/2$	$/\diamond$
$A_4$	$(2, 2)/3$	$(2, 0)/2$	$/\Delta$	$(0, 0)/0$	$(2, 0)/2$	$(2, 2)/3$	$(2, 0)/2$	$(0, 0)/0$	$(2, 0)/2$
$B_1$	$(2, 2)/3$	$(2, 0)/2$	$(2, 0)/2$	$/\Delta$	$/\diamond$	$(2, 2)/3$	$(2, 0)/2$	$/\Delta$	$/\diamond$
$B_2$	$(0, 2)/2$	$/\diamond$	$(2, 0)/2$	$/\diamond$	$/\Delta$	$(0, 2)/2$	$/\diamond$	$/\diamond$	$/\Delta$
$B_3$	$/\Delta$	$(4, 0)/3$	$(4, 4)/5$	$(2, 1)/2$	$(0, 0)/0$	$/\Delta$	$(4, 0)/3$	$(2, 1)/2$	$/\diamond$
$B_4$	$(2, 2)/3$	$(2, 0)/2$	$/\Delta$	$(0, 0)/0$	$(2, 0)/2$	$(2, 2)/3$	$(2, 0)/2$	$(0, 0)/0$	$(2, 0)/2$
$C_1$	$(1, 1)/2$	$(1, 0)/1$	$(1, 0)/1$	$/\Delta$	$/\diamond$	$(2, 2)/3$	$(2, 0)/2$	$/\Delta$	$/\diamond$
$C_2$	$(0, 1)/1$	$/\diamond$	$(1, 0)/1$	$/\diamond$	$/\Delta$	$(0, 2)/2$	$/\diamond$	$/\diamond$	$/\Delta$
$C_3$	$/\Delta$	$(2, 0)/2$	$(2, 2)/4$	$(1, 1)/2$	$(0, 0)/0$	$/\Delta$	$(4, 0)/3$	$(2, 1)/2$	$/\diamond$
$C_4$	$(1, 1)/2$	$(2, 0)/2$	$/\Delta$	$(0, 0)/0$	$(1, 0)/1$	$(2, 2)/3$	$(2, 0)/2$	$(0, 0)/0$	$(2, 0)/2$

Таблица 7. Совместимость

	$C_1$	$C_2$	$C_3$	$C_4$
$A_1$	$/\Delta$	$(0, 1)/1$	$(2, 1)/2$	$(1, 0)/1$
$A_2$	$(0, 1)/1$	$/\Delta$	$/\diamond$	$(2, 0)/2$
$A_3$	$(2, 1)/2$	$/\diamond$	$/\Delta$	$(2, 2)/3$
$A_4$	$(2, 0)/2$	$(2, 0)/2$	$(2, 2)/3$	$/\Delta$
$B_1$	$/\Delta$	$(0, 2)/2$	$(4, 2)/4$	$(2, 0)/2$
$B_2$	$(0, 2)/2$	$/\Delta$	$/\diamond$	$(4, 0)/3$
$B_3$	$(4, 2)/4$	$/\diamond$	$/\Delta$	$(4, 4)/5$
$B_4$	$(2, 0)/2$	$(4, 0)/3$	$(4, 4)/5$	$/\Delta$

Таблица 8. Совместимость

	$Y_1$	$Y_2$	$Y_3$	$Y_4$
$X_1$	$/\Delta$	$(2, 0)/2$	$(2, 3)/3$	$(0, 0)/0$
$X_2$	$(2, 0)/2$	$/\Delta$	$(2, 2)/3$	$/\diamond$
$X_3$	$(2, 2)/3$	$(2, 0)/2$	$(2, 2)/3$	$(0, 0)/0$
$X_4$	$(2, 2)/3$	$(1, 2)/3$	$/\Delta$	$(0, 1)/1$
$X_5$	$(0, 0)/0$	$/\diamond$	$(0, 1)/1$	$/\Delta$

Порядковые оценки соответствия сотрудников помещениям основаны на экспертном оценивании (с учетом указанных выше правил и персональных предпочтений). Таблица 4 содержит два бинарных отношения между сотрудниками, которые основаны на данных из Рис. 11 и Таблицы 2: совместный проекты (р) и дружба (f). Таблицы 5, 6, 7, 8 содержат оценки совместимости для локальных решений. При этом использованы следующие оценки:

1. Векторная оценка, состоящая из двух компонентов:

(1) выполнение правил для совместного проекта: (i) оценка равна 2, если выполняются правила 1, 2, 5; (ii) оценка равна 1, если правила 1, 3, 5 выполняются 'слабо'; (iii) оценка обозначается  $\Delta$  в противном случае (т.е., невыполнение правил), это соответствует несовместимости;

(2) выполнение правил о дружбе: (i) оценка равна 2, если выполняется правило 8; (ii) оценка равна 1, если правило 8 выполняется 'слабо'; (iii) оценка равна 0 в противном случае.

2. Результирующая порядковая оценка: (i) наилучшая совместимость: (a) 5 (вектор оценок равен (4, 4) или (8, 0)) или (b)  $\diamond$  (т.е., независимость DAs); (ii) почти наилучшая совместимость 4 (вектор оценок равен (4, 2)); (iii) хорошая совместимость 3 (вектор оценок равен (4, 0) ил (2, 2)); (iv) удовлетворительная совместимость 2 (вектор оценок равен (1, 1), (2, 0) (0, 2), (2, 1)); (v) плохая совместимость 1 (вектор оценок равен (0, 1), (1, 0)); (vi) несовместимость: (a) 0 (вектор оценок равен (0, 0)); (b)  $\triangle$  (если соответствующий компонент вектора оценок обозначен  $\triangle$ ).

На Рис. 14 представлены результирующие составные решений по размещению сотрудников:  
 $S' = A_3 \star B_1 \star C_4 \star X_2 \star Y_4 = (H_3 \& H_7) \star H_1 \star (H_5 \star H_6) \star H_4 \star H_2$ ;  $N(S') = (2; 3, 2, 0)$ .

$X_2 :$ $H_4$	$Y_4 :$ $H_2$	
$A_3 :$ $H_3 \& H_7$	$B_1 :$ $H_1$	$C_4 :$ $H_5 \& H_6$

Рис. 14. Составные решения

### 2.7. Задачи раскраски и перекраски графа

Следует отметить, что многие комбинаторные задачи являются близкими с нескольких точек зрения: формулировки, сложность, алгоритмы/схемы решения, приложения, например: задача клики, задача независимого множества, задача о покрытии, задача трансверсали, задача раскраски графа ([114], [133], [158], [159], [178], [188], [287], [294], [358]). Фактически задачи раскраски графа (раскраска вершин, раскраска ребер) (т.е., назначение цветов) также соответствуют построению конфигураций. Ключевые приложения задач раскраски графа относятся к следующим областям: (i) назначение регистров в компиляторах ([48], [184] и др.), (ii) коммуникационные системы (назначение частот/каналов, проектирование сетевых протоколов, составление расписаний в сенсорных/коммуникационных сетях и т.п.) ([99], [117], [125], [154], [306], [307], [345]), (iii) составление расписаний в образовании, спорте, медицине (timetabling) ([96], [133], [283], [320] и др.). В [292] эта задача используется для проектирования производственных систем и технологических процессов. При этом цвета соответствуют свойствам систем, функциям, версиям.

Задача раскраски вершин графа может быть представлена следующим образом ([40], [47], [50], [114], [133], [158], [159], [255], [294], [345], [358]). Дан неориентированный граф  $G = (A, E)$  ( $A$  - множество узлов/вершин,  $E$  - множество ребер,  $|A| = n$ ). Имеется набор цветов (номеров, меток)  $X = \{x_1, x_2, \dots, x_k\}$ . Будем называть  $C(G) = \{C(a_1), \dots, C(a_i), \dots, C(a_i)\}$  ( $C_{a_i} \in X$ ) (или  $\langle C(a_1) \star \dots \star C(a_i) \star \dots \star C(a_i) \rangle$ ) *конфигурацией цветов* (т.е., назначение цвета для каждой вершины). Постановка имеет вид:

*Назначить для каждой вершины  $\forall a_i \in A$  цвет или метку (т.е.,  $C(a_i)$ ) так, что вершины одного цвета не соединены ребром, т.е.,  $\forall a_i, a_j \in A$  если  $(a_i, a_j) \in E$  (т.е., смежные вершины) тогда  $C(a_i) \neq C(a_j)$ .*

Таким образом, для заданного графа  $G = (A, E)$  осуществляется поиск *конфигурации цветов* (т.е.,  $C(G) = \{C(a_1), \dots, C(a_i), \dots, C(a_n)\}$ ). Ясно, что  $|C(G)|$  равно числу использованных цветов (меток). (Минимальное число требуемых цветов для графа  $G$  называется *хроматическим числом* данного графа  $\chi(G)$ ). Следует отметить, что другие типы задачи раскраски графа могут быть трансформированы в задачу раскраски вершин. Рис. 15 иллюстрирует задачу

раскраски вершин:  $G = (A, E)$ ,  $A = \{p, q, u, v, w\}$ ,  $E = \{(p, q), (q, v), (u, v), (w, p)(w, q)(w, u)(w, v)\}$  и три цвета  $\{x_1, x_2, x_3\}$  (т.е., соответствующие индексы для альтернативных цветов вершин). Здесь решение имеет вид:  $C(G) = \{P_2, Q_3, U_3, V_2, W_1\}$ . При этом число возможных результирующих конфигураций цветов (три цвета) равно 6.

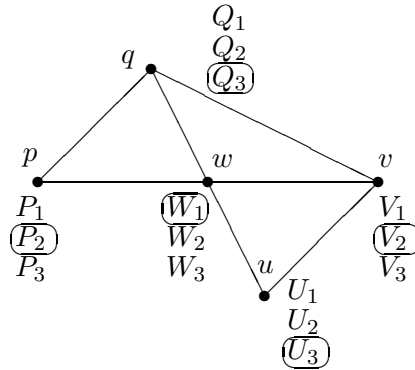


Рис. 15. Пример раскраски вершин

Исследуются различные модификации задачи раскраски вершин, например: (1) рассмотрение специальных структур графа (цепочки, деревья, гиперграфы и др.) ([114], [133], [274], [275], [358]); (2) помеченные графы с условием на расстоянии два ([65], [66], [141], [375], [376]); (3) назначение нескольких цветов для одной и той же вершины; (4) оценивание совместимости между цветами смежных (соседних) вершин [219]; (5) помеченные графы с условием на расстоянии 3, 4 и т.п.; (6) оценивание соответствия каждого цвета по отношению к каждой вершине [219]. Можно рассматривать агрегированный вес (т.е., аддитивная функция) использованных цветов (каждый цвет имеет неотрицательный вес  $w(x_l) \forall x_l \in X, l = \overline{1, k}$ ). В этом случае можно сформулировать задачу минимизации:

Найти раскраску вершин  $C^*(G)$  для заданного графа  $G = (A, E)$  с минимальным числом использованных цветов (меток):

$$\min_{\{C(G)\}} |C^*(G = (A, E))| \quad s.t. \quad C^*(a_i) \neq C^*(a_j) \quad \forall (a_i, a_j) \in E, i \neq j.$$

Эта задача является NP-трудной ([114], [133], [178], [358]). Пусть  $\overline{C^*(G)} = \{c_\theta^*\}$  будет множество использованных цветов (i.e.,  $\overline{C^*(G)} \subseteq C^*(G)$ ). В случае взвешенных цветов (и аддитивной агрегирующей функции) может быть рассмотрена следующая постановка:

$$\min \sum_{\forall c_\theta^* \in \overline{C^*(G)}} w(c_\theta^*) \quad s.t. \quad C^*(a_i) \neq C^*(a_j) \quad \forall (a_i, a_j) \in E, i \neq j.$$

Очевидно, что если  $w(x_l) = 1 \forall x_l \in X$  эта формулировка является эквивалентной предыдущей. При векторном весе цвета  $(w^1(c_\theta), \dots, w^\mu(c_\theta), \dots, w^\lambda(c_\theta)) \quad \forall c_\theta \in C$  и аддитивных агрегирующих функций целевая векторная функция имеет вид:

$$\left( \sum_{\forall c_\theta^* \in \overline{C^*(G)}} w^1(c_\theta^*), \dots, \sum_{\forall c_\theta^* \in \overline{C^*(G)}} w^\mu(c_\theta^*), \dots, \sum_{\forall c_\theta^* \in \overline{C^*(G)}} w^\lambda(c_\theta^*) \right)$$

и осуществляется поиск решений, которые являются Парето-эффективными по указанной векторной функции.

В общем случае, представляется перспективным рассматривать следующие целевые функции (критерии): (i) число использованных цветов, (ii) агрегированный вес использованных

цветов, (iii) соответствие цветов вершинам (например, наихудшее соответствие, среднее соответствие) [219]; (iv) качество совместимости цветов, которые назначены смежным (соседним) вершинам (например, наихудшая совместимость, средняя совместимость) [219]; (v) учет условий на расстоянии 3, 4 и т.д.

Можно указать следующие широко применяемые для задач раскраски графа типы алгоритмов: (1) точные переборные алгоритмы ([58], [114], [183], [249], [316], [328]); (2) случайные алгоритмы (random algorithms) ([114], [327]); (3) приближенные алгоритмы ([40], [41], [187], [274], [281], [288], [361]); (4) эвристики и метаэвристики ([244], [333]), включая следующие: методы локального поиска ([59], [60], [61], [131]), методы отжига (simulated annealing) ([64], [177], [250]), метод поиска с запретами (tabu search) ([102], [162]), поточечный поиск (scatter search) [157], "муравьиная" оптимизация (Ant colony optimization) ([53], [86]), поиск на основе переменной окрестности [11]; (5) генетические алгоритмы/ эволюционная оптимизация ([22], [110], [123], [130], [210], [255]); (6) метод генерации столбцов [262]; (7) алгоритмы на основе нейронных сетей ([99], [125]); (8) решатели на основе обучающихся автоматов [45]; (9) алгоритмы на основе использования агентов [52].

Таблица 9. Совместимость цветов

	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>	V <sub>6</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>	W <sub>5</sub>	W <sub>6</sub>	U <sub>1</sub>	U <sub>2</sub>	U <sub>3</sub>	U <sub>4</sub>	U <sub>5</sub>	U <sub>6</sub>
P <sub>1</sub>	0	1	2	3	4	3	4	4	4	4	4	4	0	1	2	3	4	3	4	4	4	4	4	4
P <sub>2</sub>	1	0	4	2	3	4	4	4	4	4	4	4	1	0	1	2	3	4	4	4	4	4	4	4
P <sub>3</sub>	2	4	0	1	2	4	4	4	4	4	4	4	2	1	0	1	2	4	4	4	4	4	4	4
P <sub>4</sub>	3	2	1	0	3	2	4	4	4	4	4	4	3	2	1	0	3	2	4	4	4	4	4	4
P <sub>5</sub>	4	3	2	3	0	2	4	4	4	4	4	4	4	3	2	3	0	2	4	4	4	4	4	4
P <sub>6</sub>	3	4	4	2	2	0	4	4	4	4	4	4	3	4	4	2	2	0	4	4	4	4	4	4
Q <sub>1</sub>							0	1	2	3	4	3	0	1	2	3	4	3	4	4	4	4	4	4
Q <sub>2</sub>							1	0	4	2	3	4	1	0	1	2	3	4	4	4	4	4	4	4
Q <sub>3</sub>							4	1	0	1	2	4	2	1	0	1	2	4	4	4	4	4	4	4
Q <sub>4</sub>							3	2	1	0	3	2	3	2	1	0	3	2	4	4	4	4	4	4
Q <sub>5</sub>							4	3	2	3	0	2	4	3	2	3	0	2	4	4	4	4	4	4
Q <sub>6</sub>							3	4	4	2	2	0	3	4	4	2	2	0	4	4	4	4	4	4
V <sub>1</sub>													0	1	2	3	4	3	0	1	2	3	4	3
V <sub>2</sub>													1	0	1	2	3	4	1	0	4	2	3	4
V <sub>3</sub>													2	1	0	1	2	4	2	4	0	1	2	4
V <sub>4</sub>													3	2	1	0	3	2	4	3	2	3	0	2
V <sub>5</sub>													4	3	2	3	0	2	4	3	2	3	0	2
V <sub>6</sub>													3	4	4	2	2	0	3	4	4	2	2	0
W <sub>1</sub>																			0	1	2	3	4	3
W <sub>2</sub>																			1	0	1	2	3	4
W <sub>3</sub>																			2	1	0	1	2	4
W <sub>4</sub>																			3	2	1	0	3	2
W <sub>5</sub>																			4	3	2	3	0	2
W <sub>6</sub>																			3	4	4	2	2	0

В [219] автором была предложена версия задачи раскраски вершин графа с учетом совместимости и соответствия цветов вершинам (численный пример, Рис.16). При этом схема решения основана на задаче морфологической клики (т.е., ИММП). В примере использованы 6 цветов:  $x_1, x_2, x_3, x_4, x_5, x_6$ . Оценки соответствия цветов вершинам показаны в круглых скобках на Рис. 16 (оценка 1 является наилучшей). Таблица 9 содержит оценки совместимости для цветов (4 соответствует наилучшему уровню); если ребро между вершинами отсутствует, то соответствующие оценки совместимости цветов равны наилучшему уровню (т.е., 4) для пар вершин  $(p, u), (p, v), (q, u)$ . Результирующая комбинация цветов (композиция цве-



тов) в виде Парето-эффективного решения и соответствующий вектор качества имеют вид:  $C^*(G) = \langle W_6 \star P_2 \star Q_3 \star U_2 \star V_3 \rangle$ ,  $N(C^*(G)) = (4; 4, 0, 0)$ .

