

О реконфигурации решений в комбинаторной оптимизации ¹

Марк Ш. Левин

*Институт проблем передачи информации, Российская академия наук
Большой Каретный пер. 19, Москва 127994, Россия
email: mslevin@acm.org*

Поступила в редколлегию 15.12.2016

Аннотация—В статье рассматривается новый класс задач модификации в комбинаторной оптимизации. Подход заключается в реконфигурации (реструктуризации) решений (в виде множества/структуры). Две характеристики процесса реконфигурации решения исследуются: (а) стоимость изменения решения, (б) близость получаемого реконфигурированного (реструктурированного) решения к некоторому целевому решению. Предлагаются три типа задач реконфигурации: (1) одностадийная реконфигурация решения, (2) многостадийная реконфигурация решения, (3) реконфигурация при изменении набора элементов решения. Рассматриваются постановки задачи с одним или многими критериями. Задачи реконфигурации решений соответствуют прикладным задачам перепроектирования (улучшения, обновления) модульных систем. Предложенный подход применен к двум задачам комбинаторной оптимизации: задаче о рюкзаке, задаче блочного рюкзака. Численные примеры иллюстрируют задачи реконфигурации и схемы решения.

КЛЮЧЕВЫЕ СЛОВА: комбинаторная оптимизация, реконфигурация, реструктуризация, рюкзак, блочный рюкзак, эвристики

1. ВВЕДЕНИЕ

Современные исследования в области комбинаторной оптимизации включают три основных направления: (i) многокритериальные задачи [20, 21, 49], (ii) задачи в условиях неопределенности [34, 43, 53, 56], (iii) динамические задачи [53, 54]), включая задачи реального времени [4, 28]. Данная статья представляет новый класс задач комбинаторной оптимизации, который направлен на реконфигурацию (реструктуризацию) решений задач. Указанные задачи соответствуют перепроектированию (улучшению, обновлению) модульных систем в различных приложениях [5, 10, 15, 32, 34, 42, 44, 45]. В качестве базовой задачи рассматривается одностадийная реконфигурация решений для двух моментов времени: τ_0 и τ_1 . Для каждого момента определяются соответствующие оптимальные решения S^0 and S^1 . Задача заключается в “дешевой” модификации решения S^0 в решение S^* так, чтобы полученное решение было очень близко к S^1 . Дополнительно предлагаются две задачи: (1) многостадийная реконфигурация (реструктуризация) решений, (2) реконфигурация (реструктуризация) решений при изменении множества элементов задачи. Используются следующие классификационные параметры для задач реконфигурации: (1) время: (а) одностадийная задача, (б) многостадийная задача; (2) тип критериев и/или оценок параметров: (i) однокритериальная задача, (ii) многокритериальная задача, (iii) задача с порядковыми оценками, (iv) задача с оценками в виде мультимножеств. Следует отметить, что похожий подход к реконфигурации решений в одностадийном варианте был предложен в работе [47].

¹ Работа выполнена при частичной финансовой поддержке гранта Российского фонда фундаментальных исследований ном. 15-07-01241.

В данной работе рассматриваются две комбинаторные задачи: задача о рюкзаке, задача блочного рюкзака. Следует отметить, подход к реконфигурации решений описан и применительно к некоторым другим задачам комбинаторной оптимизации: задачи кластеризации и сортировки кратко рассмотрены в [36], упрощенная задача упаковки в контейнеры рассмотрена в [37], задача покрывающих деревьев кратко рассмотрена в [35].

Основные общие предварительные материалы были опубликованы как электронные препринты [33, 35].

2. О МОДИФИКАЦИИ В КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Оптимизационная задача может быть представлена как комбинация трех компонентов: $P = \langle B, X, F \rangle$, где B - параметры задачи, X - множество элементов решения (например, аргументы), F - целевая функция (или функции). Таблица 1 содержит краткое описание основных модификационных подходов в оптимизации.

Таблица 1. Основные типы модификации в комбинаторной оптимизации

Ном.	Направление	Источники
I.	Модификация параметров задачи (B):	
1.1.	Инверсная оптимизация (модификация параметров задачи с минимальной ценой модификации так, чтобы заданное заранее решение стало оптимальным)	[2, 22, 29, 52]
1.2.	Реверсная оптимизация (модификация параметров задачи при условии удовлетворения заданного бюджетного ограничения и получения наилучшего возможного значения целевой функции)	[17, 57, 58]
1.3.	Динамическая комбинаторная оптимизация (включая задачи в реальном времени и задачи с изменением запросов)	[4, 28, 31, 53, 54]
1.4.	Задачи дополнения, задачи модификации графов (модификация, добавление/удаление вершин/ребер, корректировка параметров задачи для получения требуемых свойств решений)	[19, 24, 26, 27, 39] [41, 48, 50, 55]
II.	Модификация элементов решения задачи X :	
2.1.	Модификация элементов решения (переназначение, изменение связей, перемещение, редактирование, перекраска и др.) чтобы улучшить значение целевой функции (например, локальная оптимизация)	[9, 40, 59]
2.2.	Реоптимизация (малые изменения решения для улучшения его качества в виде улучшения значения целевой функции)	[6, 8, 11, 16]
2.3.	Реконфигурация/реструктуризация решения задачи с учетом двух критериев: минимум цена модификации решения, минимум близости к решению на следующей временной стадии:	
2.3.1.	Одностадийная реконфигурация/реструктуризация решений (построение реконфигурированного/реструктуризованного решения)	[33, 35, 36, 47]
2.3.2.	Многостадийная реконфигурация/реструктуризация решений (построение траектории реструктуризованных решений)	[35, 36]

Задачи инверсной и реверсной оптимизации рассматриваются следующим образом [18, 29]:

1. В реверсной оптимизации задан бюджет для модификации параметров задачи, в рамках этого бюджета необходимо требуется получить возможно лучшее значение целевой функции.

2. В инверсной оптимизации предварительно задается некоторое решение. Требуется изменить параметры задачи (при минимальной стоимости их изменения) чтобы заданное решение стало оптимальным.

Динамические задачи иллюстрированы на Рис. 1.

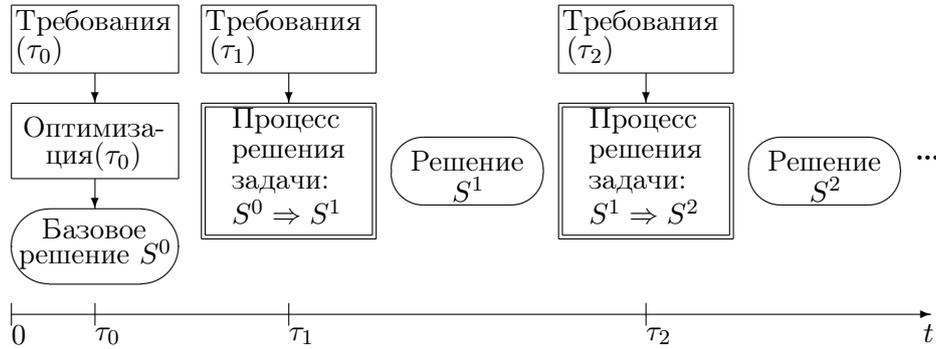


Рис. 1. Иллюстрация динамического решения задач [35]

Здесь новые требования получают в режиме реального времени и требуется решать задачу в каждый момент времени [1, 25, 31, 51]. Для Рис. 1, траектория решений может быть представлена так: $\hat{S} = \langle S^0 \rightarrow S^1 \rightarrow S^2 \rightarrow \dots \rangle$.

Задачи дополнения (augmentation) направлены на получение решений с требуемыми свойствами (Рис. 2 [35]), например [24, 27, 39, 50]: (а) требуемый уровень связности сетей при проектировании топологий сетей (например, k -связный сети); (б) требуемый тип структуры графа/сети (например: дерево или иерархия с требуемыми характеристиками, множество клик/квази-клик). Здесь можно также привести класс задач модификации графов (модификация вершин/ребер исходного графа для получения определенного свойства) [19, 41, 48, 55].

Методы и приемы улучшения (модификации) параметров полученных решений много лет используются, например, в методах локальной оптимизации [3, 9, 30, 40, 59].

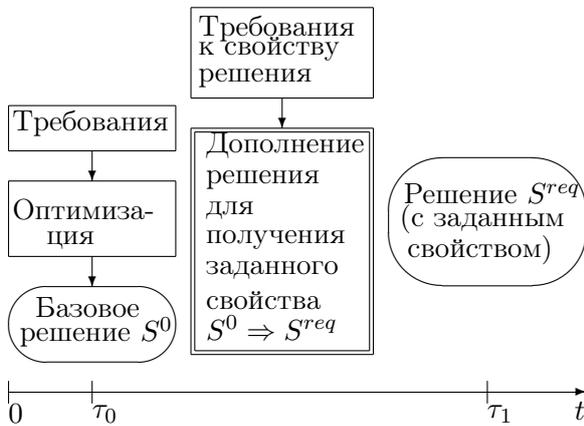


Рис. 2. Схема задачи пополнения



Рис. 3. Схема реоптимизации

В последние годы проводятся исследования в области реоптимизации известных задач комбинаторной оптимизации, например: (i) задача коммивояжера [6], (ii) задачи составления расписаний [46], (iii) задача о рюкзаке [7], (iv) построение кратчайшей надпоследовательности (надслова) [13], (v) покрытие взвешенного графа [12], (vi) минимальное покрывающее дерево [45], (vii) построение дерева Штейнера [23]. При этом обычно используется следующий сценарий (Рис. 3) [13]:

Дан некий пример (an instance) оптимизационной задачи вместе с ее оптимальным решением, необходимо найти хорошее решение для локально модифицированного примера (добавление, удаление дуг/связей).

Реоптимизационный подход направлен на улучшение полученного решения (постоптимизация). При этом обычно получаются NP-трудные модели [14]. Для некоторых упрощенных версий задач реоптимизации были предложены полиномиальные приближенные алгоритмы (PTAS) [12].

Простейший вариант одностадийной задачи реконфигурации (реструктуризации) решений иллюстрирован на Рис. 4 [33, 35]. Подход на основе реконфигурации (реструктуризации) решений подробно описан в следующем разделе.

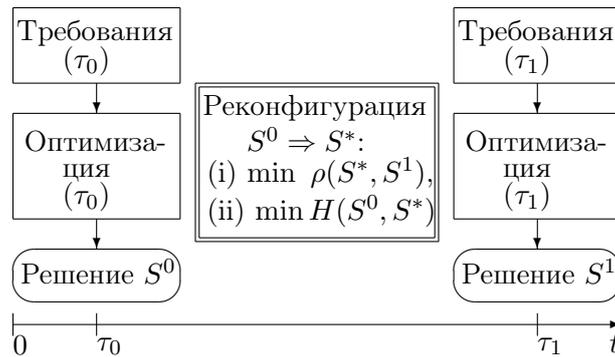


Рис. 4. Схема одностадийной реконфигурации

3. ЗАДАЧИ РЕКОНФИГУРАЦИЯ РЕШЕНИЙ

3.1. Базовая одностадийная задача реконфигурации решений

Рассмотрим задачу комбинаторной оптимизации: (1) задача P , (2) исходные данные задачи Ω (элементы, параметры элементов и др.), (3) целевая функция (или функции) $f(P)$ ($f(S)$ может быть векторной), (4) решение задачи в виде структуры S (например, подмножество элементов, структура). (5) решение задачи $S(\Omega)$ для исходных данных Ω , (6) значение целевой функции для решения $S(\Omega)$: $f(S(\Omega))$. Предполагаются два момента времени t_1, t_2 ($t_2 > t_1$). Исходные данные и решений для указанных моментов времени соответственно имеют вид: (i) $\Omega^1, f(S(\Omega^1))$, (ii) $\Omega^2, f(S(\Omega^2))$. В результате получаются решения: (a) $S^1 = S(\Omega^1)$ с $f(S(\Omega^1))$, (b) $S^2 = S(\Omega^2)$ с $f(S(\Omega^2))$. Дополнительно исследуется изменение решения (одного в другое): $H(S^\alpha \rightarrow S^\beta)$. Близость для решений S^α, S^β , рассматривается так: $\rho(S^\alpha, S^\beta) = |f(S^\alpha) - f(S^\beta)|$. Базовая версия задачи реконфигурации (реструктуризации) имеет вид:

Найти решение S^* с учетом следующего (т.е., двух критериев):

$$(i) H(S^1 \rightarrow S^*) \rightarrow \min, \quad (ii) \rho(S^*, S^2) \rightarrow \min.$$

Соответствующая оптимизационная модель с ограничением имеет вид:

$$\min \rho(S^*, S^2) \quad s.t. \quad H(S^1 \rightarrow S^*) \leq \hat{h},$$

где \hat{h} - константа (ограничение) для цены (стоимости) изменения решения.

Очевидно, имеется другой вариант оптимизационной задачи:

$$\min H(S^1 \rightarrow S^*) \quad s.t. \quad \rho(S^*, S^2) \leq \rho^0,$$

где ρ_0 - ограничение для близости цены решения.

В простейшем случае задача реконфигурации решения формулируется как задача о рюкзаке с выбором подмножества из всех возможных операций операций изменения решения [33,34]:

$$\max \sum_{i=1}^n c_i^1 x_i \quad s.t. \quad \sum_{i=1}^n a_i^1 x_i \leq b^1, \quad x_i \in \{0, 1\}.$$

При наличии связей между операциями изменения следует использовать модель комбинаторного синтеза (т.е., с учетом совместимости операций) [32,34].

3.2. Многокритериальная одностадийная задача реконфигурации

В случае многокритериальных постановок задачи реконфигурации (Рис. 5), проводится поиск Парето-оптимальных решений.

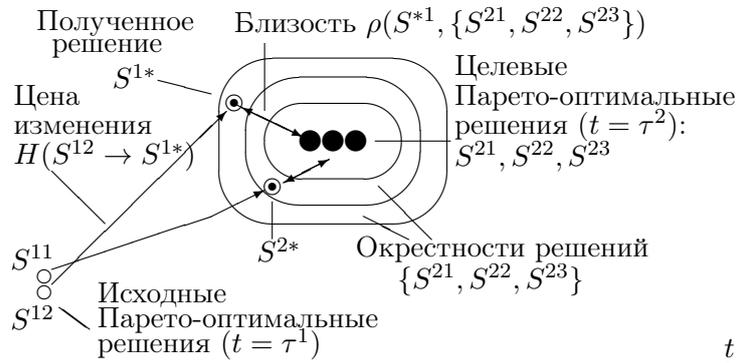


Рис. 5. Многокритериальная реконфигурация [33,35]

Кроме того, близость решений $\rho(S^*, S^2)$ (от $\rho(S^{*j}, \{S^{21}, S^{22}, S^{23}\})$) может иметь вид векторной функции (аналогично для стоимости изменения решения). Формальная постановка такой задачи имеет вид (Рис. 5, [33,35]):

Найти решение S^* с учетом двух критериев: (i) $\bar{H}(S^1 \rightarrow S^*) \rightarrow \min$, (ii) $\bar{\rho}(S^*, S^2) \rightarrow \min$.