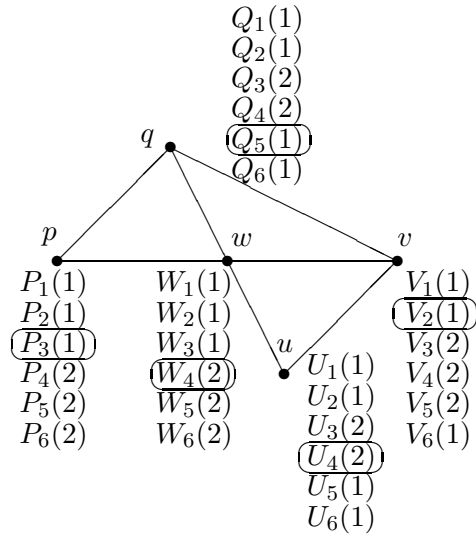


Рис. 16. Раскраска с совместимостью

В последние годы активно исследуется задача перекраски графа ([15], [20], [153], [274], [275]). Здесь базовая формулировка такой задачи имеет следующий вид. Задано следующее: граф  $G = (A, E)$  ( $|A| = n$ ), множество цветов  $X = \{x_1, \dots, x_k\}$ , исходная *конфигурация цветов* для графа  $G$ :  $C^o(G) = \{C^o(a_1), \dots, C^o(a_i), \dots, C^o(a_n)\}$ ,  $C^o(a_i) \in X \ \forall a_i \in A \ (i = \overline{1, n})$ . Пусть задана неотрицательная матрица стоимости перекраски для каждой вершины (индивидуальная матрица перекраски):  $d_{a_i}(x_{\delta_1}, x_{\delta_2}) \ (\delta_1 = \overline{1, k}, \ \delta_2 = \overline{1, k}) \ \forall a_i \in A$ . Для графа  $G$  задается также некоторая целевая *комбинация цветов*  $C^g(G)$  или множества целевых *комбинаций цветов*  $\{C^g(G)\}$ . Теперь для каждых двух *комбинаций цветов* ( $C^1(G)$  и  $C^2(G)$ ) можно рассматривать интегрированную стоимость (стоимость перекраски) как некоторую аддитивную функцию  $D(C^1(G), C^2(G)) = \sum_{i=1}^n d_{a_i}(C^1(a_i), C^2(a_i))$ . Дополнительно необходимо рассмотреть близость двух *конфигураций цветов*:  $\rho(C^{\beta_1}(G), C^{\beta_2}(G))$ . Таким образом рассматривается следующая цепочка преобразований:  $C^o(G = (A, E)) \Rightarrow C^*(G = (A, E)) \Rightarrow C^g(G = (A, E))$ , где  $C^*(G = (A, E))$  является результирующей *конфигурацией цветов*.

В более общем случае можно рассматривать две обобщенные цели: (i) стоимость трансформации  $C^o(G) \Rightarrow C^*(G)$  и (ii) качество близости между  $C^*(G)$  и  $C^g(G)$  (или  $\{C^g(G)\}$ ). Оценивание указанных стоимости и близости может основываться на различных подходах (например, аддитивная функция, минимизация, функция 'min/max').

Теперь задача перекраски графа может быть рассмотрена следующим образом:

Найти новую *конфигурацию цветов*  $C^*(G)$  для заданного графа  $G = (A, E)$  так, чтобы минимизировать близость результирующей *конфигурации цветов* графа  $C^*(G(A, E))$  к целевой *конфигурации цветов* графа  $C^g(G(A, E))$  с учетом ограниченной интегрированной стоимости перекраски ( $\leq \overline{D}$ ):

$$\min_{\{C(G=(A,E))\}} \rho(C^*(G), C^g(G)) \quad s.t. \ D(C^o(G), C^*(G)) \leq \overline{D}, \ C^*(a_i) \neq C^*(a_j) \ \forall (a_i, a_j) \in E, \ i \neq j.$$

Рис. 17 иллюстрирует задачу перекраски вершин графа:

$$C^o(G) = (P_2 \star Q_3 \star U_3 \star V_2 \star W_1) \Rightarrow C^*(G) = (P_1 \star Q_2 \star U_2 \star V_1 \star W_3).$$

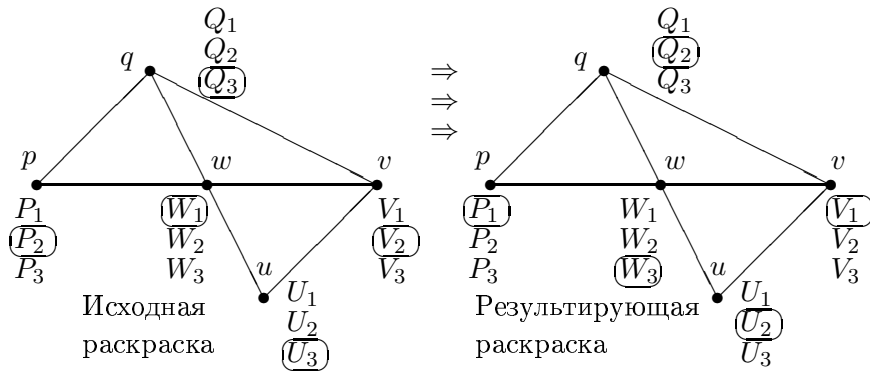


Рис. 17. Пример перекраски вершин

2.8. Построение иерархии

**Иерархическая кластеризация**

Иерархическая кластеризация реализует процесс (Снизу-Вверх) по интеграции элементов (объектов) для получения иерархии на множестве исходных объектов. Пусть имеется множество элементов (объектов):  $A = \{A_1, \dots, A_i, \dots, A_n\}$ . Для оценки объектов имеется  $r$  характеристик (параметров)  $(T_1, \dots, T_\xi, \dots, T_r)$  и для каждого объекта  $\forall A_i \in A$  вектор оценок имеет вид:  $z_i = (z_{i,1}, \dots, z_{i,\xi}, \dots, z_{i,r})$ . Базовый алгоритм для иерархической кластеризации (agglomerative algorithm, полиномиальный, алгоритм 1) используется давно и заключается в интеграции пар 'близких' объектов ([171], [173]):

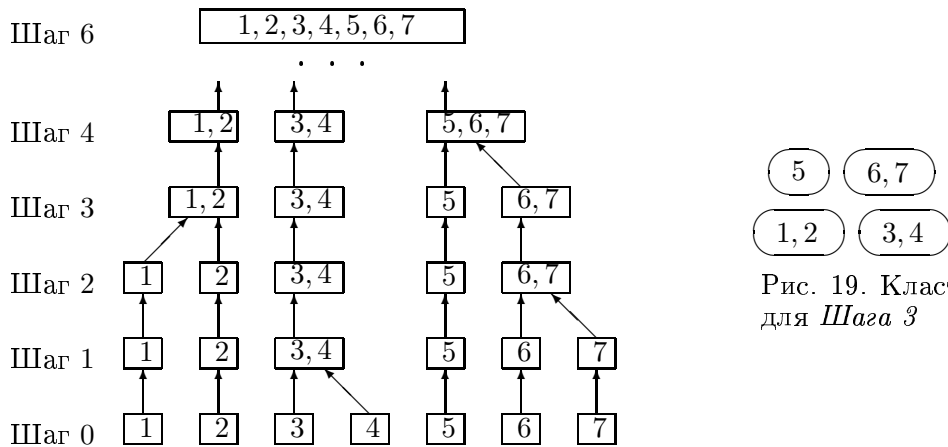


Рис. 19. Кластеры для Шага 3

Рис. 18. Пример иерархической кластеризации

*Стадия 1.* Вычисление матрицы парных "расстояний":  $\forall (A(i_1), A(i_2)), A(i_1) \in A, A(i_2) \in A, i_1 \neq i_2$  (в простейшем случае можно использовать метрику  $l_2$ ):  $d_{i_1 i_2} = \sqrt{\sum_{\xi=1}^r (z_{i_1, \xi} - z_{i_2, \xi})^2}$ .

*Стадия 2.* Выявление пары объектов с наименьшим "расстоянием" между ними и интеграция соответствующей пары объектов в один результирующий "интегрированный" элемент.

*Стадия 3.* Остановка процесса вычисления или вычисление новой матрица парных "расстояний" и переход к *Стадии 2*.

В результате получается древовидная структура (интегрирование одной пары объектов на каждом шаге процесса). Некий базовый подход к интеграции элементов (т.е., переход к средним характеристикам) имеет вид:  $(J_{i_1, i_2} = A_{i_1} \& A_{i_2}): \forall j z_{J_{i_1, i_2}, j} = \frac{z_{i_1, j} + z_{i_2, j}}{2}$ . Очевидно, что интеграцию пары объектов можно осуществлять на основе других функций, например:

max, min. Интеграция нескольких объектов может проводиться аналогично. Рис. 18 содержит иллюстративный пример иерархической кластеризации [225]. На Рис. 19 представлены кластеры, полученные на шаге 3 примера. Очевидно, что вычислительная сложность шага интеграции равна  $O(n)$  и сложность алгоритма 1 равна  $O(n^3r)$ . Следует отметить, что на каждом шаге алгоритма получаются кластеры без пересечения (т.е. без общих элементов).

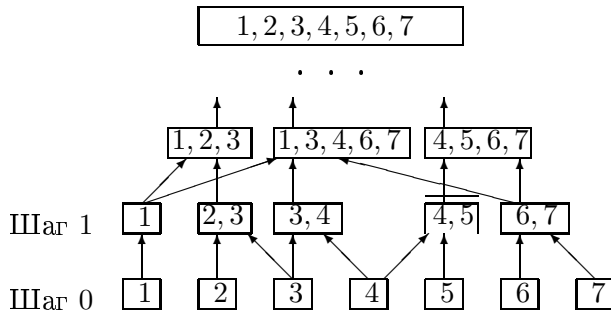


Рис. 20. Иллюстрация иерархии

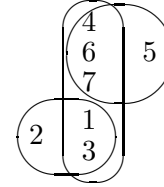


Рис. 21. Кластеры для Шага 2

Модификация агломеративного подхода к иерархической кластеризации была предложена в [225]. Данная версия алгоритма направлена на генерацию иерархии кластеров, которые могут включать общие элементы. Кроме того, могут использоваться более 'корректные' методы определения 'близости' объектов. В алгоритм включены следующие дополнительные операции: (i) построение порядковой 'близости' для пар объектов (в том числе на базе использования многокритериальных подходов); (ii) возможность интеграции нескольких пар объектов на одном шаге алгоритма; (iii) возможность включения одного и того же элемента в несколько различных интегрированных объекта (т.е., кластера). На Рис. 20 и Рис. 21 представлен пример, иллюстрирующий работу модифицированного алгоритма. Результаты работы данного алгоритмы похожи на размытую кластеризацию (fuzzy clustering) ([165] и др.).

С другой стороны, процессы кластеризации могут быть рассмотрены на основе оптимизационного подхода (т.е., как решение некой оптимизационной задачи). Обычно при кластеризации используются следующие целевые функции: (1) расстояние между объектами внутри получающихся кластеров (inter-cluster "distances"), (2) расстояние между объектами из различных кластеров (intra-cluster "distances"). Дополнительно могут использоваться следующие цели: число получающихся кластеров (или близость к заданному интервалу числа кластеров), мощность кластеров (т.е., число элементов в кластерах) и др. В некоторых работах указываются и другие целевые функции ([165], [181]). Таким образом можно формулировать и использовать многокритериальные модели кластеризации на основе указанных критериев (целевых функций).

Приведенная иерархическая кластеризация может использоваться как основа для процедур построения иерархий систем, включая интерактивные человеко-машинные процедуры.

### Иерархические покрывающие структуры

Основная задача *покрытия* заключается в построении *минимального покрывающего дерева* или *покрывающего дерева минимального веса* ([104], [126], [127], [133], [253], [295], [367], [374]). Пусть дан связный граф  $G = (A, E)$  ( $A$  - множество вершин,  $E$  - множество ребер/дуг) с неотрицательными весами ребер/дуг. Покрывающее дерево для графа представляет собой подграф  $T = (A, E')$  ( $E' \subseteq E$ ), который является деревом и соединяет все вершины. Общий вес (стоимость) покрывающего дерева  $c(T)$  определяется как сумма весов всех его ребер/дуг (т.е.,  $E'$ ). *Минимальное покрывающее дерево* или *покрывающего дерева минимального веса*

$T^*$  - это покрывающее дерево, у которого общий вес меньше или равен весу любого другого покрывающего дерева  $c(T^*) = \min_{\{T\}} c(T)$ . Эта задача является полиномиально разрешимой ([126], [127], [133], [295], [374] и др.). Имеется несколько известных полиномиальных алгоритмов, которые широко используются (Prim's algorithm, Kruskal's algorithm, Boruvka's algorithm и др.) ([85], [133] и др.). В более общем случае покрывающая структура соответствует *покрывающему лесу*: некий граф (не обязательно связный) имеет *минимальный покрывающий лес*, который представляет собой объединение минимальных покрывающих деревьев для связных компонентов исходного графа ([133], [296]).

В последние годы активно исследуется задача *многокритериального покрывающего дерева* ([74], [98], [132], [156], [197]). При этом вектор весов соответствует каждому ребру (дуге) и целевая функция представляет собой вектор, где каждый компонент равен сумме соответствующих компонентов веса-вектора ребра (дуги):  $\vec{c}(T)$ . Ясно, что здесь целесообразно проводить поиск Парето-эффективных (по  $\vec{c}(T)$ ) решений. Эта задача является NP-трудной (даже в случае двух критериев). В качестве алгоритмов в основном используются следующие ([9], [74], [197], [273]): (i) многокритериальная версия алгоритма Прима (Prim) для покрывающего дерева, (ii) многокритериальная версия алгоритма Крускала (Kruskal) для покрывающего дерева, (iii) генетические алгоритмы, (iv) алгоритмы многокритериальной эволюционной оптимизации, (v) эвристики (например, GRASP алгоритмы, алгоритмы "отжига" - simulated annealing algorithm), (vi) подходы на основе систем знаний.

В случае *задачи дерева Штейнера* ([3], [76], [106], [107], [133], [139], [167], [170], [199], [363]), покрывающее дерево может включать дополнительные вершины. Таким образом, общий вес (стоимость) результирующей покрывающей структуры может быть меньше, чем в случае базовой *задачи покрывающего дерева*. Базовая формулировка *задачи дерева Штейнера* имеет следующий вид. Дан неориентированный граф  $G = (A, E)$ ,  $A$  - множество узлов/вершин,  $E$  - множество ребер, неотрицательные веса ассоциированы с ребрами. Дополнительно дано множество вершин (точки Штейнера)  $Q$  (это терминальные или обычные вершины). Задача заключается в следующем:

*Найти подграф минимальной стоимости  $T_s = (Q, E') \subseteq G$  ( $E' \subseteq E$ ) такой, что существует путь в подграфе  $T_s$  между всеми парами базовых вершин.*

Здесь оптимальное решение  $T_s$  является деревом и называется деревом Штейнера. *Задача дерева Штейнера* является NP-трудной.