Соответствующая оптимизационная модель с ограничением имеет вид:

$$\min \bar{\rho}(S^*, S^2) \quad st. \quad \bar{H}(S^1 \rightarrow S^*) \leq \hat{h},$$

где \hat{h} - вектор ограничение для стоимости изменения решения.

Аналогично как в базовой задаче, можно рассматривать простой вариант задачи в виде многокритериальной задачи о рюкзаке (т.е., выбор подмножества из всех возможных операций изменения решения) [33,34]:

$$\max \sum_{i=1}^n \bar{c}_i^1 x_i \quad s.t. \quad \sum_{i=1}^n \bar{a}_i^1 x_i \leq \bar{b}^1, \quad x_i \in \{0, 1\}$$

где \bar{b}^1 - вектор ограничение.

Аналогично, при наличии связей между операциями изменения следует использовать модель комбинаторного синтеза (т.е., с учетом совместимости операций) [32,34].

С другой стороны, возможно использование оценок параметров указанных моделей на основе порядковых шкал и/или шкал на основе мультимножеств [34]. В этом случае ищутся Парето-оптимальные решения в специальных частично-упорядоченных множествах (т.е., областей качества или ограничений) для \bar{H} и для $\bar{\rho}$, основанных на порядковых шкалах и/или шкалах в виде мультимножеств [34]:

Найти решение S^* с учетом следующего:

(i) $\bar{H}(S^1 \rightarrow S^*) \rightarrow \min$, (ii) $\bar{\rho}(S^*, S^2) \rightarrow \min$ (или ограничение),

где оценки для \bar{H} и $\bar{\rho}$ основаны на порядковых шкалах или шкалах в виде мультимножеств.

3.3. Многостадийная задача реконфигурации решений

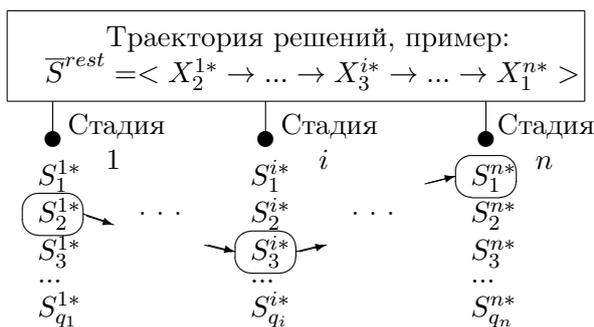
Многостадийные задача реконфигурации иллюстрированы на Рис. 6, Рис. 7 and Рис. 8. Имеются последовательные стадии и соответствующие временные моменты $t = \tau_0, \tau_1, \tau_2, \dots, \tau_n$. Рассматриваются две базовые траектории решений:

(а) n -стадийная траектория из локально оптимальных решений:

$$\bar{S}^{opt} = \langle S^0 \rightarrow S^1 \rightarrow S^2 \rightarrow \dots \rightarrow S^n \rangle,$$

(б) n -стадийная траектория из реконфигурированных (реструктуризованных) решений:

$$\bar{S}^{rest} = \langle S^0 \rightarrow S^{1*} \rightarrow S^{2*} \rightarrow \dots \rightarrow S^{n*} \rangle.$$



В результате получается модель:

Найти Парето-оптимальное решение(я) \bar{S}^{rest} с учетом следующего:

(i) $\bar{H}(\bar{S}^{rest} \rightarrow \bar{S}^{opt}) \rightarrow \min$, (ii) $\bar{\rho}(\bar{S}^{rest}, \bar{S}^{opt}) \rightarrow \min$,

где $\overline{H}(\overline{S}^{rest} \rightarrow \overline{S}^{opt}) = (H(S^0 \rightarrow S^{1*}), H(S^{1*} \rightarrow S^{2*}), \dots, H(S^{(n-1)*} \rightarrow S^{n*}))$,
 $\overline{\rho}(\overline{S}^{rest}, \overline{S}^{opt}) = (\rho(S^{1*}, S^1), \rho(S^{2*}, S^2), \dots, \rho(S^{n*}, S^n))$.

Здесь можно рассматривать две упрощенные оптимизационные модели:

- (a) $\min \overline{\rho}(\overline{S}^{rest}, \overline{S}^{opt}) \quad s.t. \quad \overline{H}(\overline{S}^{rest} \rightarrow \overline{S}^{opt}) \preceq \widehat{h}$,
 (b) $\min \overline{\rho}(\overline{S}^{rest}, \overline{S}^{opt}) \quad s.t. \quad \overline{\rho}(\overline{S}^{rest}, \overline{S}^{opt}) \preceq \widehat{\rho}$,

где $\widehat{h} = (\widehat{h}_1, \widehat{h}_2, \dots, \widehat{h}_n)$ - множество (вектор) ограничений для стоимости изменения решения (компоненты вектора соответствуют временным стадиям), $\widehat{\rho} = (\widehat{\rho}_1, \widehat{\rho}_2, \dots, \widehat{\rho}_n)$ - множество (вектор) ограничений для близости решений (компоненты вектора соответствуют временным стадиям).

Далее, следующие эвристические схемы решения могут быть рассмотрены:

Схема 1 (последовательный процесс решения):

Шаг 1. Решение оптимизационной задачи на стадии 1 (τ_1):

- (1.1.) Нахождение оптимального решения S^1 (базовая оптимизация).
 (1.2.) Нахождение реструктуризованного решения S^{1*} (т.е., $S^0 \rightarrow S^{1*}$).

Шаг 2. Решение оптимизационной задачи на стадии 2 (τ_2):

- (2.1) Нахождение оптимального решения S^2 (базовая оптимизация).
 (2.2) Нахождение реструктуризованного решения S^{2*} (т.е., $S^{1*} \rightarrow S^{2*}$).

...

Шаг n. Решение оптимизационной задачи на стадии n (τ_n):

- (n.1) Нахождение оптимального решения S^n (базовая оптимизация).
 (n.2) Нахождение реструктуризованного решения S^{n*} (т.е., $S^{(n-1)*} \rightarrow S^{n*}$).

Схема 2 (процесса решения как “композиция”):

Шаг 1. Решение оптимизационных задач:

- (1.1.) Нахождение оптимального решения S^1 (базовая оптимизация на стадии 1) (τ_1).
 (1.2.) Нахождение оптимального решения S^2 (базовая оптимизация на стадии 2) (τ_2).

...

- (1.n.) Нахождение оптимального решения S^n (базовая оптимизация на стадии n) (τ_n).

Шаг 2. Решение одностадийной задача реконфигурации для каждой стадии с целью получения нескольких “хороших” решений:

- (2.1.) Нахождение “хорошего” решения на стадии 1: $S^0 \rightarrow \{S_1^{1*}, S_2^{1*}, \dots, S_{q_1}^{1*}\}$.
 (2.2.) Нахождение “хорошего” решения на стадии 2:
 $\{S_1^{1*}, S_2^{1*}, \dots, S_{q_1}^{1*}\} \rightarrow \{S_1^{2*}, S_2^{2*}, \dots, S_{q_2}^{2*}\}$.

...

- (2.n.) Нахождение “хорошего” решения на стадии n:

$$\{S_1^{(n-1)*}, S_2^{(n-1)*}, \dots, S_{q_{n-1}}^{(n-1)*}\} \rightarrow \{S_1^{n*}, S_2^{n*}, \dots, S_{q_n}^{n*}\}.$$

Шаг 3. Композиция n-стадийной траектории из реструктуризованных решений (т.е., выбор реструктуризованного решения на каждой временной стадии) (Рис. 7) (исходная точка данной траектории соответствует S^0):

$$\overline{S}^{rest} = \langle S^0 \rightarrow S_{\xi_1}^{1*} \rightarrow S_{\xi_2}^{2*} \rightarrow \dots \rightarrow S_{\xi_n}^{n*} \rangle,$$

где $\xi_1 \in \{1, \dots, q_1\}$, $\xi_2 \in \{1, \dots, q_2\}$, ... , $\xi_n \in \{1, \dots, q_n\}$.

Схема решения 3 расширяет схему 2 за счет нахождения нескольких хороших траекторий и выбора наилучшей результирующей траектории (Рис. 7, Рис. 8):

Схема 3 (процесс решения как “композиция & выбор”):

Шаг 1. Решение оптимизационной задачи:

(1.1.) Нахождение оптимального решения S^1 (базовая оптимизация на стадии 1) (τ_1).

(1.2.) Нахождение оптимального решения S^2 (базовая оптимизация на стадии 2) (τ_2).

...

(1.n.) Нахождение оптимального решения S^n (базовая оптимизация на стадии n) (τ_n).

Шаг 2. Решение одностадийной задачи реконфигурации для каждой стадии с целью получения нескольких “хороших” решений (как в схеме 2).

Шаг 3. Композиция $k(k > 1)$ многостадийных траекторий реконфигурации решений: (выбор решения реконфигурации на каждой стадии и их композиция) (Рис. 7) (исходная точка каждой траектории соответствует S^0):

$$(3.1.) \bar{S}_1^{rest} = \langle S^0 \rightarrow S_{\xi_1^1}^{1*} \rightarrow S_{\xi_2^1}^{2*} \rightarrow \dots \rightarrow S_{\xi_n^1}^{n*} \rangle,$$

где $\xi_1^1 \in \{1, \dots, q_1\}$, $\xi_2^1 \in \{1, \dots, q_2\}$, ..., $\xi_n^1 \in \{1, \dots, q_n\}$;

$$(3.2.) \bar{S}_2^{rest} = \langle S^0 \rightarrow S_{\xi_1^2}^{1*} \rightarrow S_{\xi_2^2}^{2*} \rightarrow \dots \rightarrow S_{\xi_n^2}^{n*} \rangle,$$

где $\xi_1^2 \in \{1, \dots, q_1\}$, $\xi_2^2 \in \{1, \dots, q_2\}$, ..., $\xi_n^2 \in \{1, \dots, q_n\}$;

...

$$(3.k.) \bar{S}_k^{rest} = \langle S^0 \rightarrow S_{\xi_1^k}^{1*} \rightarrow S_{\xi_2^k}^{2*} \rightarrow \dots \rightarrow S_{\xi_n^k}^{n*} \rangle,$$

где $\xi_1^k \in \{1, \dots, q_1\}$, $\xi_2^k \in \{1, \dots, q_2\}$, ..., $\xi_n^k \in \{1, \dots, q_n\}$.

Шаг 4. Выбор наилучшей траектории реконфигурации решений \bar{S}_*^{rest} (Рис. 8):

$$\{\bar{S}_1^{rest}, \bar{S}_2^{rest}, \dots, \bar{S}_k^{rest}\} \Rightarrow \bar{S}_*^{rest}$$

3.4. Реконфигурация при изменении множества элементов

Задача реконфигурации решений рассматривается на примере задачи о рюкзаке. Имеются следующие множества элементов (Рис. 9): (i) исходное множество A_0 , (ii) новое множество A_1 , (iii) множество добавляемых элементов A^+ , (iv) множество удаляемых элементов $A^- = A_0 \setminus \{A_0 \cap A_1\}$, (v) множество фиксированных элементов (неизменяемые) $\hat{A} = \{A_0 \cap A_1\}$.

Одностадийная задача реконфигурации имеет вид (Рис. 10 [35]):

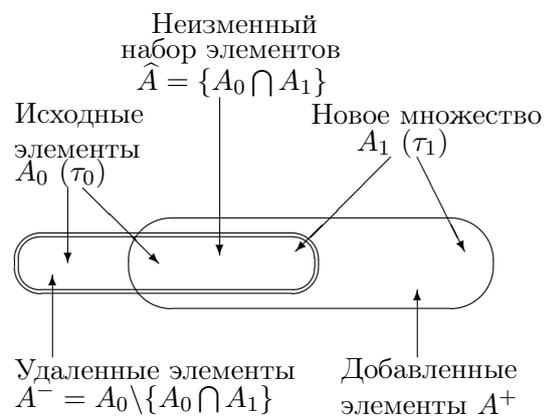


Рис. 9. Изменение состава элементов [35]

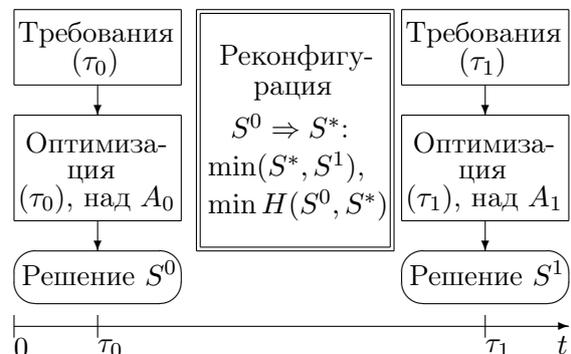


Рис. 10. Схема при изменении элементов

Найти решений S^* с учетом следующего:

(i) $H(S^1 \rightarrow S^*) \rightarrow \min$, (ii) $\rho(S^*, S^2) \rightarrow \min$,

где стоимость $\rho(S^*, S^2)$ включает следующие компоненты:

- (a) стоимость удаления элементов $A^- = A_0 \setminus \{A_0 \cap A_1\}$,
- (b) стоимость обработки фиксированных элементов $\hat{A} = \{A_0 \cap A_1\}$,
- (c) стоимость обработки новых A^+ .

Таким образом корректирующая задача (базовый вариант) решается над множеством элементов $\hat{A} \cup A^+$ с учетом стоимости удаления элементов A^- . Очевидно, эта задача может быть обобщена для многостадийного случая.

4. ПРИМЕРЫ РЕКОНФИГУРАЦИИ РЕШЕНИЙ

4.1. Задача о рюкзаке

При одностадийной реконфигурации в задаче о рюкзаке исходное множество элементов $A = \{1, \dots, i, \dots, n\}$ рассматриваются для двух моментов времени τ_0 и τ_1 (для τ_1 используются параметры $\{c_i^1\}, \{a_i^1\}, b^1$):

$$\max \sum_{i=1}^n c_i^0 x_i \quad s.t. \quad \sum_{i=1}^n a_i^0 x_i \leq b^0, \quad x_i \in \{0, 1\}.$$

Соответствующие решения имеют вид: $S^0 \subseteq A$ ($t = \tau_0$) и $S^1 \subseteq A$ ($t = \tau_1$) ($S^0 \neq S^1$).