В случае направленного графа необходимо искать направленный подграф минимальной стоимости для заданного направленного графа, при этом подграф должен содержать направленные пути между корнем и каждой базовой вершиной.

Список основных приложений *задачи минимального дерева Штейнера* включает следующее: (i) коммуникационные сети, в том числе проектирование топологий ([76], [106], [107], [118], [290], [355]), маршрутизация при групповой передаче данных (multicast routing) ([284], [285], [305]), многоточечная маршрутизация (multipoint routing) (например, [169]), и др.; (ii) транспортные сети ([76] и др.); (iii) сети на основе многозвеньевых линий/путей (multi-hop network) ([30], [88], [353]); (iv) проектирование сверх-больших интегральных схем (VLSI design) ([144] и др.); (v) распределенные вычислительные системы ([134] и др.); (vi) фидерная реконфигурация в электрических сетях распределений ([12] и др.); (vii) распределенные вычислительные сети ([134] и др.); (viii) компьютерная биология ([76] и др.).

Среди различных вариантов постановок *задачи дерева Штейнера* можно указать следующие: (1) задача дерева Штейнера с минимальным числом точек Штейнера [107]; (2) задача дерева Штейнера с узкими местами (bottleneck Steiner tree problem) [107]; (3) задача дерева Штейнера с взвешенными узлами ([146], [196], [325], [386]); (4) задача дерева Штейнера на основе сбора "призов" (prize-collecting Steiner problem) (e.g., [152], [167], [246], [344]); (5) зада-

ча "живучей" сети Штейнера (survivable Steiner network problem) [107]; (6) задачи раскраски дерева Штейнера (Steiner tree coloring problem) [107]; (7) задача теории расписаний на дереве Штейнера (Steiner tree scheduling problem) [107]; (8) ограниченная задача дерева Штейнера (constrained Steiner tree problem) ([88], [311]); (9) задача дерева Штейнера с ограничением на числе звеньев в пути (Steiner tree problem with hop constraints) ([88], [353]); (10) задача дерева Штейнера с прибылями (Steiner tree problems with profits) [87]; (11) обобщенная задача Штейнера (generalized Steiner problem) ([2], [13], [103], [104], [359]); (12) обобщенная задача Штейнера на звездной структуре (generalized Steiner star problem) [194]; (13) стохастическая задача дерева Штейнера (stochastic Steiner tree problem) ([147], [148]); (14) динамическая задача дерева Штейнера (dynamic Steiner tree problem) [169]; (15) задача дерева Штейнера в режиме реального времени (on-line Steiner tree problem) ([6], [13], [359]); (16) групповая задача дерева Штейнера (group Steiner tree problem) ([72], [109], [200]).

В основном, используются следующие схемы решения: (1) точные алгоритмы (переборные алгоритмы, например: ветви и границ, ветвей и разрезов) ([79], [199], [246]) и (2) различные эвристики ([88], [106], [352], [354]) включая следующее (а) быстрые (жадные) алгоритмы ([72], [88], [202]); (б) приближенные алгоритмы ([2], [28], [69], [107], [146], [149], [152], [196], [201], [381], [382], [386]); (в) генетические алгоритмы ([185] и др.); (г) методы на основе искусственного интеллекта ([182] и др.); (д) эвристики для решения двойственной задачи (dual heuristics) ([95], [365]); (е) распределенная эвристика для одновременного решения прямой и двойственной задач (distributed primal-dual heuristic) [318]; (ж) методы локального поиска [57]. Следует отметить, что была разработана специальная библиотека численных примеров для задачи Штейнера на графах [200].

Очевидно, что можно также рассматривать и *многокритериальную задачу дерева Штейнера* ([219], [243], [356]). Здесь рассматривается вес в виде вектора для каждого ребра (дуги) и используется соответствующая векторная целевая функция (например, как вектор из соответствующих сумм в простейшем случае):  $\vec{c}(T_s)$ . Таким образом, проводится поиск Парето-эффективных решений (по  $\vec{c}(T_s)$ ) [243]. При этом можно использовать следующие подходы:

- (i) эвристики [356];
- (ii) специальные много-уровневые макро-эвристики: (а) "разбиение-синтез" эвристика [219], (б) эвристика на основе покрывающего дерева [243].

Специальная версия задачи дерева Штейнера на плоскости была исследована в [219]). Дан неориентированный связный граф  $G = (A, E)$ ,  $A$  - множество узлов/вершин  $E$  - множество ребер, неотрицательные веса ассоциированы с ребрами. дано дополнительное множество вершин  $Q$  и имеются соответствующие дополнительные ребра (с ассоциированными весами)  $W = \{(q, a)\}$ ,  $\forall q \in Q, \forall a \in A$ . Две операции (отбор подмножества  $Q' \subseteq Q$  и размещения его элементов) реализованы в рамках процесса решения. В результате получается граф:  $G' = (A \cup Q, E \cup W)$ . Задача имеет вид:

*Найти подграф минимальной стоимости  $T_s = (A \cup Q', E') \subseteq G'$  ( $E' \subseteq E \cup W$ ) такой, что существует путь в подграфе  $T_s$  между любыми парами вершин.*

Здесь оптимальное решение  $T_s$  представляет собой дерево (дерево Штейнера). При этом дополнительное множество вершин  $Q'$  рассматривается как дополнительный (ограниченный) ресурс (например, дополнительные вершины, сумма весов вершин).

### Об оптимальных иерархиях

В последние годы интерес к построению 'оптимальных' иерархий возрос в нескольких прикладных областях, например: наука об организациях, управление, информационные системы, транспорт, сети, распределенные компьютерные системы, биоинформатика ([145], [216], [265],

[267], [351], [369]). Очевидно, что здесь могут использоваться методы на основе покрывающих структур (см. предыдущий раздел) или прямые методы (на основе непосредственного использования оптимизационных подходов). При этом можно рассмотреть два типа задач: 1. построить иерархию, которая обладает наилучшим значением некоторого ее свойства (или нескольких свойств); 2. построить иерархию, которая является наиболее 'близкой' к некоторой заданной идеальной иерархии (или набору идеальных иерархий).

Пусть задано исходное множество графов (структур)  $\{G_i = (A_i, E_i), i = \overline{1, n}\}$ . Пусть также задана некая целевая (идеальная) структура  $G^g(A_g, E_g)$  ( $G^g \in \{G_i\}$ ), пусть  $\rho(G_{i_1}, G_{i_2})$  будет определять близость (или 'расстояние') между графами  $\forall G_{i_1}, G_{i_2} \in \{G_i\}$ . Дополнительно следует рассмотреть свойства графа (например, степень вершины, связность вершин)  $G_{i_1} \in \{G_i\}$ :  $\overline{\psi(G_{i_1})} = (\psi_1(G_{i_1}), \dots, \psi_\mu(G_{i_1}), \dots, \psi_m(G_{i_1}))$ . Здесь ищется граф  $G^* \in \{G_i\}$ . Тогда в задаче первого типа (случай максимизации значения свойства) целевая функция имеет вид:  $\max_{G^* \in \{G_i\}} \psi_\mu(G^*) \quad \forall \mu = \overline{1, m}$ . Для второго типа задачи имеем следующую целевую функцию:  $\min_{G^* \in G_i} \rho(G^*, G^g)$ . Ясно, что можно использовать интегрированные формулировки задач, например:

$$\min_{G^* \in G_i} \rho(G^*, G^g) \quad s.t. \quad \psi_\mu(G^*) \geq d^\mu \quad \forall \mu = \overline{1, m},$$

где  $(d^1, \dots, d^\mu, \dots, d^m)$  - вектор ограничений на значения свойств. Указанные оптимизационные задачи, как правило, соответствуют сложным соответствующим моделям целочисленного (или смешанного целочисленного) программирования, для которых используются различные схемы решения (например, переборные методы, эвристики, методы искусственного интеллекта).

### 2.9. Родственные Задачи

В нескольких прикладных областях (например, сенсорные сети, радарные системы, обработка последовательностей изображений) применяются постановки задач в виде многомерных назначений (сопоставлений) для поиска ассоциаций данных. Такие задачи возникают, в частности, при выявлении и сопровождении большого числа объектов в сенсорных системах ([18], [19], [21], [36], [73], [78], [97], [179], [213], [276], [289], [299], [300]). Многомерностные задачи назначения (т.е., задачи сопоставления в  $n$ -дольном графе) интенсивно исследуются в последние годы ([17], [133], [151], [160], [188]). Рис. 22 иллюстрирует динамическую задачу сопоставления для 3-х дольного графа.

Дополнительно следует указать, что исследуются комбинаторные задачи в области современных подходов к динамическому выявлению некоторых "плотных" структур (в том числе в случае многих информационных источников), например: (i) выявление "плотного" ядра (dense core) в заданном множестве взаимосвязанных объектов ([266], [279] и др.); (ii) выявление последовательных образов (или множеств объектов) в потоках данных (data streams) ([77], [81], [372] и др.); (iii) интеграция (или синтез) графов/структур ([236] и др.).

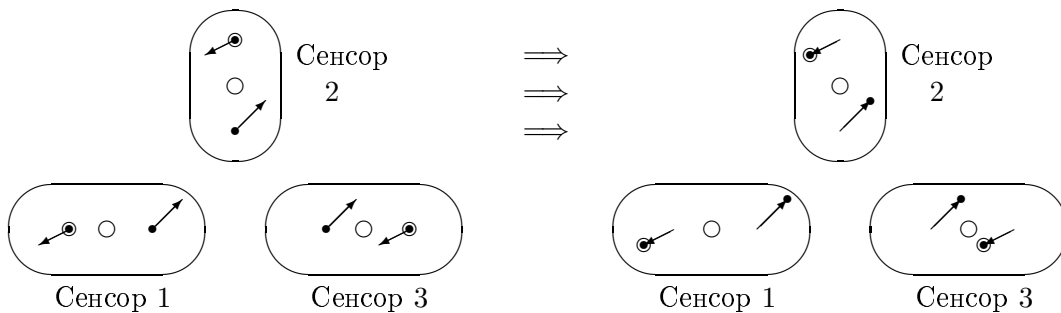


Рис. 22. Пример динамической (двух-стадийной) задачи 3-сопоставление

## 3. БАЗОВЫЕ ТЕХНОЛОГИЧЕСКИЕ ЗАДАЧИ КОНФИГУРАЦИИ СИСТЕМ

В данном разделе базовая задача построения конфигурации системы рассматривается как поиск (отбор) множества системных компонентов (Рис. 2, Таблица 1). Можно указать следующие случаи для исходного множества базовых альтернативных элементов или проектных альтернатив DAs: (а) основной случай (множество элементов), (б) множество совместимых элементов, (в) пересечение множеств элементов, соответствующих различным частям системы, (г) иерархическая многоуровневая модель. Наш набор прикладных проектных задач включает следующее (Таблица 10):

Таблица 10. Технологические задачи конфигурации систем (отбор/размещение)

#	Тип	Название	Данные	Результат	Модель/схема решения
1.	<b>P</b>	Проектирование конфигурации (без совместимости)	Набор множеств элементов (альтернатив)	Множество представителей	1. Задача о представителях 2. Задача о блочном рюкзаке 3. Назначение/размещение
2.	<b>G</b>	Проектирование конфигурации с совместимостью	Набор множеств элементов с совместимостью	Множество совместимых представителей	1. Задача совместимых представителей 2. Задача кратчайшего пути 3. Задача назначения/размещения с бинарными отношениями 4. Морфологическая клика 5. Клика в многодольном графе/кластеризация в многодольном графе
3.	<b>R</b>	Реконфигурация (изменение конфигурации)	Набор элементов (альтернатив) - конфигурация системы	Новая конфигурация	1. Задача о блочном рюкзаке 2. Морфологическая клика (ИММП)
4.	<b>T</b>	Многостадийное проектирование траекторий конфигураций	Набор множеств элементов (альтернатив)-конфигураций	Траектория конфигурации системы	1. Морфологическая клика на двух уровнях 2. Морфологическая клика или задача о блочном рюкзаке & морфологическая клика / задача о кратчайшем пути
5.	<b>M</b>	Проектирование системы из многих продуктов (т.е., семейство продуктов)	Несколько конфигураций систем (продуктов)	Несколько конфигураций систем (с общими элементами)	1. Морфологическая клика (ИММП) 2. Другие методы

1. Базовая задача построения конфигурации системы: структура системы как множество компонентов (т.е., их реализаций) (**P**). Задача имеет вид:

*поиск (отбор) множества системных компонентов (т.е., состав системы).*

При этом следующие модели могут использоваться: (i) задача о представителях (Рис. 5) [155], (ii) задача о блочном рюкзаке (т.е., без совместимости элементов, Рис. 4) ([133], [190], [237], [256]), (iii) задача назначения/размещения ([91], [133], [319]).

2. Построение конфигурации системы с совместимостью компонентов (**G**). Здесь могут использоваться следующие модели: (i) задача о совместимых представителях [198], (ii) задача кратчайшего пути ([10], [133]), (iii) кластеризация в многодольном графе ([71], [251]), задача

клик в многодольном графе ([94]), (iv) задача морфологической клики (ИММП) ([219], [221], [222]) (Рис. 7 и Рис. 8).

**3. Задача реконфигурации:** построение новой конфигурации системы на основе предыдущей конфигурации системы (**R**). В общем случае, структура системный изменений может иметь вид: ([219], [224]): 1. компонент/проектная альтернатива: 1.1. улучшенная прежняя проектная альтернатива (системный компонент) DAs, 1.2. новая проектная альтернатива DAs; 2. улучшенная совместимость; 3. новая структура системы. В задаче реконфигурации рассматриваются, в основном, случаи 1.1, 1.2 и 3. Задача о блочном рюкзаке может быть использована для изменение системных компонентов (случаи 1.1, 1.2) [237]. Рис. 23 иллюстрирует процесс перепроектирования (модификации, реконфигурации) системы: от исходной системы  $S_a$  к модифицированной системе (посредством изменение компонентов):  $S_a \Rightarrow S_b$ . ИММП может быть использовано для более сложных задач, когда имеются оценки совместимости между изменениями компонентов ([219], [221], [222]).

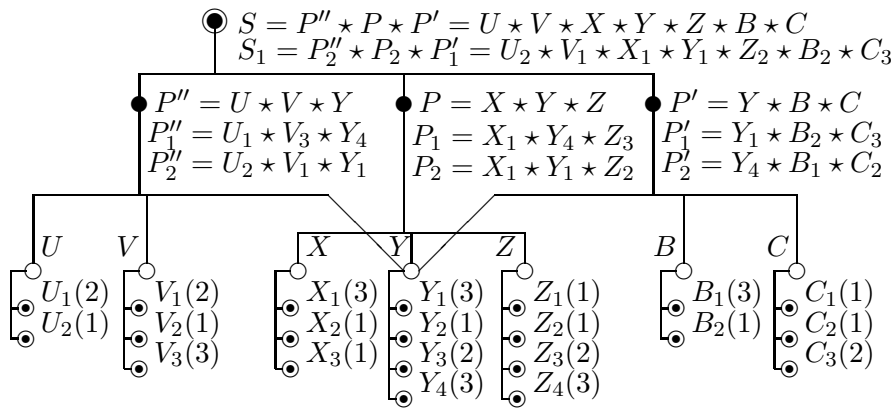
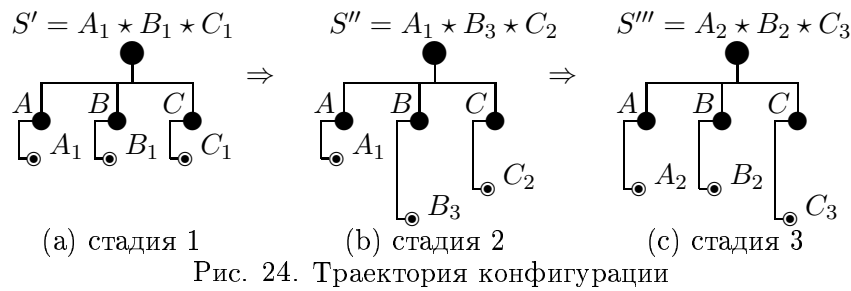
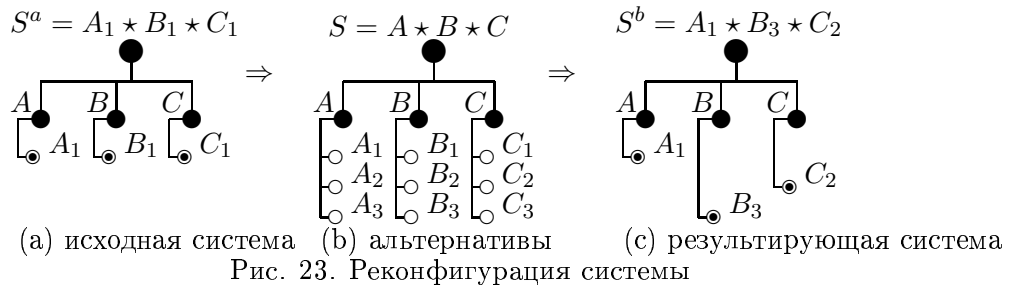


Рис. 25. Трех-продуктовая система с одним общим модулем

**4. Многостадийная задача для построения траектории конфигурации (Т):** построение конфигурации системы для нескольких последовательных стадий (т.е., траектории конфигурации системы).