Иллюстративный численный пример:

Имеется множество элементов и решения $A = \{1, 2, 3, 4, 5, 6, 7\}$, $S^0 = \{1, 3, 4, 5\}$, $S^1 = \{2, 3, 5, 7\}$, $S^* = \{2, 3, 4, 6\}$. Реконфигурация $S^0 \Rightarrow S^*$ основывается на следующем (Рис. 11): (a) удаление элементов: $S^{*-} = S^0 \setminus S^* = \{1, 5\}$, (b) добавление элементов: $S^{*+} = S^* \setminus S^0 = \{2, 6\}$.

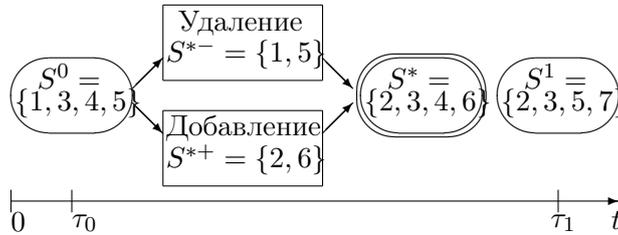


Рис. 11. Пример реконфигурации [33, 35]

В начале имеется следующее: $S^{*-} = S^0$ и $S^{*+} = A \setminus S^0$. Модель реконфигурации можно рассмотреть так:

$$\min \rho(S^*, S^1) \quad s.t. \quad H(S^0 \Rightarrow S^*) = \left(\sum_{i \in S^{*-}} h_i^- + \sum_{i \in S^{*+}} h_i^+ \right) \leq \hat{h}, \quad \sum_{i \in S^*} a_i^2 \leq b^2,$$

где \hat{h} - константа для ограничения стоимости изменения, $h^-(i)$ - стоимость удаления элемента $i \in A$, $h^+(i)$ - стоимость добавления элемента $i \in A$. можно рассматривать эквивалентную модель:

$$\max \sum_{i \in S^*} x_i c_i^1 \quad s.t. \quad H(S^0 \Rightarrow S^*) = \left(\sum_{i \in S^{*-}} h_i^- + \sum_{i \in S^{*+}} h_i^+ \right) \leq \hat{h}, \quad \sum_{i \in S^*} a_i^1 \leq b^1,$$

поскольку $\max \sum_{i \in S^*} x_i c_i^1 \leq \max \sum_{i \in S^2} x_i c_i^1$ с учетом ограничения $\sum_{i \in S^*} a_i^1 \leq b^1$. Очевидно, модифицированная модель является также задачей о рюкзаке.

С другой стороны можно использовать упрощенный процесс решения на основе анализа изменяемых элементов (т.е., кандидатов на добавление/удаление): (а) генерация кандидатов на удаление (выбор S^{*-} из S^0), (b) генерация кандидатов на добавление (выбор S^{*+} из $A \setminus S^0$). Указанный процесс выбора может основываться на многокритериальном ранжировании.. В результате получается задача с уменьшенной размерностью.

В многокритериальном случае получается практически такая же постановка, только соответствующая рюкзачная модель является многокритериальной.. Аналогичная ситуация имеет место и при использовании порядковых шкал и/или шкал в виде мультимножеств [34].

Прикладной пример для трехстадийной реконфигурации модульного учебного курса:

Имеется набор учебных тем/модулей $A = \{1, \dots, i, \dots, 13\}$ и их характеристик (Таблица 3) и три временные стадии ($t \in \{\tau_0, \tau_1, \tau_2\}$) [35]. Для каждого параметра элемента i рассматриваются значения на трех стадиях (γ - номер стадии $\gamma = 0, 1, 2$): (а) полезность c_i^γ , (b) требуемый ресурс a_i^γ , (c) стоимость удаления $cost\ i\ h_i^{\gamma-}$, (d) стоимость добавления $h_i^{\gamma+}$.

Во-первых, рассматривается задача о рюкзаке для каждой стадии ($\gamma = 0, 1, 2$) (при следующих общих ресурсных ограничениях $b^0 = 14, b^1 = 20, b^2 = 23$):

$$\max \sum_{i=1}^{13} c_i^\gamma x_i^\gamma \quad s.t. \quad \sum_{i=1}^{13} a_i^\gamma \leq b^\gamma.$$

Получаются следующие решения:

$$S^0 = \{1, 2, 4, 8, 11, 12, 13\}, c(S^0) = 22.0, b(S^0) = 13.8;$$

$$S^1 = \{1, 2, 4, 8, 10, 11\}, c(S^1) = 31.0, b(S^1) = 19.7;$$

$$S^2 = \{1, 2, 3, 4, 7, 8, 11\}, c(S^2) = 29.5, b(S^2) = 22.5.$$

Предполагается следующее: элементы $B = \{1, 2, 4, 8, 11\}$ включены в решение на каждой стадии. Таким образом следующее множество элементов рассматривается с точки зрения возможных изменений $\hat{A} = \{3, 5, 6, 7, 9, 10, 12, 13\}$.

Таблица 2. Темы (модули) курса по комбинаторной оптимизации, их параметры

i	Тема (модуль)	$\tau_0 : c_i^0$	a_i^0	$\tau_1 : c_i^1$	a_i^1	h_i^{1-}	h_i^{1+}	$\tau_2 : c_i^2$	a_i^2	h_i^{2-}	h_i^{2+}
1.	Сложность моделей, алгоритмов	4.0	1.5	5.0	2.0	0.5	1.0	5.0	2.0	0.5	1.0
2.	Рюкзак	4.0	3.0	5.0	4.0	0.6	1.5	5.0	4.0	0.6	1.5
3.	Маршрутизация	1.0	3.5	5.0	4.0	0.8	1.4	6.0	5.0	0.8	1.4
4.	Назначение/размещения	4.0	2.5	5.0	3.0	0.7	1.5	5.0	3.0	0.7	1.5
5.	Расписания	1.5	5.0	2.0	4.0	0.4	1.5	3.0	5.0	0.4	1.5
6.	Упаковка	1.0	3.0	3.0	3.0	0.3	1.0	3.5	4.0	0.3	1.0
7.	Покрытия	1.0	1.5	2.5	2.0	0.4	1.0	3.0	3.5	0.4	1.0
8.	Покрывающие деревья	3.0	2.0	4.0	2.5	1.0	1.5	5.0	3.5	1.0	1.5
9.	Клики	1.0	1.5	1.5	2.0	0.3	0.8	2.0	2.0	0.3	0.8
10.	Раскраска графов	1.0	1.5	2.0	1.7	0.2	0.7	2.5	2.0	0.2	0.7
11.	Кластеризация, сортировка	4.0	2.0	5.0	2.5	1.0	1.7	5.5	2.5	1.0	1.7
12.	Выравнивание	1.0	0.8	0.9	1.0	0.2	0.3	0.9	1.0	0.2	0.3
13.	Выполнимость	2.0	2.0	1.5	2.0	0.2	0.3	1.0	2.0	0.2	0.3