Эта задача может быть рассмотрена как двухстадийный процесс: (а) построение исходной конфигурации системы, (б) проведение нескольких последовательных системных реконфигураций. Другая возможная макростратегия (двух-уровневый процесс проектирования) представляет собой следующее: (i) построение конфигурации системы (или нескольких конфигураций) на каждой стадии (использование задачи о блочном рюкзаке или ИММП), (ii) на каждой стадии выбор наилучшей конфигурации с учетом совместимости между отобранными конфигурациями (использование ИММП). Эта стратегия соответствует ИММП ([219], [221], [222]). Рис. 24 иллюстрирует 3-стадийную траекторию конфигурации системы:  $\alpha = \langle S', S'', S''' \rangle$ .

5. Построение конфигурации системы для многопродуктовой системы (**М**). Следует отметить, что важность семейств продуктов значительно возрастает. В [222] показано, что ИММП может быть использован как основа для проектирования многопродуктовых систем. Рис. 25 иллюстрирует 3-продуктовую систему с одним общим модулем.

Таблица 11 содержит список ряда прикладных примеров задач конфигурации систем (эти задачи были подготовлены автором и опубликованы).

Таблица 11. Список прикладных примеров

#	Приложение	Тип задачи	Ссылки
1.	Информационные системы	<b>G, R</b>	[219]
2.	Программный комплекс	<b>G, R</b>	[221], [222]
3.	Пользовательский интерфейс	<b>G, T</b>	[217], [219]
4.	Группа сотрудников	<b>G, R</b>	[219]
5.	Размещение персонала	<b>G</b>	[219]
6.	План учебного курса	<b>G</b>	[219]
7.	Стратегия решения	<b>G</b>	[219]
8.	Человеко-машинная система	<b>G</b>	[219], [220]
9.	План геологических исследований	<b>G</b>	[219]
10.	Технология бетона	<b>G</b>	[222], [231]
11.	Здания	<b>R</b>	[222], [234]
12.	Вибрационный конвейер	<b>G</b>	[219]
13.	Стратегия маркетинга	<b>G</b>	[222]
14.	Жизненный цикл продукции	<b>G</b>	[222]
15.	План медицинского лечения	<b>G</b>	[222], [233]
16.	Обработка сигналов	<b>T</b>	[222]
17.	Телеметрическая система	<b>G</b>	[240]
18.	Иммунотехнологический анализ	<b>G</b>	[222], [235]
19.	Тестирование системы	<b>G</b>	[236]
20.	GSM коммуникационная сеть	<b>G</b>	[238]
21.	Система безопасности	<b>G</b>	[241]
22.	Коммуникационное оборудование	<b>P, R</b>	[237]
23.	Беспроводной сенсор	<b>G, R</b>	[239]
24.	Задача 'последняя миля'	<b>P, G, R</b>	[229]
25.	Электронная торговля	<b>P, G</b>	[228]
26.	Информационная инфраструктура	<b>P, G, T</b>	[242]

#### 4. О ЗАДАЧАХ ДИНАМИЧЕСКОГО РЕКОНФИГУРИРОВАНИЯ СИСТЕМ

В последние годы все больше внимания уделяется задачам динамической реконфигурации систем, состоящих из компонентов (component-based systems) ([25], [75], [101], [136], [140], [150], [164], [211], [258], [350], [368]). При этом рассматриваются многие сетевые системы. Обычно рассматриваемые системы состоят из различных компонентов как узлов и ребер/дуг (например, программы, электронные системы, коммуникационная аппаратура, информация). Часто системные компоненты являются междисциплинарными. Базовая инженерная задача

динамической реконфигурации в реальном времени (online) имеет следующий вид [258]: (i) накопление требований к изменению системы; (ii) отбор системных компонентов для реконфигурации; (iii) проектирование операций реконфигурации (iv) отбор временного интервала для спроектированных операций реконфигурации; (v) выполнение операций реконфигурации. Эта *инженерная* задача динамической реконфигурации может быть реализована в виде схемы/плана решения (на основе комбинаторных оптимизационных моделей (*схема А* как 1-компонентная реконфигурация):

*Стадия А.1.* Накопление новых требований (требования к системным изменениям) на основе интеграции статистических данных об эксплуатации системы.

*Стадия А.2.* Остановка процесса накопления требований об изменении системы и реконфигурация: (*А.2.1*) отбор компонента системы для реконфигурации (комбинаторные модель: многокритериальный выбор/ранжирование); (*А.2.2*) проектирование составного плана реконфигурации системы (комбинаторные модели: рюкзак, блочный рюкзак, морфологический синтез на основе морфологической клики); (*А.2.3*) выбор временного интервала для изолирования системного компонента и его реконфигурации (комбинаторные модели: многокритериальный выбор и/или специальные оптимизационные подходы в виде непрерывной оптимизации); (*А.2.4*) выполнение плана реконфигурации системного компонента.

Очевидно, что на стадиях указанного плана могут также быть использованы методы искусственного интеллекта. Дополнительно следует рассмотреть *схему В* для случая многокомпонентной реконфигурации:

*Стадия В.1.* Накопление новых требований (требования к системным изменениям).

*Стадия В.2.* Остановка процесса накопления требований об изменении системы и реконфигурация: (*В.2.1*) отбор множества компонентов системы для реконфигурации (комбинаторные модели: многокритериальный выбор/ранжирование, задача о рюкзаке, задача блочного рюкзака); (*В.2.2*) проектирование составного плана реконфигурации системы для каждого отобранного компонента системы (комбинаторные модели: рюкзак, блочный рюкзак, морфологический синтез на основе морфологической клики); (*В.2.3*) выбор временного интервала для изолирования системных компонентов и их реконфигурации (комбинаторные модели: многокритериальный выбор и/или специальные оптимизационные подходы в виде непрерывной оптимизации); (*В.2.4*) выполнение плана реконфигурации системных компонентов.

Обе представленные схемы реконфигурации могут быть расширены для случая многостадийной реконфигурации (т.е., для построения траектории системных конфигураций). Такой подход будет актуальным при выявленном тренде требований к изменению системы.

## 5. ИСПОЛЬЗОВАНИЕ СИСТЕМНЫХ КОНФИГУРАЦИЙ В ПРЕПОДАВАНИИ

Рассмотренные задачи построения системных конфигураций являются чрезвычайно полезными в преподавании в таких областях, как техника, компьютерные науки, управление, прикладная математика. Умение строить конфигурации систем является критичным со следующих точек зрения: (i) моделирование систем, (ii) анализ и оценки модульных систем, (iii) выявления узких мест, (iv) проектирование и перепроектирование (upgrade) модульных, (v) модульное построение стратегий решения задач. Курс автора по проектированию систем, в котором используются конфигурации систем, описан в ряде статей ([223], [226]). Некоторые результаты студенческих исследовательских проектов опубликованы ([237], [238], [239], [240], [241]).

## 6. ЗАКЛЮЧЕНИЕ

В статье были представлены несколько основных проектных подходов к построению конфигураций систем: (i) поиск (отбор) множества (или структуры) системных компонентов;

(ii) поиск (отбор) множества (или структуры) системных компонентов с учетом совместимости компонентов; (iii) размещение системных компонентов; (iv) поиск (отбор) множества (или структуры) системных компонентов с учетом совместимости компонентов и размещение отобранных компонентов; (v) построение системных иерархий. Указанные подходы основываются на моделях комбинаторной оптимизации (рюкзак, назначение/размещение, клика и т.п.).

Три прикладные технологические задачи рассмотрены как дополнительные постановки: (а) реконфигурация систем как перепроектирование структур систем, (б) многостадийное построение и перепроектирование системных конфигураций, (в) проектирование и перепроектирование конфигураций систем для многопродуктовых систем (т.е., систем, включающих несколько продуктов-частей, общие части). Результирующая общая схема задач построения конфигураций систем, основанная на моделях комбинаторной оптимизации, представлена на Рис. 26. Следует подчеркнуть, что представленный материал представляет собой один из первых шагов в данной области. Представляется очевидным, что практически все части указанной схемы требуют продолжения исследований.



Рис. 26. Общая схема задач построения конфигураций систем

Список перспективных направлений исследования включает следующее: (1) анализ дополнительных проблем конфигураций (например, построение конфигураций систем на основе многоуровневой иерархической схемы как иерархической кластеризации [225] или на основе экспертных процедур, построения 'оптимальных' иерархий, проектирование системных конфигураций с учетом структуры связи между отобранными системными компонентами); (2) исследования моделей построения системных конфигураций с отбором и размещением системных компонентов; (3) исследования рассмотренных комбинаторных моделей с точки зрения робастной оптимизации (robust optimization) и учета неопределенности (uncertainty) ([29], [120], [172], [203], [271], [272], [349], [373]); (4) исследование новых специальных типов иерархических покрывающих структур; (5) дополнительное исследование других комбинаторных моделей (например, задачи о покрытии); (6) анализ моделей, основанных на мульти-

множествах; (7) исследование задач динамической реконфигурации для различных сетевых систем (электрические сети, сети распределения тепла, логистические сети, коммуникационные сети и др.), включая рассмотрение процесса реконфигурации как переключения (switching processes) (например, коммуникационные радио сети с направленными антеннами); (8) исследование задач динамической реконфигурации для различных прикладных распределенных компьютерных систем; (9) моделирование и анализ эволюции (развития) системных конфигураций; (10) исследование перспективных приложений (например, конфигурации сложных составных математических моделей, конфигурации схем решения задач, конфигураций в области комбинаторной химии); (11) исследование многопродуктовых систем (семейства продуктов, платформы); (12) исследование задач построения конфигураций систем для динамических систем с учетом сценариев развития/эволюции систем [380]; (13) анализ взаимосвязи рассмотренных задач и исследовательского направления "управление конфигурациями" (*configuration management*) ([83], [115]); (14) анализ рассмотренных задач с точки зрения распределенного (децентрализованного) подхода к решению (15) использование задач построения конфигураций систем в технологическом прогнозировании ([14], [43], [193], [222], [364]); (16) использование задач построения конфигураций систем в преподавании (инженерные науки, компьютерные науки, управление, прикладная математика).

## СПИСОК ЛИТЕРАТУРЫ

1. Agerfalk P.J., Brinkkemper S., Gonzalez C., Henderson-Seller B., Karlsson F., Kelly S., Ralyte J., Modularization constructs in method engineering: Towards common ground? *IFIP International Federation for Information Processing*, 2007, vol. 244, pp. 359-368.
2. Agrawal A., Klein P., Ravi R., When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. on Computing*, 1995, vol. 24, no. 3, pp. 440-456.
3. Ahuja R.K., Magnanti T.L., Orlin J.B., *Network Flows, Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice Hall, 1993.
4. Айзерман М.А., Гусев Л.А., Петров С.В., Смирнов И.М., Динамический подход к анализу структурных представителей графов. (основы графодинамики). I и II. *Автоматика и телемеханика*, 1977, вып. 7, С. 135-151 и вып. 9, С. 123-136.
5. Akella M.R., Batta R., Sudit M., Rogerson P., Blatt A., Cellular network configuration with co-channel and adjacent-channel interference constraints. *Comp. & Oper. Res.*, 2008, vol. 35, no. 12, pp. 3738-3757.
6. Alon N., Awerbuch B., Azar Y., Buchbinder N., Naor J., A general approach to online network optimization problems. *ACM Trans. on Algorithms*, 2006, vol. 2, no. 4, pp. 640-660.
7. Antonsson E.K., Cagan J. (Eds.), *Formal Engineering Design Synthesis*. UK: Camb. Univ. Press, 2001.
8. Applegate L.M., Konsynski B.R., Nunamaker J.F., Model management systems: designs for decision support. *DSS*, 1986, vol. 2, no. 1, pp. 81-91.
9. Arroyo J.E.C., Vieira P.S., Vianna D.S., A GRAPS algorithm for the multi-criteria minimum spanning tree problem. *Annals of Operations Research*, 2008, vol. 159, no. 1, pp. 125-133.
10. Артамонов Е.И., Хачумов В.М., *Синтез структур специальных средств машинной графики*. Москва: ИПУ РАН, 1991.
11. Avanthay C., Hertz A., Zufferey N., A variable neighborhood search for graph coloring. *EJOR*, 2003, vol. 151, no. 2, pp. 379-388.
12. Avella P., Villacci D., Sforza A., A Steiner arborescence model for the feeder reconfiguration in electric distribution networks. *EJOR*, 2005, vol. 164, no. 2, pp. 505-509.
13. Awerbuch B., Azar Y., Bartal Y., On-line generalized Steiner problem. *Theoretical Computer Science*, 2004, vol. 324, no. 2-3, pp. 313-324.

14. Ayres R.U., *Technological Forecasting and Long-Time Planning*. New York: McGraw-Hill, 1969.
15. Bachoore E.H., Boldaender H.L., *Convex Recoloring of Leaf-Colored Trees*. TR UU-CS-2006-010, Dept. of Inform. & CS, Utrecht Univ., 2006.
16. Baligh H.H., *Organization Structures: Theory and Design, Analysis and Prescription*. New York: Springer, 2006.
17. Bang-Jensen J., Gutin G., Alternating cycles and paths in edge-colored multigraphs: A survey. *Discr. Math.*, 1997, vol. 167-168, pp. 101-114.
18. Bar-Shalom Y., Fortmann T., *Tracking and Data Association*. New York: Academic Press, 1988.
19. Bar-Shalom Y., Li X.-R., *Multitarget-Multisensor Tracking: Principles and Techniques*, Connecticut, USA: YBS Publishing, 1995.
20. Bar-Yehuda R., Fekdman I., Rawitz D., Improved approximation algorithm for convex recoloring of trees. *Theory of Computing Systems*, 2008, vol. 43, no. 1, pp. 3-18.
21. Barbosa P.R., Chong E.K.P., Suvorova S., Moran B., Multitarget-multisensor tracking in an urban environment: A closed-loop approach. *Proc. of SPIE*, 2008, vol. 6969, no. 69690W.
22. Barbosa V.C., Asis C.A.G., Do Nascimento J.O., Two novel evolutionary formulations of the graph coloring problem. *J. of Combinatorial Optimization*, 2004, vol. 8, no. 1, pp. 41-63.
23. Barker V., O'Connor D., Expert system for configuration at digital: Xcon and beyond. *Comm. of the ACM*, 1989, vol. 32, no. 3, pp. 298-318.
24. Basu A., Blanning R.W., Shtub A., Metagraphs in hierarchical modeling. *Manag. Sci.*, 1997, vol. 43, no. 5, pp. 623-639.
25. Batista T.V., Joolia A., Coulson G., Managing dynamic reconfiguration in component-based systems. *Proc. of the 2nd Eur. Workshop on Software Architecture EWSA '05*, 2005, pp. 1-17.
26. Belaziz M., Bouras A., Brun J.M., Morphological analysis for product design. *CAD*, 2000, vol. 32, no. 5, pp. 377-388.
27. Benatallah B., Sheng Q.Z., Dumas M., The self-serv environment for web services composition. *IEEE Internet Computing*, 2003, vol. 7, no. 6, pp. 40-48.
28. Berman P., Ramaiyer V., Improved approximations for the Steiner tree problem. *J. of Algorithms*, 1994, vol. 17, no. 3, pp. 381-408.
29. Bertsimas D., Sim M., *Robust Discrete Optimization*. Working Paper, Oper. Res. Center, MIT, 2002.
30. Bhatnagar S., Ganguly S., Izmailov R., Design of IEEE 802.16-based multi-hop wireless backhaul network. *Proc. of the 1st Int. Conf. on Access Networks*, Greece, 2006, Article no. 5.
31. Bi Z.M., Gruver W.A., Lang S.Y.T., Analysis and synthesis of reconfigurable robotic systems. *Concurrent Eng.: Res. & Appl.*, 2004, vol. 12, no. 2, pp. 145-153(9).
32. Binbasioglu M., Key features for model building decision support systems. *EJOR*, 1995, vol. 82, no. 3, pp. 422-437.
33. Birmingham W.P., Brennan A., Gupta A.P., Siewiorek D.P., Micon: A single board computer synthesis tool. *IEEE Circuits and Devices Magazine*, 1988, vol. 4, no. 1, pp. 37-46.
34. Blake M.B., Agent-based workflow configuration and management of on-line services. *Proc. of the Int. Conf. on Electronic Commerce Research (ICECR-4)*, Dallas, TX, 2001, pp. 567-588.
35. Blake M.B., Gomaa H., Agent-oriented compositional approaches to services-based cross-organizational workflow. *DSS*, 2005, vol. 40, no. 1, pp. 31-50.
36. Blackman S.S., *Multiple-Target Tracking with Radar Applications*. Dedham, MA: Artech House, 1986.
37. Bless P.N., Kapoor S.G., DeVor R.E., Klabjan D., An algorithmic strategy for automated generation of multicomponent software tools for virtual manufacturing. *J. of Manufacturing Science & Engineering*, 2005, vol. 127, no. 4, pp. 866-874.

38. Bless P.N., Klabjan D., Chang S.Y., Heuristics for automated knowledge source integration and service composition. *Comp. & Oper. Res.*, 2008, vol. 35, no. 4, pp. 1292-1314.
39. Blomqvist E., Levashova T., Ohgren A., Sandkuhl K., Smirnov A., Tarassov V., Configuration of dynamic SME supply chains based on ontologies. In: Marik, V., Brennan R.W., Pechoucek M. (Eds.), *Holonic and Multi-Agent Systems for Manufacturing*. LNAI 3593, Springer, 2005, pp. 246-256.
40. Blum A., New approximation algorithms for graph coloring. *J. of the ACM*, 1994, vol. 41, no. 3, pp. 470-516.
41. Bodlaender H.L., Kloks T., Tan R.B., Van Leeuwen J., Approximations for  $\lambda$ -coloring of graphs. *Computer Journal*, 2004, vol. 47, no. 2, pp. 193-204.
42. Bondalapati K., Prasanna V.K., Reconfigurable computing systems. *Proc. of the IEEE*, 2002, vol. 90, no. 7, pp. 1201-1217.
43. Boon B., Park Y., A systematic approach for identifying technology opportunities: Keyword-based morphology analysis. *Technol. Forecasting & Social Change*, 2005, vol. 72, no. 2, pp. 145-160.
44. Bordetsky A.B., Peltsverger B., Peltsverger S., Statnikov R.B., Multi-criteria approach in configuration of energy efficient sensor networks. *Proc. of 43rd ACM Southeast Regional Conf.*, 2005, vol. 2, pp. 28-29.
45. Bouhmala N., Granmo O.-C., Solving graph coloring problems using learning automata. In: J. van Hemert, C. Cotta (Eds.), *Proc. of 8th Eur. Conf. on Evolutionary Computation in Combinatorial Optimization EvoCOP 2008*, LNCS 4972, Springer, 2008, pp. 277-288.
46. Braha D., Design-as-satisfiability: A new approach to automated synthesis. *AI EDAM*, 2001, vol. 15, no. 5, pp. 385-399.
47. Brelaz D., New methods to color the vertices of a graph. *Comm. of the ACM*, 1979, vol. 22, no. 4, pp. 251-256.
48. Briggs P., Cooper K.D., Torczon L., Improvements to graph coloring register allocation. *ACM Trans. on Programming Languages and Systems*, 1994, vol. 16, no. 3, pp. 428-455.
49. Brinkkemper S., Method engineering: Engineering of information systems development methods and tools. *Inform.&Software Techn.*, 1996, vol. 38, no. 4, pp. 275-280.
50. Brooks R.L., On coloring the nodes of network. *Proc. Cambridge Phil. Soc.*, 1940, vol. 37, pp. 194-197.
51. Brown D.C., Defining configuring. *AI EDAM*, 1998, vol. 12, no. 4, pp. 301-305.
52. Bui T.N., Nguyen T.H., An agent-based algorithm for generalized graph colorings. In: *Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation GECCO'06*, Seattle, 2006, pp. 19-28.
53. Bui T.N., Nguyen T.H., Patel C.M., Phan K.-A.T., An ant-based algorithm for coloring graphs. *Discr. Appl. Math.*, 2008, vol. 156, no. 2, pp. 190-200.
54. Cagan J., Campbell M.I., Finger S., Tomiyama T., A framework for computational design synthesis: Model and applications. *J. of Comput. and Inform. Sci. in Eng.*, 2005, vol. 5, no. 3, pp. 171-181.
55. Campbell M.I., Cagan J., Kotovsky K., Agent-based synthesis of electromechanical design configurations. *J. of Mechanical Design*, 2000, vol. 122, no. 1, pp. 61-69.
56. Campbell M.I., Cagan J., Kotovsky K., The A-design approach to managing automated design synthesis. *Res. in Eng. Des.*, 2003, vol. 14, no. 1, pp. 12-24.
57. Canuto S.A., Resende M.G.C., Ribeiro C.C., Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 2001, vol. 38, no. 1, pp. 50-58.
58. Caramia M., Dell'Olmo P., Bounding vertex coloring by truncated multistage branch and bound. *Networks*, 2004, vol. 44, no. 4, pp. 231-242.
59. Caramia M., Dell'Olmo P., Italiano G.F., CHECKGOL: Improved local search for graph coloring. *J. of Discrete Algorithms*, 2006, vol. 4, no. 2, pp. 277-298.

60. Caramia S.A., Dell'Olmo P., Embedding a novel objective function in a two-phase local search for robust vertex coloring. *EJOR*, 2008, vol. 189, no. 3, pp. 1358-1380.
61. Caramia S.A., Dell'Olmo P., Coloring graphs by iterated local search transversing feasible and infeasible solutions. *Discr. Appl. Math.*, 2008, vol. 156, no. 2, pp. 201-217.
62. Cawse J.N., (Ed.), *Experimental Design for Combinatorial and High Throughput Material Development*. New York: J.Wiley & Sons, 2002.
63. Cela E., *The Quadratic Assignment Problem*. Dordrecht: Kluwer Academic Publishers, 1998.
64. Chams M., Hertz A., de Werra D., Some experiments with simulated annealing for coloring graphs. *EJOR*, 1987, vol. 32, no. 2, pp. 260-266.
65. Chang G.J., Lu C., Distance-two labelings of graphs. *Eur. J. of Combin.*, 2003, vol. 24, no. 1, pp. 53-58.
66. Chang G.J., Chen J.-J., Kuo D., Liaw S.-C., Distance-two labelings of graphs. *Discr. Appl. Math.*, 2007, vol. 155, no. 8, pp. 1007-1013.
67. Chari K., Model composition using filter spaces. *Inform. Syst. Res.*, 2002, vol. 13, no. 1, pp. 15-35.
68. Chari K., Model composition in a distributed environment. *DSS*, 2003, vol. 35, no. 3, pp. 399-413.
69. Charicar M., Chekuri C., Cheung T.-Y., Dai Z., Goel A., Guha S., Li M., Approximation algorithms for directed Steiner problems. *J. of Algorithms*, 1999, vol. 33, no. 1, pp. 73-91.
70. Chandra C., Grabis J., *Supply Chain Configuration*. New York: Springer, 2007.
71. Charon I., Hudry O., Optimal clustering of multipartite graphs. *Discr. Appl. Math.*, 2008, vol. 156, no. 8, pp. 1330-1347.
72. Chekuri C., Even G., Kortsarz G., A greedy approximation algorithm for the group Steiner problem. *Discr. Appl. Math.*, 2006, vol. 154, no. 1, pp. 15-34.
73. Chen G., Hong L., A genetic algorithm based multi-dimensional data association algorithm for sensor-multi-target tracking. *Math. and Computer Modelling*, 1997, vol. 26, no. 4, pp. 57-69.
74. Chen G., Chen S., Guo W., Chen W., The multicriteria minimum spanning tree problem based genetic algorithm. *Inform. Sci.*, 2007, vol. 177, no. 22, pp. 5050-5063.
75. Chen X., Simons M., A component framework for dynamic reconfiguration of distributed systems. In: Bisshop J., (Ed.), *Proc. of the IFIP/ACM Working Conf. on Component Deployment*, LNCS 2370, Springer, 2002, pp. 82-96.
76. Cheng X., Du D.-Z., *Steiner Trees in Industries*. Dordrecht: Kluwer Academic Publishers, 2001.
77. Cheng J., Ke Y., Ng W., A survey on algorithms for mining frequent itemsets over data streams. *Knowledge and Information Systems*, 2008, vol. 16, no. 1, pp. 1-127.
78. Chong C.-Y., Mori S., Barker W.H., Chang K.-C., Architectures and algorithms for track association and fusion. *IEEE Aerospace and Electronic Systems Magazine*, 2000, vol. 15, no. 1, pp. 5-13.
79. Chorpa S., Gorres E.R., Rao M.R., Solving the Steiner tree problem on a graph using branch and cut. *ORSA J. Comput.*, 1992, vol. 4, no. 3, pp. 320-355.
80. Ciuksys D., Caplinkas A., Reusing ontological knowledge about business processes in IS engineering: Process configuration problem. *Informatica*, 2007, vol. 18, no. 4, pp. 585-602.
81. Coble J., Cook D.J., Holder L.B., Structure discovery in sequentially-connected data streams. *Int. J. on Artif. Intell. Tools*, 2006, vol. 15, no. 6, pp. 917-944.
82. Colombo G., Mosca A., Sartori F., Towards the design of intelligent CAD systems: An ontological approach. *Advanced Engineering Informatics*, 2007, vol. 21, no. 2, pp. 153-168.
83. Conradi R., Westfechtel B., Versions models for software configuration management. *ACM Comput. Surv.*, 1998, vol. 30, no. 2, pp. 232-282.

84. Corbett B., Rosen D.W., A configuration design based method of platform commonization for product families. *AI EDAM*, 2004, vol. 18, no. 1, pp. 21-39.
85. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C., *Introduction to Algorithms*, 2nd ed., Boston: MIT Press and McGraw-Hill, 2001.
86. Costa D., Hertz A., Ants can color graphs. *J. of the ORS*, 1997, vol. 48, no. 3, pp. 295-305.
87. Costa A.M., Cordeau J.-F., Laporte G., Steiner tree problems with profits. *INFOR*, 2006, vol. 44, no. 2, pp. 99-115.
88. Costa A.M., Cordeau J.-F., Laporte G., Fast heuristics for the Steiner tree problem with revenues, budget and hop constraints. *EJOR*, 2008, vol. 190, no. 1, pp. 68-78.
89. Couch A., System configuration management. In: Bergstra, J., Burgess, M., Eds., *Handbook on Network and System Administration*, Amsterdam: Elsevier, 2008, pp. 75-133.
90. Couch A., Sun Y., On observed reproducibility in network configuration management. *Science of Computer Programming*, 2004, vol. 53, no. 2, pp. 215-253.
91. Current J., Min H., Schilling D., Multiobjective analysis of facility location decisions. *EJOR*, 1990, vol. 49, no. 3, pp. 295-300.
92. Current J., Marsh M., Multiple transportation network design and routing problems: taxonomy and annotation. *EJOR*, 1993, vol. 65, no. 1, pp. 4-19.
93. Daskin M.S., *Networks and Discrete Location. Models, Algorithms, and Applications*. New York: J.Wiley & Sons, 1995.
94. Dawande M., Keskinocak P., Swaminathan J.M., Tayur S., On bipartite and multipartite clique problems. *J. of Algorithms*, 2001, vol. 41, no. 2, pp. 388-403.
95. de Aragao M.P., Uchoa E., Werneck R.F., Dual heuristics on the exact solution of large Steiner problems. *Electronic Notes in Discrete Mathematics*, 2001, vol. 7, pp. 150-153.
96. de Werra D., An introduction to timetabling. *EJOR*, 1985, vol. 19, no. 2, pp. 151-162.
97. Deb S., Yeddapanudi M., Pattipati L., Bar-Shalom Y., A generalized s-d assignment algorithm for multisensor-multitarget state estimation. *IEEE Trans. on Aerospace and Electronic Systems*, 1997, vol. 33, no. 2, part 1, pp. 523-538.
98. Dell'Amico M., Maffioli F., Combining linear and non-linear objectives in spanning tree problems. *J. of Combinatorial Optimization*, 2000, vol. 4, no. 2, pp. 253-269.
99. Dietmar K., Channel assignment for cellular radio using neural networks. *IEEE Trans. on Vehicular Technology*, 1991, vol. 40, no. 1-2, pp. 188-193.
100. Dijkstra E.W. A note on two problems in connection with graphs. In: *Numerische Mathematics*, Amsterdam, The Netherlands, Mathematisch Centrum, 1956, vol. 1, pp. 269-271.
101. Дивеев А.И., Управляемые сети и их приложения. *Ж. выч. мат. и мат. физики*, 2008, том 48, вып. 8, С. 1510-1525.
102. Dorne R., Hao J.-K., Tabu search for graph coloring, T-coloring and set T-coloring. In: Voss S.A., Martello S., Osman I.H., Roucairol C., (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, 1999, pp. 77-92.
103. Dror M., Haouari M., Generalized Steiner problems and other variants. *J. of Combinatorial Optimization*, 2000, vol. 4, no. 4, pp. 415-436.
104. Dror M., Haouari M., Chaouachi J., Generalized spanning trees. *EJOR*, 2000, vol. 120, no. 3, pp. 583-592.
105. Дубов Ю.А., Травкин С.И., Якимец В.М., *Многокритериальные модели формирования и выбора вариантов систем*. Москва: Наука, 1986.



106. Du D.-Z., Smith J.M., Rubinstein J.H., *Advances in Steiner Trees*. Dordrecht: Kluwer Academic Publishers, 2000.
107. Du D.-Z., Hu X., *Steiner Tree Problems in Computer Communication Networks*. Singapore: World Scientific Publishing Co., 2008.
108. Duff I.S., On algorithms for obtaining a maximum transversal. *ACM Trans. on Mathematical Software*, 1981, vol. 7, no. 3, pp. 315-330.
109. Duin C.W., Volgenant A., Voss S., Solving group Steiner problems as Steiner problem. *EJOR*, 2004, vol. 154, no. 1, pp. 323-329.
110. Eiben A.E., Van Der Hauw J.K., Van Hemert J.I., Graph coloring with adaptive evolutionary algorithms. *J. of Heuristics*, 1998, vol. 4, no. 1, pp. 25-46.
111. Eiter T., Gottlob G., Identifying the minimal transversals of hypergraph and related problems. *SIAM J. on Computing*, 1995, vol. 24, no. 6, pp. 1278-1304.
112. Eiter T., Gottlob G., Hypergraph transversal computation and related problems in logic and AI. In: *Proc. of Eur. Conf. on Logics in AI JELIA 2002*, LNCS 2424, Springer, 2002, pp. 549-564.
113. Емельянов С.В., Озерной В.М., Ларичев О.И. и др., Выбор рациональных вариантов технологической шахты при многих критериях. *Горный журнал*, 1972, вып. 5, С. 62-71.
114. Ensen T.R., Toft B., *Graph Coloring Problems*. New York: J.Wiley&Sons, 1995.
115. Estublier J., Leblang D., Van Der Hoek A., Conradi R., Clemm G., Tichy W., Wiborg-Weber D., Impact of software engineering research on the practice of software configuration management. *ACM Trans. on Software Engineering and Methodology*, 2005, vol. 14, no. 5, pp. 383-430.
116. Even S., Itai A., Shamir A., On the complexity of timetable and multicommodity flow problems. *SIAM J. on Comput.*, 1976, vol. 5, no. 4, pp. 691-703.
117. Even G., Lotker Z., Ron D., Smorodinsky S., Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM J. on Computing*, 2003, vol. 33, no. 1, pp. 94-136.
118. Even S., Kortsarz G., Slany W., On network design problems: fixed cost flows and the covering Steiner problem. *ACM Trans. on Algorithms*, 2005, vol. 1, no. 1, pp. 74-101.
119. Fassina G., Miertus S., (Eds.), *Combinatorial Chemistry and Technologies: Methods and Applications*. 2nd ed., Boca Raton, FL: CRC Press, 2005.
120. Feder T., Motwani R., O'Callaghan L., Olston C., Panigrahy R., Computing shortest paths with uncertainty. *J. of Algorithms*, 2007, vol. 62, no. 1, pp. 1-16.
121. Feldman S.I., Software configuration management: past uses and future challenges. *Proc. of the 3rd Eur. Software Eng. Conf. ESEC'91*, LNCS 550, Springer, 1991, pp. 1-6.
122. Fleischanderl G., Friedrich G.E., Haselbock A., Schreiner H., Stumptner M., Configuring large systems using generative constraint satisfaction. *IEEE Intell. Syst. & Their Appl.*, 1998, vol. 13, no. 4, pp. 59-68.
123. Fleurent C., Ferland J.A., Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 1996, vol. 63, no. 3, pp. 437-461.
124. Fredman M.L., Tarjan R.E., Fibonacci heaps and their uses in improved network optimization algorithms. *J. of the ACM*, 1987, vol. 34, no. 3, pp. 596-615.
125. Funabiki N., Takefuji Y., Neural network parallel algorithm for channel assignment in cellular radio networks. *IEEE Trans. on Vehicular Technology*, 1992, vol. 41, no. 4, pp. 430-437.
126. Gabow H.N., Two algorithms for generating weighted spanning trees in order. *SIAM J. on Computing*, 1977, vol. 6, no. 1, pp. 139-150.
127. Gabow H.N., Galil Z., Spencer T., Tarjan R.E., Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 1986, vol. 6, no. 2, pp. 109-122.