Далее рассматриваются последовательно две задачи реконфигурации (как задача о рюкзаке на основе удаления/добавления элементов) (по схеме 1): $S^0 \rightarrow S^{1*}$ и $S^{1*} \rightarrow S^{2*}$. Локальная задача реконфигурации для τ_1 имеет вид ($\rho(S^{1*}, S^1) = |c(S^{1*}) - c(S^1)|$, $D^1 = 1.6$ - ограничение для общей стоимости изменения):

$$\min \rho(S^{1*}, S^1) \quad s.t. \quad \left(\sum_{i \in (\widehat{A} \cap S^0)} h_i^{1-} + \sum_{i \in (\widehat{A} \setminus (\widehat{A} \cap S^0))} h_i^{1+} \right) \leq D^1, \quad \sum_{i \in (B \cup (\widehat{A} \setminus (\widehat{A} \cap S^0)))} a_i \leq b^1.$$

Получаются два решения реконфигурации для исследования (задача для τ_2 рассматривается аналогично, $D^2 = 1.6$):

$$S^{1*} = \{1, 2, 3, 4, 8, 11\}, \quad c(S^{1*}) = 29.0, \quad b(S^{1*}) = 19.0;$$

$$S^{2*} = \{1, 2, 3, 4, 8, 11\}, \quad c(S^{2*}) = 29.5, \quad b(S^{2*}) = 22.5 \quad (\text{здесь } S^{2*} = S^2).$$

Результирующая траектория имеет вид: $\overline{S}^{rest} = \langle S^0, S^{1*}, S^{2*} \rangle$.

4.2. Задача блочного рюкзака

Рассматривается одностадийная реконфигурация для задачи блочного рюкзака, имеется два момента времени $t = \{\tau_1, \tau_2\}$. Рассматривается задача о рюкзаке для $t = \tau_1$ (для $t = \tau_2$, используются параметры $\{c_{ij}^2\}$, $\{a_{ij}^2\}$ и b^2):

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

Исходное множество элементов A разбивается на m подмножеств (без пересечений): $A = \bigcup_{i=1}^m A_i$, где $A_i = \{1, \dots, j, \dots, q_i\}$ ($i = \overline{1, m}$). Каждый элемент обозначается (i, j) . Эквивалентная задача имеет вид:

$$\max \sum_{(i,j) \in S^1} c_{ij}^1 \quad s.t. \quad \sum_{(i,j) \in S^1} a_{ij}^1 \leq b^1, \quad |S^1 \cap A_i| \leq 1 \quad \forall i = \overline{1, m}.$$

Для $t = \tau_2$ рассматривается аналогичная модель.

Иллюстративный численный пример:

Имеется множество элементов $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$, $A_1 = \{1, 3, 5, 12\}$, $A_2 = \{2, 7, 9\}$, $A_3 = \{4, 8, 13\}$, $A_4 = \{6, 10, 11\}$, $S^1 = \{1, 7, 8, 11\}$, $S^2 = \{3, 7, 8, 10\}$, $S^* = \{1, 2, 8, 6\}$. Процесс реконфигурации (т.е., $S^1 \Rightarrow S^*$) основан на следующем: (а) удаление элементов: $S^{1*-} = S^1 \setminus S^* = \{7, 11\}$, (б) добавление элементов: $S^{1*+} = S^* \setminus S^1 = \{2, 6\}$.

Далее рассматривается задача реконфигурации:

$$\min \rho(S^*, S^2)$$

$$s.t. \quad H(S^1 \Rightarrow S^*) = \left(\sum_{(i,j) \in S^{1*-}} h_{ij}^- + \sum_{(i,j) \in S^{1*+}} h_{ij}^+ \right) \leq \widehat{h}, \quad \sum_{(i,j) \in S^*} a_{ij}^2 \leq b^2, \quad |S^* \cap A_i| \leq 1 \quad \forall i = \overline{1, m},$$

где \widehat{h} - константа для стоимости изменения, $h^-(ij)$ - стоимость удаления элемента $(i, j) \in A$, $h^+(ij)$ - стоимость добавления элемента $(i, j) \in A$. Эквивалентная модель имеет вид:

$$\max \sum_{(i,j) \in S^*} c_{ij}^2$$

$$s.t. \quad H(S^1 \Rightarrow S^*) = \left(\sum_{(i,j) \in S^{1*-}} h_{ij}^- + \sum_{(i,j) \in S^{1*+}} h_{ij}^+ \right) \leq \widehat{h}, \quad \sum_{(i,j) \in S^*} a_{ij}^2 \leq b^2, \quad |S^* \cap A_i| \leq 12 \quad \forall i = \overline{1, m}.$$

Это блочная задача о рюкзаке. В случае многокритериальной блочной задачи о рюкзаке процесс реконфигурации является аналогичным. Аналогичная ситуация имеет место для случая использования порядковых оценки/или оценок на основе мультимножеств [34].

Прикладной пример для электронного устройства:

Рассматривается компонент электронного устройства как часть беспроводного сенсора (использованы данные из [38]). Используется упрощенная двух-уровневая структура:

0. Структура $M = R * P * D * Q$:

1. Радио часть R : 10 mw 916 MHz Radio $R_1(3)$, 1 mw 916 MHz Radio $R_2(2)$, 10 mw 600 MHz Radio $R_3(2)$, 1 mw 600 MHz Radio $R_4(1)$.

2. Микропроцессор P : MAXQ 2000 $P_1(1)$, AVR with embedded DAC/ ADC $P_2(2)$, MSP $P_3(3)$.

3. DAC/ADC D : Motorola $D_1(2)$, AVR embedded DAC/ADC $D_2(1)$, Analog Devices 1407 $D_3(2)$.

4. Память Q : 512 byte RAM $Q_1(3)$, 512 byte EEPROM $Q_2(3)$, 8 KByte Flash $Q_3(2)$, 1 MByte Flash $Q_4(1)$.

Здесь предполагается, что построение системы основано на блочной задаче о рюкзаке (в [38] композиция системы основана на модели морфологической клики с учетом совместимости выбранных компонентов системы). Предполагаются два решения:

(1) решения для стадии 1: M^1 ($t = \tau_1$, Рис. 12),

(2) решения для стадии 2: M^2 ($t = \tau_2$, Рис. 13).

Таблица 3 содержит оценки для проектных вариантов DAs (экспертное оценивание). Оценки стоимости вариантов (Таблица 3) и приоритеты (Рис. 12, Рис. 13, в круглых скобках) соответствуют примерам в [38].

Таблица 3. Оценки DAs

DAs	Стоимость (вес) (a_{ij})	Цена изменения : h_{ij}^- h_{ij}^+	Приоритеты: c_{ij}^1 c_{ij}^2
R_1	6	2 2	1 1
R_2	5	1 1	2 3
R_3	3	2 1	2 1
R_4	2	2 2	3 2
P_1	5	2 3	3 2
P_2	10	2 2	2 3
P_3	30	3 2	1 2
D_1	2	2 3	2 3
D_2	1	2 2	3 2
D_3	2	1 1	2 1
Q_1	3	2 1	1 3
Q_2	2	2 2	1 3
Q_3	3	1 2	2 2
Q_4	3	1 1	3 2

Две возможных операции изменения могут быть рассмотрены ($M^1 \Rightarrow M^*$, M^* - близко к M^2):

(а) $R_4 \rightarrow R_2$, $h_a^- = 2$, $h_a^+ = 1$ (соответствующие Булевы переменные: $x_a \in \{0, 1\}$),

(б) $Q_4 \rightarrow Q_1, h_b^- = 1, h_b^+ = 1$ (соответствующие Булевы переменные: $x_b \in \{0, 1\}$).

В результате следующая упрощенная задача о рюкзаке может использоваться:

$$\max ((c^2(R_2) - c^2(R_4)) x_a + (c^2(Q_1) - c^2(Q_4)) x_b)$$

$$s.t. H(M^* \rightarrow M^2) = (h^-(R_4 \rightarrow R_2) + h^+(R_4 \rightarrow R_2)) x_a + (h^-(Q_4 \rightarrow Q_1) + h^+(Q_4 \rightarrow Q_1)) x_b \leq \hat{h}.$$

Результирующие решения реконфигурации имеют вид: (i) $\hat{h} = 2: M^{*1} = R_4 \star P_2 \star D_2 \star Q_1$,
 (ii) $\hat{h} = 3: M^{*2} = R_2 \star P_2 \star D_2 \star Q_4$, (iii) $\hat{h} = 5: M^{*3} = M^2 = R_2 \star P_2 \star D_2 \star Q_1$.

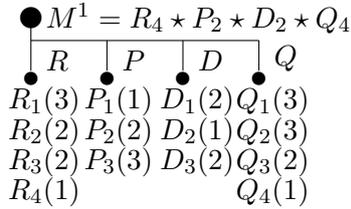


Рис. 12. Структура M^1

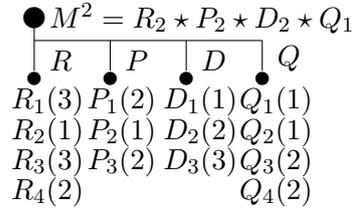


Рис. 13. Структура M^2

5. ЗАКЛЮЧЕНИЕ

Данная статья содержит описание подхода к реконфигурации (реструктуризации) решений в задачах комбинаторной оптимизации. Предлагается краткий обзор подходов к модификации задач/решения в оптимизации. Описываются особенности предлагаемого подхода и три его варианта: (1) одностадийная рекофигурация, (2) многостадийная рекофигурация, (3) реконфигурация при изменении множества элементов исходной задачи. Задачи реконфигурации рассматриваются с одним или многими критериями. В основном, данные задачи сводятся к добавлению элементов и удалению элементов и предлагаются комбинаторные модели рюкзачного типа. В целом, задачи реконфигурации являются достаточно сложными (особенно в многостадийном случае) и требуется применения составных много-компонентных и многоэтапных эвристических схем решения. В работе описаны некоторые простые составные эвристические схемы решения. Предложенный подход применен к задаче о рюкзаке и задаче блочного рюкзака, включая иллюстративные численные и прикладные примеры.

Следует отметить, что предложенный подход реконфигурации решений соответствует многим прикладным динамическим задачам (например, системы мобильной связи, многоканальные системы связи, распределенные вычислительные системы) и представляется актуальным для приложений.

В дальнейшем, важно применить предложенный подход к другим задачам комбинаторной оптимизации. Кроме того, следует провести исследование применения данного подхода с учетом неопределенности исходных данных и использования различных моделей для близости решений и оценки стоимости изменения решений.

СПИСОК ЛИТЕРАТУРЫ

1. Adibi A.M., Zandieh M., Amiri M., Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Exp. Syst. Appl.*, 2010, vol. 37, no. 1, pp. 282–287.
2. Ahuja R.K., Orlin J.B., Inverse optimization. *Oper. Res.*, 2001, vol. 49, no. 5, pp. 771–783.
3. Aiex R.M., Resende M.G.C., Pardalos P.M., Toraldo G., GRASP with path relinking for three-index assignment. *INFORMS J. on Comput.*, 2005, vol. 17, no. 2, pp. 224–247.
4. Albers S., Online algorithms: a survey. *Math. Program. Ser. B*, 2003, vol. 97, no. 1–2, pp. 3–26.

5. Arain F.M., IT-based approach for effective management of project changes: A change management system (CMS). *Adv. Eng. Inform.*, 2008, vol. 22, no. 4, pp. 457–472.
6. Archetti C., Bertazzi L., Speranza M.G., Reoptimizing the traveling salesman problem. *Networks*, 2003, vol. 42, no. 3, pp. 154–159.
7. Archetti C., Bertazzi L., Speranza M.G., Reoptimizing the 0-1 knapsack problem. Technical Report 267, Univ. of Brescia, 2006.
8. Austello G., Escoffer B., Monnot J., Paschos V., Reoptimization of minimum and maximum traveling salesman's tours. *J. of Disc. Alg.*, 2009, vol. 7, no. 4, pp. 453–463.
9. Berman O., Repositioning of distinguishable urban service units on networks. *Comp. Oper. Res.*, 1981, vol. 8, no. 2, pp. 105–118.
10. Bi Z.M., Lang S., Shen W., Wang L., Reconfigurable manufacturing systems: the state of the art. *Int. J. of Prod. Res.*, 2008, vol. 46, no. 4, pp. 967–992.
11. Bilo D., Bockenhauer H.-J., Hromkovic J., Kralovic R., Momke T., Widmayer P., Zych A., Reoptimization of Steiner trees. In: Gudmundsson J. (ed), *Proc. of Scandinavian Workshop on Alg. Theory SWAT'08*, LNCS 5124, Springer, 2008, pp. 258–269.
12. Bilo D., Widmayer P., Zych A., Reoptimization of weighted graph and covering problems. In: Bampis E., Skutella M. (eds), *Proc. of 6th Int. Workshop on Approx. and Online Alg. WAOA'08*, LNCS 5426, Springer, 2009, pp. 201–213.
13. Bilo D., Bockenhauer H.-J., Komm D., Kralovic R., Momke T., Seibert S., Zych A., Reoptimization of the shortest common superstring problem. *Algorithmica*, 2011, vol. 61, no. 2, pp. 227–251.
14. Bockenhauer H.-J., Hromkovic J., Momke T., Widmayer P., On the hardness of reoptimization. In: *34th Int. Conf. on Current Trends in Theory and Practice of Comp. Sci. SOFSEM*, 2008, pp. 50–65.
15. Bondalapati K., Prasana V.K., Reconfigurable computing systems. *Proc. of the IEEE*, 2002, vol. 90, no. 7, pp. 1201–1217.
16. Boria N., Paschos V.P., Fast reoptimization for the minimum spanning tree problem. *J. of Disc. Alg.*, 2010, vol. 8, no. 3, pp. 296–310.
17. Burkard R.E., Gassner E., Hatzl J., Reverse 2-median problem on trees. *Disc. Appl. Optim.*, 2008, vol. 156, no. 11, pp. 1963–1976.
18. Burkard R.E., Galavi M., Gassner E., The inverse Fermat-Weber problem. *Eur. J. Oper. Res.*, 2010, vol. 206, no. 1, pp. 11–17.
19. Burzyn P., Bonomo F., Duran G., NP-completeness results for edge modification problems. *Disc. Appl. Math.*, 2006, vol. 154, no. 13, pp. 1824–1844.
20. Coello Coello C., Lamont G.B., Van Veldhuizen D.A., *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd ed., Springer, 2007.
21. Deb K., *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2009.
22. Duin C.W., Volgenant A., Some inverse optimization problems under the Hamming distance. *Eur. J. Oper. Res.*, 2006, vol. 170, no. 3, pp. 887–899.
23. Escoffier B., Milanic M., Paschos V.T., Simple and fast reoptimizations for the Steiner tree problem. *Algor. Oper. Res.*, 2009, vol. 4, no. 2, pp. 86–94.
24. Eswaran K.P., Tarjan R.E., Augmentation problems. *SIAM J. on Comput.*, 1976, vol 5, no. 4, pp. 653–665.
25. Exposito-Izquierdo C., Lalla-Ruiz E., Melian-Batista B., Moreno-Vega J.M., A study of rescheduling strategies for the quay crane scheduling problem under random disruptions. *Inteligencia Artificial*, 2014, vol. 17, no. 54, pp. 35–47.
26. Garey M.R., Johnson D.S., *Computers and Intractability. The Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, 1979.