128. Гафт М.Г., Миловидов Н.Н., Серов В.И., Гусев Е.Д., Метод принятия решений для выбора рациональных конфигураций автомобиля. *Проблемы и методы принятия решений в организационных системах управления*. Москва: ИСА РАН, 1982, С. 77-82.
129. Galbraith J.R., *Designing Organizations: An Executive Guide to Strategy, Structure, and Process Revised*. 2nd ed., New York: Jossey-Bass, 2001.
130. Galinier P., Hao J.-K., Hybrid evolutionary algorithms for graph coloring. *J. of Combinatorial Optimization*, 1999, vol. 3, no. 4, pp. 379-397.
131. Galinier P., Hertz A., A survey of local search methods for graph coloring. *Comp. & Oper. Res.*, 2006, vol. 33, no. 9, pp. 2547-2562.
132. Gao J., Lu M., Fuzzy quadratic minimum spanning tree problem. *Appl. Math. & Comput.*, 2005, vol. 164, no. 3, pp. 773-788.
133. Garey M.R., Johnson D.S., *Computers and Intractability. The Guide to the Theory of NP-Completeness*, San Francisco: W.H. Freeman & Company, 1979.
134. Gen-Huey C., Houle M.E., Ming-Ter K., The Steiner problem in distributed computing systems. *Inform. Sci.*, 1993, vol. 74, no. 1-2, pp. 73-96.
135. Georgas J.C., Taylor R.N., Towards a knowledge-based approach to architectural adaptation management. *Proc. of the 1st ACM SIGSOFT Workshop of Self-managed Systems*, New Beach, California, 2004, pp. 39-43.
136. Georgas J.C., van der Hoek A., Taylor R.N., Architectural runtime configuration management in support of dependable self-adaptation software. *ACM SIGSOFT Software Engineering Notes*, 2005, vol. 30, no. 4, pp. 1-6.
137. Gerlach J., Kuo F.-Y., An approach to dialog management for presentation and manipulation of composite models in decision support systems. *DSS*, 1990, vol. 6, no. 3, pp. 227-242.
138. Geyer P., Multidisciplinary grammars supporting design optimization of buildings. *Research in Engineering Design*, 2008, vol. 18, no. 4, pp. 197-216.
139. Гордеев Е.Н., Тарасцов О.Г., Задача Штейнера. Обзор. *Дискретная математика*, 1993, том 5, вып. 2, С. 3-28.
140. Гребенюк Г.Г., Крыгин А.А., Алгоритмы оптимизации числа переключений при реконфигурации сетей теплоснабжения, *Автоматика и телемеханика*, 2007, вып. 12, pp. 101-112.
141. Griggs J.R., Yeh R.K., Labeling graphs with a condition at distance 2. *SIAM J. on Discr. Math.*, 1992, vol. 5, no. 4, pp. 586-595.
142. Grishicheva J.V., Selivestrov A.V., A note on cliques in multipartite graph. [archiv: math/0404125v1 \[math.CO\]](http://math.ubc.ca/~math/co/), Apr 6, 2004.
143. Grosh D., Agarwal R., Model selection and sequencing in decision support systems, *OMEGA*, 1991, vol. 19, no. 2/3, pp. 157-167.
144. Grotschel M., Martin A., Weismantel R., The Steiner tree packing problem in VLSI design. *Math. Program.*, Ser. B, 1997, vol. 78, no. 2, pp. 2675-281.
145. Губко М.В., Поиск оптимальных организационных иерархий при однородных функциях затрат менеджеров. *Автоматика и телемеханика*, 2008, вып. 1, С. 97-113.
146. Guha S., Khuller S., Improved methods for approximating node weighted Steiner trees and connected dominating sets. *Information and Computation*, 1999, vol. 150, no. 1, pp. 57-74.
147. Gupta A., Pal M., Stochastic Steiner tree without root. In: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), *Proc. of 32nd Int. Colloquium on Automata, Languages and Programming ICALP 2005 LNCS 3580*, 2005, Springer, pp. 1051-1063.

148. Gupta A., Hajiaghayi M., Kumar A., Stochastic Steiner tree with non-uniform inflation. In: Charikar M., Jansen K., Reingold O., Rolim J.D.P. (Eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX-RANDOM 2007*, LNCS 4627, Springer, 2007, pp. 134-148.
149. Gupta A., Kumar A., Pal M., Roughgarden T., Approximation via cost sharing: simpler and better approximation algorithms for network design. *J. of the ACM*, 2007, vol. 54, no. 3, pp. 11-49.
150. Gupta D., Jalote P., Barua G., A formal framework for on-line software version change. *IEEE Trans. on Software Eng.*, 1996, vol. 22, no. 2, pp. 120-131.
151. Gutin G., Cycles and paths in semicomplete multipartite digraphs, theorems and algorithms: A survey. *J. of Graph Theory*, 1995, vol. 19, no. 4, pp. 481-505.
152. Gutner S., Elementary approximation algorithms for prize collecting Steiner tree problems. In: Yang B., Du D.-Z., Wang C.A. (Eds.), *2nd Annual Int. Conf. on Combinatorial Optimization and Applications COCOA 2008*, LNCS 5165, Springer, 2008, pp. 246-254.
153. Hack S., Goos G., Copy coalescing by graph recoloring. *ACM SIGPLAN Notices*, 2008, vol. 43, no. 6, pp. 227-237.
154. Hale W.K., Frequency assignment: theory and applications. *Proc. of the IEEE*, 1980, vol. 68, no.12, pp. 1497-1514.
155. Hall M., *Combinatorial Theory*, 2nd ed., New York: J.Wiley&Sons, 1986.
156. Hamacher H.W., Ruhe G., On spanning tree problem with multiple objectives. *Annals of Oper. Res.*, 1995, vol. 52, no. 4, pp. 209-230.
157. Hamirez J.-P., Hao J.-K., Scatter search for graph coloring. *Selected Papers from the 5th Eur. Conf. on Artificial Evolution*, 2001, pp. 168-179.
158. Harary F., *Graphs Theory*, Readings, MA: Addison-Wesley, 1964.
159. Harary F., Maybee J., (Eds.), *Graphs and Applications*, New York: Wiley, 1985.
160. Hazan E., Safra S., Schwartz O., On the hardness of approximating k-dimensional matching. *ECCC, TR03-20*, 2003.
161. Hebert C., Bretto A., Cremilleux B., A data mining formalization to improve hypergraph minimal transversal computation. *Fundamenta Informaticae*, 2007, vol. 80, no. 4, pp. 415-433.
162. Hertz A., de Werra D., Using tabu search techniques for graph coloring. *Computing*, 1987, vol. 39, no. 4, pp. 345-351.
163. Heydon A., Levin R., Mann T., Yu Y., *Software Configuration Management System Using Vesta*. New York: Springer, 2006.
164. Hillman J., Warren I., Open framework for dynamic reconfiguration. *Proc. of the 24 Int. Conf. on Software Engineering (ICSE 2004)*, Edinburgh, UK, 2004, pp. 594-603.
165. Hoppner F., Klawonn F., Kruse R., Runkler T., *Fuzzy Cluster Analysis*. New York: J.Wiley & Sons, 1999.
166. Hotz L., Wolter K., Krebs T., *Configuration in Industrial Product Families*. Amst.: IOS Press, 2006.
167. Hwang F.K., Richards D.S., Winter P., *The Steiner Tree Problem*. Amst.: Elsevier, 1992.
168. Iijima J., Automatic model building and solving for optimization software. *DSS*, 1996, vol. 18, no. 3-4, pp. 293-300.
169. Imase M., Waxman B.M., Dynamic Steiner tree problem. *SIAM J. on Discr. Math.*, 1991, vol. 4, no. 3, pp. 369-384.
170. Ivanov A.O., Tuzhilin A.A., *Minimal Networks: The Steiner Problem and Its Generalizations*. Boca Raton, FL: CRC Press, 1994.

171. Jain A.K., Murty M.N., Flynn P.J., Data clustering: a review. *ACM Comput. Surv.*, 1992, vol. 31, no. 3, pp. 264-323.
172. Janiak A., Kasperski A., The minimum spanning tree problem with fuzzy costs. *Fuzzy Optimization and Decision Making*, 2008, vol. 7, no. 2, pp. 105-118.
173. Jardine N., Sibson R., *Mathematical Taxonomy*, London: J.Wiley & Sons, 1971.
174. Jarke M., Rademacher F.J., The AI potential of model management and its central role in decision support. *DSS*, 1998, vol. 4, no. 4, pp. 387-404.
175. Jhoti H., Leach A.R., (Eds.), *Structure-based Drug Discovery*. New York: Springer, 2007.
176. Johnson D.S., Yannakakis M., Papadimitriou C.H., On generating all maximal independent sets. *Inf. Process. Lett.*, 1988, vol. 27, no. 3, pp. 119-123.
177. Johnson D.S., Aragon C.R., McGeoch L.A., Schevon C., Optimization by simulated annealing: An experimental evaluation, Part II: Graph coloring and number partitioning. *Oper. Res.*, 1991, vol. 39, no. 3, pp. 378-406.
178. Johnson D.S., Trick M.A., (Eds.), *Cliques, Coloring, and Satisfiability*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26, Providence: American Math. Soc., 1996.
179. Johnston S.L., An efficient decentralized multiradar multitarget tracker for air surveillance. *IEEE Trans. on Aerospace and Electronic Systems*, 2008, vol. 44, no. 2, pp. 655-677.
180. Jones J.C., *Design Methods*, New York: J.Wiley & Sons, 1981.
181. Jonyer I., Cook D.J., Holder L.B., Graph-based hierarchical conceptual clustering. *J. of Machine Learning Research*, 2001, vol 2, pp. 19-43.
182. Joobbani R., Siewiorek D.P., WEAVER: A knowledge-based routing expert. *IEEE Design and Test of Computers*, 1986, vol. 3, no. 1, pp. 12-23.
183. Kabale M., Jaskowski B., Generalized implicit enumeration algorithm for graph coloring. *Comm. of the ACM*, 1985, vol. 28, no. 4, pp. 412-418.
184. Kannan S., Proebsting T., Register allocation in structured programs. *J. of Algorithms*, 1998, vol. 29, no. 2, pp. 223-237.
185. Kapsalis A., Rayward-Smith V.J., Smith G.D., Solving the graphical Steiner tree problem using genetic algorithm. *J. of the ORS*, 1993, vol. 44, no. 4, pp. 397-406.
186. Karam N., Linckels S., Mainel C., Semantic composition of lecture subparts for a personalized e-learning. In: Franconi E., Kifer M., May W. (Eds.), *Proc. of 4th Eur. Semantic Web Conference ESWC 2007*, LNCS 4519, Springer, 2007, pp. 716-728.
187. Karger D., Motwani R., Sudan M., Approximate graph coloring by semidefinite programming. *J. of the ACM*, 1998, vol. 45, no. 2, pp. 246-265.
188. Karp R.M., Reducibility among combinatorial problems. In: Miller R.E., Thatcher J.W. (Eds.), *Complexity of Computer Computations*. New York: Plenum, 1972, pp. 85-103.
189. Kavvadias D.J., Stavropoulos E.C., An efficient algorithm for the transversal hypergraph generation. *J. of Graph Algorithms Appl.*, 2005, vol. 9, no. 2, pp. 239-264.
190. Kellerer H., Pferschy U., Pisinger D., *Knapsack Problems*. Berlin: Springer, 2004.
191. Kemp C., Tenenbaum J.B., The discovery of structural form. *PNAS*, 2008, vol. 105, no. 31, pp. 10687-10692.
192. Khachiyan L., Boros E., Elbassioni K., Gurvich V., An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals and its application in joint generation. *Discr. Appl. Math.*, 2006, vol. 154, no. 16, pp. 2350-2372.
193. Khadke K., Gershenson J.K., Technology change analysis for product and product platform design. *2007 Proc. of the ASME Int. Design Eng. Technical Conferences & Computers and Information in Eng. Conf. DETC2007*, 2008, vol. 3, part A, pp. 461-470.

194. Khuller S., Zhu A., The general Steiner star problem. *Inf. Proc. Lett.*, 2002, vol. 84, no. 4, pp.215-220.
195. Kirkwood C.W., Slaven M.P., Maltz A., Reconfiguring supply-chain decision making at IBM. *Interfaces*, 2005, vol. 35, no.6, pp. 460-473.
196. Klein P., Ravi R., A nearly best-possible approximation algorithm for node-weighted Steiner trees. *J. of Algorithms*, 1995, vol. 19, no. 1, pp. 104-115.
197. Knowles J.D., Corne D.W., Enumeration of Pareto optimal multi-criteria spanning trees - a proof of the incorrectness of Zhou and Gen's proposed algorithm. *EJOR*, 2002, vol. 143, no. 3, pp. 543-547.
198. Knuth D., Ratghunathan A., The problem of compatible representatives. *SIAM J. on Discr. Math.*, 1992, vol. 5, no. 3, pp. 422-427.
199. Koch T., Martin A., Solving Steiner tree problems in graphs to optimality. *Networks*, 1998, vol. 32, no. 3, pp. 207-232.
200. Koch T., Martin A., Voss S., *SteinLib: An Updated Library on Steiner Problems in Graphs*, ZIB-Report 00-37, 2000, <http://elib.zib.de/steinlib>
201. Konemann J., Leonardi S., Schafer G., A group-strategy proof mechanism for Steiner forest. *Proc. of the 16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2005, pp. 612-619.
202. Kou L., Markowsky G., Berman L., A fast algorithm for Steiner trees. *Acta Informatica*, 1981, vol. 15, no. 2, pp. 141-145.
203. Kouvelis P., Yu G., *Robust Descrete Optimization and Its Applications*. Dordrecht: Kluwer Academic Publishers, 1997.
204. Kramer J., Magee J., Dynamic configuration for distributed systems, *IEEE Trans. on Software Eng.*, 1985, vol. 11, no. 4, pp. 424-436.
205. Krikhaar R., Crnkovic I., Software configuration management. *Sci. of Comp. Progr.*, 2007, vol. 65, no. 3, pp. 215-221.
206. Krishnan R., PDM: A knowledge-based tool for model construction. *DSS*, 1991, vol. 7, no. 4, pp. 301-314.
207. Kroqsqaard P., Liljefors T., Madsen U., (Eds.), *Textbook of Drug Design and Discovery*. 3rd ed., Boca Raton, FL: CRC Press, 2002.
208. Kuhn, H.W., The Hungarian Method for the Assignment Problems, *Nav. Res. Log.*, 1955, vol. 2, no. 1-2, pp. 83-97 (reprinted in *Nav. Res. Log.*, 2005, vol. 52, no. 1, pp. 7-21).
209. Kuntz I.D., Structure-based strategies for drug design and discovery. *Science*, 1992, vol. 257, no. 5073, pp. 1078-1082.
210. Lai W.K., Coghill G.G., Channel assignment through evolutionary optimization. *IEEE Trans. on Vehicular Technology*, 1996, vol. 45, no. 1, pp. 91-96.
211. Lange S., Middendorf M., Multi task hyperreconfigurable architectures: models and reconfiguration problems. *Int. J. of Embedded Systems*, 2005, vol. 1, no. 3/4, pp. 154-164.
212. Ларичев О.И., Стернин М.Ю., Человеко-машинные методы решения многокритериальной задачи о назначениях. *Автоматика и телемеханика*, 1998, вып. 7, С. 135-156.
213. Левин М.Ш., Алгоритмы отождествления информации в АСУ. *Вопросы специальной радиоэлектроники, сер. Радиоэлектронная техника*, 1973, вып. 1, С. 95-101.
214. Левин М.Ш., Одна экстремальная задача организации данных. *Изв. АН СССР, Сер. Техн. Киберн.*, 1981, вып. 5, С. 103-112.
215. Левин М.Ш., *Применение комбинаторных оптимизационных моделей в автоматизированных системах*. Москва: ВНИИТЭМР, 1986.
216. Левин М.Ш., Иерархическая гипертекстовая система. *НТИ, сер 2.*, 1989, вып. 6, С. 10-13.