27. Gunasekaran C., Mehrotra K., Mohan C.K., Multi-objective restructuring in social networks. In: Proc. of The 2013 IEEE/ACM Int. Conf. on Adv. in Soc. Netw. Anal. and Mining (ASONAM 2013), ACM, 2013, pp. 277–281.
28. Van Hentenryck P., Bent R., Online Stochastic Combinatorial Optimization. The MIT Press, 2009.
29. Heuberger C., Inverse optimization: A survey on problems, methods, and results. J. of Comb. Optim., 2004, vol. 8, no. 3, pp. 329–361.
30. Ho S.C., Szeto W.Y., Solving a static repositioning problem in bike-sharing systems using iterated tabu search. Transp. Res. Part E, 2014, vol. 69, pp. 180–198.
31. Khouadjia M.R., Sarasola B., Alba E., Jourdan L., Talbi E.G., A comparative study between dynamic adapted pso and vns for the vehicle routing problem with dynamic requests. Appl. Soft Comput., 2012, vol. 12, no. 4, pp. 1426–1439.
32. Levin M.Sh., Composite Systems Decisions. Springer, New York, 2006.
33. Levin M.Sh., Restructuring in combinatorial optimization. Electronic preprint. 11 pp., Febr. 8, 2011. <http://arxiv.org/abs/1102.1745> [cs.DS]
34. Levin M.Sh., Modular System Design and Evaluation, Springer, 2015.
35. Levin M.Sh., Towards integrated glance to restructuring in combinatorial optimization. Electronic preprint. 31 p., Dec. 20, 2015. <http://arxiv.org/abs/1512.06427> [cs.AI]
36. Левин М.Ш., О динамической комбинаторной кластеризации. Информационные процессы, 2016, том 16, ном. 2, стр. 177–193.
37. Levin M.Sh., Data allocation on disks with solution reconfiguration (problems, heuristics). Elec. prepr. 10 p., Dec. 14, 2016. <http://arxiv.org/abs/1612.04519> [cs.DC]
38. Levin M.Sh., Fimin A.V., Configuration of alarm wireless sensor element. In: Proc. of 2nd Int. Conf. on Ultra Modern Telecommunication ICUMT-2010, Moscow, 2010, pp. 924–928.
39. Manna B., Cluster editing problem for points on the real line: A polynomial time algorithm. Inform. Proc. Lett., 2010, vol. 110, no. 21, pp. 961–965.
40. Moran S., Snir S., Convex recoloring of strings and trees: definitions, hardness results and algorithms. J. Comput. Syst. Sci., 2008, vol. 74, no. 5, pp. 850–869.
41. Natanzon A., Shamir R., Sharan R., Complexity classification of some edge modification problems. Disc. Appl. Math., 2001, vol. 113, no. 1, pp. 1009–128.
42. Noltmeier H., Wirth H.-C., Krumke S.O., Network design and improvement. ACM Comput. Surv., 1999, vol. 32(3es), Art. no. 2.
43. Pardalos P.M., Aydogan E.K., Gurbuz F., Demirtas O., Bakiri B.B., Fuzzy combinatorial optimization problems. In: Pardalos P.M., Du D.-Z., Graham R.L. (eds), Handbook of Combinatorial Optimization, Springer, 2013, pp. 1357–1413.
44. Pollock L.L., Soffa M.L., Incremental global reoptimization of programs. ACM Trans. on Progr. Lang. Syst. 1992, vol. 14, no. 2, pp. 173–200.
45. Qiu J., Liu Y., Mohan G., Chua K.C., Fast spanning tree reconnection mechanism for resilient Metro Ethernet networks. Comp. Netw., 2011, vol. 55, pp. 2717–2729.
46. Schaffter M.W., Scheduling with forbidden sets. Disc. Appl. Math., 1997, vol. 72, no. 12, pp. 155–166.
47. Shieber B., Shachnai H., Tamir G., Tamor T., A theory and algorithms for combinatorial reoptimization. In: Fernandez-Baca D. (ed), Proc. of 10th Latin American Symp. LATIN 2012: Theoretical Informatics, LNCS 7256, Springer, 2012, pp. 618–630.
48. Sritharan R., Graph modification problem for some classes of graphs. J. Discr. Alg., 2016, vol. 38–41, pp. 32–37.

49. T'Kindt V., Billaut J.C., *Multicriteria Scheduling: Theory, Models and Algorithms*. 2nd ed., Springer, 2006.
50. Watanabe T., Nakamura A., A minimum 3-connectivity augmentation of graph. *J. Comput. Syst. Sci.*, 1993, vol 46, no. 1, pp. 91–128.
51. Wzorek M., Doherty P., Reconfigurable path planning for an autonomous unmanned aerial vehicle. In: *Int. Conf. on Hybrid Information Technology ICHIT'06*, 2006, vol. 2, pp. 242–249.
52. Yang C., Zhang J., Two general methods for inverse optimization problems. *Appl. Math. Lett.*, 1999, vol. 12, no. 2, pp. 69–72.
53. Yang S., Ong Y.-S., Jin Y. (eds), *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer, Berlin, 2007.
54. Yang S., Jiang Y., Nguyen T.T., Metaheuristics for dynamic combinatorial optimization problems. *IMA J. of Manag. Math.*, 2013, vol. 24, pp. 451–480.
55. Yannakakis M., Node-deletion problems on bipartite graphs. *SIAM J. Comput.*, 1981, vol. 10, pp. 310–327.
56. Yemets O.A., Roskladka A.A., Combinatorial optimization under uncertainty. *Cybern. Syst. Anal.*, 2008, vol. 44, no. 5, pp. 655–663.
57. Zhang J., Liu Z., Ma Z., Some reverse location problems. *Eur. J. Oper. Res.*, 2000, vol. 124, pp. 77–88.
58. Zhang J., Lin Y., Computation of the reverse shortest-path problem. *J. Glob. Optim.*, 2003, vol. 25, pp. 243–261.
59. Zhu W., Qin H., Lim A., Zhang H., Iterative deeping A' algorithm for the container relocation problem. *IEEE Trans. Autom. Sci. Eng.*, 2012, vol. 9, pp. 710–722.

On solution reconfiguration in combinatorial optimization

Levin M.Sh.

The paper focuses on a new class of combinatorial problems which consists in reconfiguration (restructuring) of solutions (as sets/structures) in combinatorial optimization. Two main features of the restructuring process are examined: (i) a cost of the solution reconfiguration, (ii) a closeness to a goal solution. Three types of the restructuring problems are under study: (a) one-stage restructuring, (b) multi-stage restructuring, and (c) restructuring over changed element set. One-criterion and multicriteria problem formulations can be considered. The solution reconfiguration problems correspond to redesign (improvement, upgrade) of modular systems or solutions. The approach is described and illustrated (problem statements, solving schemes, examples) for two combinatorial optimization problems: knapsack problem, multiple choice problem. Numerical examples illustrate the restructuring problems and solving schemes.

KEYWORDS: Combinatorial optimization, reconfiguration, restructuring, knapsack problem, multiple choice problem, heuristics