217. Levin, M.Sh., Hierarchical design of user interfaces, in: Blumenthal, B., Gornostaev J., Unger C., Eds., *Selected Papers of 4th Int. Conf. on Human-Computer Interaction EWHCI'94*, LNCS 876, Springer, 1994, pp. 140-151.
218. Levin M.Sh., Combinatorial design in hypertext. *5th Int. Conf on Inform. Syst. Development*, Gdansk, 1996, pp. 255-263.
219. Levin M.Sh., *Combinatorial Engineering of Decomposable Systems*, Dordrecht: Kluwer Academic Publishers, 1998.
220. Levin M.Sh., Towards combinatorial analysis, adaptation, and planning of human-computer systems. *Appl. Intell.*, 2002, vol. 16, no. 3, pp. 235-247.
221. Levin M.Sh., Modular system synthesis: Example for composite packaged software. *IEEE Trans. on SMC, Part C*, 2005, vol. 35, no. 4, pp. 544-553.
222. Levin M.Sh., *Composite Systems Decisions*. New York: Springer, 2006.
223. Levin M.Sh., Course 'System design: structural approach'. *18th Int. Conf. Design Meth. and Theory DTM2006*, DETC2006-99547, 2006.
224. Levin M.Sh., Combinatorial technological systems models (Examples for communication system). *2nd Int. Conf. on Syst. Eng. and Modeling*, Haifa, 2007, pp. 24-32.
225. Levin M.Sh., Towards hierarchical clustering. In: Diekert V., Volkov M., Voronkov A., (Eds.), *CSR 2007*, LNCS 4649, Springer, 2007, pp. 205-215.
226. Левин М.Ш., О преподавании проектирования систем. *Информ. технол. и вычисл. системы*, 2007, вып. 2, С. 89-94.
227. Levin M.Sh., Towards system configuration design. *Int. Conf. on Complex Systems ICCS 2007*, InterJournal *Complex Systems*, 2007, Paper 2162.
228. Levin M.Sh., Morphological approach to electronic shopping. *IEEE Region 8 Int. Conf. Sibircon-2008*, Novosibirsk, 2008, pp. 280-285.
229. Levin M.Sh., Selection of user's connection in last mile problems, 2008, (under submission).
230. Левин М.Ш., Михайлов А.А., *Фрагменты технологии стратификации множества объектов*, Препринт, Москва: ИСА РАН, 1988.
231. Levin M.Sh., Nisnevich M.L., Combinatorial scheme for management of life cycle: example for concrete macrotechnology. *J. of Intell. Manuf.*, 2001, vol. 12, no. 4, pp. 393-401.
232. Levin M.Sh., Elperin T., Algorithm scheme for spatio-temporal pattern synthesis. *Unpublished manuscript*, The Ben-Gurion Univ., Israel, 2003.
233. Levin M.Sh., Sokolova L.V., Hierarchical combinatorial planning of medical treatment. *Computer Methods and Programs in Biomedicine*, 2004, vol. 73, no. 1, pp. 3-11.
234. Levin M.Sh., Danieli M.A., Hierarchical morphological decision making framework for evaluation and improvement of composite systems (example for building). *Informatika*, 2005, vol. 16, no. 2, pp. 213-240.
235. Levin M.Sh., Firer M., Hierarchical morphological design of immunoassay technology. *Computers in Biology and Medicine*, 2005, vol. 35, no. 3, pp. 229-245.
236. Levin M.Sh., Last M., Design of test inputs and their sequences in multi-function system testing, *Appl. Intell.*, 2006, vol. 25, no. 1, pp. 105-124.
237. Левин М.Ш., Сафонов А.В., Проектирование и перепроектирование конфигурации оборудования в коммуникационной сети. *Информ. технологии и вычисл. системы*, 2006, вып. 4, pp. 63-73.
238. Levin M.Sh., Vishnitskiy R.O., Towards morphological design of GSM network. *Inform. Processes*, 2007, vol. 7, no. 2, 182-189.

239. Levin M.Sh., Fimin A.V., Hierarchical design of fire alarm wireless sensor element. *Proc. of 7th Int. Conf. CAD/CAM/PDM-2007*, Moscow, 2007, pp. 33-35.
240. Levin M.Sh., Khodakovskii I.A., Structural design of the telemetry system. *Automat. & Remote Control*, 2007, vol. 68, no. 9, pp. 1654-1661.
241. Левин М.Ш., Леус А.В., Модульное проектирование интегрированной системы безопасности. *Труды между. семинара 'Распределенные компьютерные и коммуникационные сети' DCCN 2007*, Москва, 2007, том 2, С. 16-22.
242. Levin, M.Sh., Grigorov, R.D., Configuration of applied Web-based systems. 2008 (submitted).
243. Levin M.Sh., Nuriahmetov R.I., Towards multicriteria Steiner tree problem, 2008 (in progress).
244. Lewandowsky G., Condon A., Experiments with parallel graph coloring heuristics and applications of graph coloring. In: Johnson D.S., Trick M.A., (Eds.), *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, Providence, RI: AMS, 1996, pp. 309-334.
245. Li L., Garibaldi J., Krasnogor N., Automated self-assembly programming paradigm: initial investigation, *Proc. of the Third IEEE Int. Workshop on Eng. of Autonomic and Autonomous Syst. EASe2006*, 2006, pp. 25-36.
246. Ljubic I., Weiskircher R., Pferschy U., Klau G., Mutzel P., Fischetti M., An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Math. Progr. Ser. B*, 2006, vol. 105, no. 2-3, pp. 427-449.
247. Любецкий В.А., Селивестров А.В., Многодольный граф с двумя вершинами в каждой доле. *Информационные процессы*, 2004, том 4, вып. 2, С. 127-132.
248. Lubetsky V.A., Selivestrov A.V., Note on cliques and alignments. *Inform. Processes*, 2004, vol. 4, no. 3, pp. 241-246.
249. Lucet C., Mendes F., Moukrim A., An exact method for graph coloring. *Comp. & Oper. Res.*, 2006, vol. 33, no. 8, pp. 2189-2207.
250. Lukasik, S., Kokosinski, Z., Swieton, G., Parallel Simulated Annealing Algorithm for Graph Coloring Problem, in: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J., Eds., *7th Int. Conf. on Parallel Processing and Appl. Math. PRAM 2007*, LNCS 4967, Springer, 2008, pp. 229-238.
251. Lytkin, N., Streltsov, S., Perlovsky, L., Muchnik, I., Petrov, S., Improving Representation of Search Results by Multipartite Graph Clustering of Multiple Reformulated Queries and a Novel Document Representation, *Manuscript*, 2005, [http://company.yandex.ru/grant/2005/02\\_Lytkin\\_\\_110104.pdf](http://company.yandex.ru/grant/2005/02_Lytkin__110104.pdf)
252. Madhusudan T., Uttamsingh N., A declarative approach to composing web services in dynamic environments, *DSS*, 2006, vol. 41, no. 2, pp. 325-357.
253. Magnanti T.L., Wolsey L.A., Optimal trees. In: M.O. Ball et al. (Eds.), *Handbooks in OR & MS*, Amsterdam: North-Holland, 1995, vol. 7, pp. 503-615.
254. Mailharro D., A classification and constraint-based framework for configuration. *AI EDAM*, 1998, vol. 12, no. 4, pp 383-397.
255. Malaguti E., Toth P., An evolutionary approach for bandwidth multicoloring problems. *EJOR*, 2008, vol. 189, no. 3, pp. 638-651.
256. Martello S., Toth P., *Knapsack Problem: Algorithms and Computer Implementation*, New York: J.Wiley & Sons, 1990.
257. Martins E.Q., On a multicriteria shortest path problem. *EJOR*, 1984, vol. 16, no. 2, pp. 236-245.
258. Matevska J., An optimized runtime reconfiguration on components-based software systems. *32rd Annual Int. IEEE Computer Software and Application Conf. COMPSAC 2008*, Turku, 2008, pp. 499-501.
259. McDermott J., R1: a rule-based configurer of computer systems. *Artif. Intell.*, 1982, vol. 19, no. 2, pp. 39-88.

260. McKeenly P.K., Sadjadi S.M., Kasten E.P., Cheng B.H.C., Composing adaptive software, *IEEE Computer*, 2004, vol. 37, no. 7, pp. 56-64.
261. Mehrabi M.G., Ulsoy A.G., Koren Y., Reconfigurable manufacturing systems, *J. of Intell. Manufact.*, 2000, vol. 11, no. 4, pp. 403-419.
262. Mehrotra A., Trick M.A., A column generation approach for graph coloring. *INFORMS J. on Computing*, 1996, vol. 8, no. 4, pp. 344-354.
263. Микрин Е.А., Некоторые задачи синтеза модульных систем обработки данных реального времени в АСУ международной космической станции "Альфа". *Автоматика и телемеханика*, 2001, вып. 1, С. 178-183.
264. Mintzberg H., *The Structuring of Organizations: A Synthesis of the Research*. Englewood Cliffs, NJ.: Prentice Hall, 1979.
265. Mirkin B., McMorris F.R., Robets F.S., Rzhetsky A., (Eds.), *Mathematical Hierarchies and Biology*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Providence: AMS, 1996.
266. Mirkin B., Muchnik I., Layered clusters of tightness set functions. *Appl. Math. Lett.*, vol. 15, no. 2, pp. 147-151.
267. Mishin S., Optimal organizational hierarchies in firms. *J. of Business Economics and Management*, 2007, vol. 8, no. 2, 79-99.
268. Mishra N., Ron D., Swaminathan R., A new conceptual clustering framework. *Machine Learning*, 2004, vol. 56, no. 1-3, pp. 115-151.
269. Mittal S., Frayman F., Towards a generic model of configuration tasks. *Proc. of the 11th Joint Conf. on Artif. Intell.*, 1989, pp. 1395-1401.
270. Mittal S., Falkenhainer B., Dynamic constraint satisfaction problem. *Proc. of the 8th National Conf. on Artif. Intell.*, 1990, pp. 25-32.
271. Montemanni R., Gambardella L., An exact algorithm for the robust shortest path problem with interval data. *Comp. & Oper. Res.*, 2004, vol. 31, no. 10, pp. 1667-1680.
272. Montemanni R., Gambardella L., A branch and bound algorithm for the robust spanning tree problem with interval data. *EJOR*, 2005, vol. 161, no. 3, pp. 771-779.
273. Moradkhan M.D., Browne W.N., A knowledge-based evolution strategy for the multi-objective minimum spanning tree problem. *2006 IEEE Congress on Evolutionary Computation CEC'2006*, 2006, article no. 1688471, pp. 1391-1398.
274. Moran S., Snir S., Efficient approximation of convex recoloring. *J. of Computer and System Sciences*, 2007, vol. 73, no. 7, pp. 1078-1089.
275. Moran S., Snir S., Convex recoloring of strings and trees: Definitions, hardness results and algorithms. *J. of Computer and System Sciences*, 2008, vol. 74, no. 5, pp. 850-869.
276. Morefield C.L., Application of 0-1 integer programming to multitarget tracking problems. *IEEE Trans. on Automatic Control*, 1977, vol. 22, no. 3, pp. 302-312.
277. Muhanna W.A., An object-oriented framework for model management and DSS development. *DSS*, 1993, vol. 9, no. 2, pp. 217-229.
278. Muhanna W.A., Pick R.A., Meta-modeling concepts and tools for model management: A systems approach, *Manag. Sci.*, 1994, vol. 40, no. 9, pp. 1093-1123.
279. Мулла́т И.Э., Экстремальные подсистемы монотонных систем. I и II. *Автоматика и телемеханика*, 1976, вып. 5, С. 130-139 и вып. 8, С. 169-178.
280. Muller-Lankenau C., Wehmeyer K., Klein S., Strategic channel alignment: an analysis of the configuration of physical and virtual marketing channels. *Inform. Syst. and E-Business Manag.*, 2006, vol. 4, no. 2, pp. 187-216.



281. Murat C., Pachos V.Th., On the probabilistic minimum coloring and minimum k-coloring. *Discr. Appl. Math.*, 2006, vol. 154, no. 3, pp. 564-586.
282. Murata S., Yoshida E., Kurikawa H., Tomita K., Kokaji S., Concept of self-reconfigurable modular robotic system. *Artificial Intelligence in Engineering*, 2001, vol. 15, no. 4, pp. 383-387.
283. Nemhauser G.L., Trick M.A., Scheduling a major college basketball conference. *Oper. Res.*, 1998, vol. 46, no. 1, pp. 1-8.
284. Novak R., Rugelj J., Kundus C., Steiner tree based distributed multicast routing in networks. In: *Steiner Trees in Industries*, Dordrecht: Kluwer Academic Publishers, 2001, pp. 327-352.
285. Oliveira C.A.S., Pardalos P.M., A survey of combinatorial optimization problems in multicast routing. *Comp. and Oper. Res.*, 2005, vol. 32, no. 8, pp. 1953-1981.
286. Oncan T., A survey on the generalized assignment problem. *INFOR*, 2007, vol. 45, no. 3, pp. 123-141.
287. Pachos V.Th., A survey of approximately optimal solutions to some covering and packing problems. *ACM Computing Surveys*, 1999, vol. 29, no. 2, pp. 171-209.
288. Pachos V.Th., Polynomial approximation and graph-coloring. *Computing*, 2003, vol. 70, no. 1, pp. 41-86.
289. Pardalos P.M., Wolkowicz H., (Eds.), *Quadratic Assignment and Related Problems*, Providence: AMS, 1994.
290. Pardalos P.M., Resende M.G.C., (Eds.), *Handbooks of Telecommunications*. New York: Springer, 2006.
291. Paredis C.J.J., Diaz-Calderon A., Sinha R., Khosla P.K., Composable models for simulation-based design. *Eng. with Comput.*, 2001, vol. 17, no. 2, pp. 112-128.
292. Павлов В.В., *Структурное моделирование в CALS-технологиях*. Москва: Наука, 2006.
293. Pei J., Jiang D., Zhang A., On mining cross-graph quasi-cliques. *Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, 2005, pp. 228-238.
294. Pemmaraju S., Skiena S., *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. New York: Cambridge Univ. Press, 2003.
295. Pettie S., Ramachandran V., An optimal minimum spanning tree algorithm. *J. of the ACM*, 2002, vol. 49, no. 1, pp. 16-34.
296. Pettie S., Ramachandran V., A randomized time-work optimal parallel algorithm for finding a minimum spanning forest. *SIAM J. on Computing*, 2002, vol. 31, no. 6, pp. 1876-1895.
297. Petzold L.R., Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Comput.*, 1983, vol. 4, no. 1, pp. 136-148.
298. Poladian V., Sousa J.P., Garlan D., Schmerl B., Shaw M., Task-based adaptation for ubiquitous computing. *IEEE Trans. on SMC, Part C*, 2006, vol. 36, no. 3, pp. 328-340.
299. Poore A.B., Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Comput. Optim. and Appl.*, 1994, vol. 3, no. 1, pp. 27-57.
300. Poore A.B., Rijavec N., A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking. *SIAM J. on Optimization*, 1993, vol. 3, no. 3, pp. 544-563.
301. Prokopska A., Application of morphological analysis methodology in architectural design. *Acta Polytechnica*, 2001, vol. 41, no. 1, pp. 46-54.
302. Pulvermuller E., Goos G., Assmann U., Preface: new software composition concepts. *Science of Computer Programming*, 2005, vol. 56, no. 1-2, pp. 1-4.
303. Рахмакулов В.З., Ахрем А.А., Герасимов В.В., Алгоритм распознавания объемных образов на основе модифицированного метода максимальной клики. В: *Управление информационными потоками*, Москва: ИСА РАН, 2002, С. 295-305.

304. Rakov D.L., Morphological synthesis method of the search for promising technical systems. *IEEE Aerospace & Elect. Syst. Magazine*, 1996, vol. 11, no. 12, pp. 3-8.
305. Ramanathan S., Multicast tree generation in networks with asymmetric links. *IEEE/ACM Trans. on Networking*, 1996, vol. 4, no. 4, pp. 558-568.
306. Ramanathan S., A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 1999, vol. 5, no. 2, pp. 81-94.
307. Ramanathan S., Lloyd E.L., Scheduling algorithms for multi-hop radio networks. *IEEE/ACM Trans. on Networking*, 1993, vol. 1, no. 2, pp. 166-177.
308. Ritchey T., Problem structuring using computer-aided morphological analysis. *J. of the ORS*, 2006, vol. 57, no. 7, pp. 792-801.
309. Robbes R., Lanza M., Lungu M., An approach to software evolution based on semantic change. In: Dwyer M.B., Lopes A., (Eds.), *Proc. of 10th Int. Conf. on Fundamental Approaches to Software Engineering FASE 2007*, LNCS 4422, 2007, Springer, pp. 27-41.
310. Rodriguez M.A., Jarur M.C., A genetic algorithm for searching spatial configurations. *IEEE Trans. on Evolutionary Computation*, 2005, vol. 9, no. 3, pp. 252-270.
311. Rosenwein M.B., Wong R.T., Constrained Steiner tree problem. *EJOR*, 1995, vol. 81, no. 2, pp. 430-439.
312. Ross A., Venkataramanan M.A., Ernstberger K.W., Reconfiguring the supply network using current performance data. *Decision Sciences*, 1998, vol. 29, no. 3, pp. 707-724.
313. Рубинштейн М.И., *Оптимальная группировка взаимосвязанных объектов*. Москва: Наука, 1989.
314. Sabin D., Freuder E.C., Configuration as composite constraint satisfaction. In: Interrante L.D., Luger G.F. (Eds.), *Workshop Notes of AAAI Fall Symp. on Configuration, USA*, 1996, pp. 26-36.
315. Sabin D., Weigel R., Product configuration frameworks - a survey. *IEEE Intell. Syst. & Their Appl.*, 1998, vol. 13, no. 4, pp. 42-49.
316. Sager T.J., Lin S.-J., Pruning procedure for exact graph coloring. *ORSA J. on Computing*, 1991, vol. 3, no. 3, pp. 226-230.
317. Sakashita M., Makino K., Nagamochi H., Fujishige S., Minimum transversals in posi-modular systems. In: Azar Y., Erlebach T. (Eds.), *Proc. of 14th Annual Eur. Symp. on Algorithms ESA 2006*, LNCS 4168, Springer, 2006, pp. 576-587.
318. Santos M., Drummond L.M.A., Uchoa E., A distributed primal-dual heuristic for Steiner problems in networks. In: C. Demetrescu (Ed.), *Proc. of 6th Int. Workshop on Experimental Algorithms WEA 2007*, LNCS 4525, Springer, 2007, pp. 175-188.
319. Scarelli A., Narula S.C., A multicriteria assignment problem. *J. of Multi-Criteria Dec. Anal.*, 2002, vol. 11, no. 2, pp. 65-74.
320. Schaerf A., A survey of automated timetabling. *Artif. Intell. Review*, 1999, vol. 13, no. 2, pp. 87-127.
321. Schierholt K., Process configuration: Combining the principles of product configuration and process planning. *AI EDAM*, 2001, vol. 15, no. 5, pp. 411-424.
322. Schmidt L.C., Cagan J., GGREADA: A graph grammar-based machine design algorithm. *Res. in Eng. Des.*, 1997, vol. 9, no. 4, pp. 195-213.
323. Schmidt L.C., Cagan J., Optimal configuration design: An integrated approach using grammars. *J. Mech. Des.*, 1998, vol. 120, no. 1, pp. 2-9.
324. Schmidt L.C., Shetty H., Chase S.C., A graph grammar approach for structure synthesis of mechanisms. *J. Mech. Des.*, 2000, vol. 122, no. 4, pp. 371-376.
325. Segev A., The node-weighted Steiner tree problem. *Networks*, 1987, vol. 17, no. 1, pp. 1-17.
326. Селивестров А.В., Любецкий В.А., Алгоритмы поиска консервативных участков нуклеотидных последовательностей. *Информационные процессы*, 2006, том 6, вып. 1, С. 33-36.

327. Sereni J.-S., Randomly colouring graphs (a combinatorial view). *Computer Science Review*, 2008, vol. 2, no. 2, pp. 63-95.
328. Sewell E.C., An improved algorithm for exact graph coloring. In: Johnson, D.S., Trick M.A., (Eds.), *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, Providence, RI: AMS, 1996, pp. 259-373.
329. Show M.J., Tu P.-L., De P., Applying machine learning to model management in decision support systems. *DSS*, 1988, vol. 4, no. 3, pp. 285-305.
330. Siddique Z., Rosen D.W., On combinatorial design spaces for the configuration design of product families. *AI EDAM*, 2001, vol. 15, no. 2, pp. 91-108.
331. Simon H.A. *The Science of the Artificial*. 3rd ed., Boston: MIT Press, 1996.
332. Smirnov A., Sheremetov L., Chilov N., Cortes J.R., Soft-computing technologies for configuration of cooperative supply chain, *Appl. Soft Comput.*, 2004, vol. 4, no. 1, pp. 87-107.
333. Smith D.H., Allen S.M., Hurley S., Characteristics of good meta-heuristics algorithms for the frequency assignment problem. *Annals of Operations Research*, 2001, vol 107, no. 1-4, pp. 285-301.
334. Skorin-Kapov J., Tabu search applied to the quadratic assignment problem, *ORSA J. Comput.*, 1990, vol. 2, no. 1, pp. 33-45.
335. Stefik M., *Introduction to Knowledge Systems*, San Francisco, CA: Morgan Kaufmann Publishers, 1995.
336. Stumptner M., Friedrich G.E., Haselbock A., Generative constraint-based configuration of large technical systems. *AI EDAM*, 1998, vol. 12, no. 4, pp 307-320.
337. Stupnikov S., Kalinichenko L., Bressan S., Interactive discovery and composition of complex Web services. In: Y. Manolopoulos, J. Pokorny, T. Sellis (Eds.), *Proc. of 10th East-West Eur. Conf. on Advances in Databases and Inform. Syst. ADBIS 2006*, LNCS 4152, Springer, 2006, pp. 216-231.
338. Sugumaran V., Tanniru M., Storney V.C., A knowledge-based framework for extracting components in agile systems development. *Information Technology & Management*, 2008, vol. 9, no. 1, pp. 37-53.
339. Tarapata Z., On a multicriteria shortest path problem. *Int. J. Appl. Math. Comput. Sci.*, 2007, vol. 17, no. 2, pp. 269-287.
340. Tate D.M., Smith A.E., A genetic approach to the quadratic assignment problem. *Comp. & Oper. Res.*, 1990, vol. 22, no. 1, pp. 73-83.
341. Thomsen E., *OLAP Solutions: Building Multidimensional Information Systems*. 2nd ed., New York: J.Wiley & Sons, 2002.
342. Ten T.A., Van Harmelen F., Schreiber A.Th., Wielinga B., Construction of problem-solving methods as parametric design. *Int. J. of Human Computer Studies*, 1998, vol. 49, no. 4, pp. 363-389.
343. Ten T.A., Van Harmelen F., Wielinga B., Configuration of Web services as parametric design. In: Davies, J., Fensel D., Bussler C., Studer R. (Eds.), *Proc. of the First Semantic Web Symposium: Research and Applications ESWS 2004*, LNCS 3053, Springer, 2004, pp. 299-311.
344. Uchoa E., Reduction tests for the prize-collecting Steiner problem. *Oper. Res. Lett.*, 2006, vol. 34, no. 4, pp. 437-444.
345. Van Den Heuvel J., Leese R.A., Shepherd M.A., Graph labeling and radio channel assignment. *J. of Graph Theory*, 1998, vol. 29, no. 4, pp. 263-283.
346. Vashist A., Kulikowsky C., Muchnik I, *Automatic Screening for Groups of Orthologous Genes in Comparative Genomics Using Multipartite Clustering*. TR 33, DIMACS, 2004.
347. Vashist A., Kulikowski C.A., Muchnik I., On bipartite and multipartite clique problems. *IEEE/ACM Trans. on Comput. Biology and Bioinformatics (TCBB)*, 2007, vol. 4, no. 1, pp. 17-27.
348. Vincke P., Problemes multicriteres. *Cahiers du Centre d'Etudes de Recherche Operationelle*, 1974, vol. 16, pp. 425-439 (in French).

349. Vincke P., Robust solutions and methods in decision-aid. *J. of Multicriteria Decision Analysis*, 1999, vol. 8, no. 3, pp. 181-187.
350. Vinh P.C., Bowen J.P., Continuity aspects of embedded reconfigurable computing. *Innovations in Systems and Software Engineering*, 2005, vol. 1, no. 1, pp. 41-53.
351. Воронин А.А., Мишин С.П., Алгоритмы поиска оптимальной структуры организационной системы. *Автоматика и телемеханика*, 2002, вып. 5, С. 120-132.
352. Voss S., Steiner's problem in graphs: heuristic methods. *Discr. Appl. Math.*, 1992, vol. 40, no. 1, pp. 45-72.
353. Voss S., The Steiner tree problem with hop constraints. *Annals of Oper. Res.*, 1999, vol. 86, pp. 321-345.
354. Voss S., Modern heuristic search methods for the Steiner tree problem in graph. In: Du D.-Z., Smith J.M., Rubinstein J.H., (Eds.), *Advances in Steiner Trees*, Boston, Kluwer Academic Publishers, 2000, pp. 283-323.
355. Voss S., Steiner tree problems in telecommunication. In: Resende, M., Pardalos P.M., (Eds.), *Handbook in Telecommunications*, New York: Springer, 2006, pp. 459-492.
356. Vujosevic M., Stanojevic M., A bicriteria Steiner tree problem on graph. *Yugoslav J. of Operations Research*, 2003, vol. 13, no. 1, pp. 25-33.
357. Wallace M., Schimpf J., Finding the right hybrid algorithm - A combinatorial meta-problem. *Annals of Math. & Artif. Intell.*, 2002, vol. 34, no. 4, pp. 259-269.
358. West D.B., *Introduction to Graph Theory*, 2nd ed., NJ: Prentice Hall, Englewood Cliffs, 2001.
359. Westbrook J., Yan D.C.K., Greedy algorithm for the on-line Steiner tree and generalized Steiner problems. In: Dehne F.K.H.A., Sack J.-R., Santoro N., Whitesides S. (Eds.), *Proc. of the Third Workshop on Algorithms and Data Structures WADS'93*, LNCS 709, Springer, 1993, pp. 622-633.
360. Wielinga B., Schreiber G., Configuration-design problem solving. *IEEE Expert: Intelligent Systems and Their Applications*, 1997, vol. 12, no. 2, pp. 49-56.
361. Wigderson A., Improving the performance guarantee for approximate graph coloring. *J. of the ACM*, 1983, vol. 30, no. 4, pp. 729-735.
362. Wilson S.R., Czarnik A.W., (Eds.), *Combinatorial Chemistry: Synthesis and Application*. New York: J.Wiley & Sons, 1997.
363. Winter P., Steiner problem in networks: A survey. *Networks*, 1987, vol. 17, no. 2, pp. 129-167.
364. Wissema J.G., Morphological analysis: Its application to a company TF-investigation. *Future*, 1976, vol. 8, no. 2, pp. 146-153.
365. Wong R.T., Dual ascent approach for Steiner tree problems on a directed graph. *Math. Progr.*, 1984, vol. 28, no. 3, pp. 271-287.
366. Wrembel R., Koncila C., *Data Warehouses and Olap: Concepts, Architectures and Solutions*. Hershey, PA: IGI Global, 2006.
367. Wu B.Y., Chao K.-M., *Spanning Trees and Optimization Problems*. Boca Raton, FL: CRC Press, 2004.
368. Wu J., Huang L., Wang D., ASM-based model of dynamic service update in OSGi. *ACM SIGSOFT Software Engineering Notes*, 2008, vol. 33, no. 2, article no. 8.
369. Wynants G., *Network Synthesis Problems*. Dordrecht: Kluwer Academic Publishers, 2001.
370. Xiang X.-D., Sun X., Briceno G., Lou Y., Wang K.-A., Chang H., Wallace-Freedman W.G., Chen S.-W., Schultz P.G., A combinatorial approach to material discovery. *Science*, 1995, vol. 268, no. 5218, pp. 1738-1740.
371. Xiang X.-D., Takeuchi I., *Combinatorial Material Synthesis*. Boca Raton, FL: CRC Press, 2003.
372. Xin D., Han J., Yan X., Cheng H., Compressing frequent patterns. *Data & Knowledge Engineering*, 2007, vol. 60, no. 1, pp. 5-29.

373. Yaman H., Karasan O.E., Pinar M.C., The robust spanning tree problem with interval data. *Oper. Res. Letters*, 2001, vol. 29, no. 1, pp. 31-40.
374. Yao A.C., An  $O(|E| \log \log |V|)$  algorithm for finding minimum spanning trees. *Information Processing Letters*, 1975, vol. 4, no. 1, pp. 21-23.
375. Yeh R.K., The edge soan of distance two labellings of graphs. *Taiwanese J. of Math.*, 2000, vol. 4, no. 4, pp. 675-683.
376. Yeh R.K., A survey on labeling graphs with a condition at distance two. *Discr. Math.*, 2006, vol. 306, no. 12, pp. 1217-1231.
377. Yigit A.S., Ulsoy A.G., Allahverdi A., Optimizing modular product design for reconfigurable manufacturing. *J. of Intell. Manuf.*, 2002, vol. 13, no. 2, pp. 309-316.
378. Yu T., Lin K.-J., Service selection algorithms for Web services selection with end-to-end QoS constraints. *J. of Information Systems and e-Business Management*, 2005, vol. 3, no. 2, pp. 103-126.
379. Yu T., Zhang Y., Lin K.-J., Efficient algorithms for Web services selection with end-to-end QoS constraints. *ACM Trans. on the Web*, 2007, vol. 1, no. 1, art. no. 6.
380. Юдицкий С.А., Операционно-целевое моделирование динамики развития организационных систем средствами сетей Петри. *Автоматика и телемеханика*, 2008, вып. 1, С. 114-123.
381. Zelikovsky A.Z., An 11/6-approximation algorithm for the network Steiner problem. *Algorithmica*, 1993, vol. 9, no. 5, pp. 463-470.
382. Zelikovsky A.Z., A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 1998, vol. 18, no. 1, pp. 99-110.
383. Zhang R., Vashist A., Muchnik I.B, Kulikovski C.A, Metaxas D.N. A new combinatorial approach to supervised learning: Application to gait recognition. In: Zhao W., Gong S., Tang X. (Eds.) *Proc. of the Second Int. Workshop on Analysis and Modeling of Faces and Gestures AMFG 2005*, LNCS 3723, Springer, 2005, pp. 55-69.
384. Zhuravlev Yu.I., An algebraic approach to recognition or classification problems. *Pattern Recognition and Image Analysis*, 1998, vol. 8, no. 1, pp. 59-100.
385. Zilberstein S., Russell S., Optimal composition of real-time systems. *Artif. Intell.*, 1996, vol. 82, no. 1-2, pp. 181-213.
386. Zou F., Li X., Kim D., Wu W., Two constant approximation algorithms for node-weighted Steiner tree in unit disk graphs. In: Yang B., Du D.-Z., Wang C.A. (Eds.), *Proc. of Second Annual Int. Conf. on Combinatorial Optimization and Applications COCOA 2008*, LNCS 5165, Springer, 2008, pp. 278-285.
387. Zwicky F., *Discovery Invention, Research Through the Morphological Approach*. New York: McMillan, 1969.