

Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет
“Высшая школа экономики””

На правах рукописи

Бычков Илья Сергеевич

**МОДЕЛИ И АЛГОРИТМЫ ДЛЯ ЗАДАЧИ О
ФОРМИРОВАНИИ ПРОИЗВОДСТВЕННЫХ
ЯЧЕЕК**

РЕЗЮМЕ

диссертации на соискание ученой степени кандидата
компьютерных наук

Нижний Новгород – 2020

Диссертационная работа выполнена в Лаборатории алгоритмов и технологий анализа сетевых структур (ЛАТАС), Национальный исследовательский университет Высшая школа экономики

Научный руководитель:

Бацын Михаил Владимирович, к.ф.-м.н., ведущий научный сотрудник Лаборатории алгоритмов и технологий анализа сетевых структур

1 Введение

Проектирование плана производства (manufacturing layout) является одним из наиболее важных этапов в процессе построения новой производственной системы или адаптации существующей к изменяющимся условиям (выделение новых ресурсов, изменение объемов или продукции и т.д.). Под планом производства в данной работе понимается эффективное расположение имеющихся ресурсов производства (станки) с целью получения различного рода преимуществ. Эти преимущества включают упрощение планирования, снижение затрат на перемещение материалов, высокую загрузку оборудования и многие другие.

В научных статьях по данной тематике рассматриваются два варианта компоновки производства: статическая и динамическая [28]. Одним из самых популярных реализаций статической компоновки является разбиение всего производства на отдельные цеха. Идея разбиения на цеха является частью концепции групповой технологии (grouping technology - GT), которая была введена российским учёным в области технологии и организации производства, Сергеем Петровичем Митрофановым [47, 48] и популяризована Джоном Барбиджем [10, 11, 12, 13]. Групповая технология - это стратегия, которая помогает оптимизировать производственный процесс с точки зрения обработки материалов и управления системой серийного производства. Ключевая идея GT заключается в том, что производимые детали, которые похожи с точки зрения производственных характеристик, должны обрабатываться в рамках одной единицы, называемой производственной ячейкой. Эта идея приводит к разделению производственной системы на подсистемы путем определения семейств деталей и групп станков. Каждая производственная ячейка связана с определенным физическим местоположением, машинами, настроенными и ответственными за обработку определенной группы деталей. Задача получения такого рода разбиений называется задачей формирования производственных ячеек (cell formation problem - CFP). Целью проектирования системы из нескольких производственных ячеек является снижение производственных затрат за счет максимизации загрузки станков внутри ячеек и минимизации перемещения деталей из одной ячейки в другую. Эффективно спланированные производственные ячейки обеспечивают снижение затрат на установку и перенос оборудования, а также экономию производственных площадей.

В настоящей диссертации рассматривается задача о разбиении производства на некоторое произвольное число ячеек. В самой общей постановке задача CFP может быть рассмотрена как задача бикластеризации. Задачи бикластеризации впервые были описаны Хартиганом [34, 35]. Термин бикластеризация был введен Миркиным [45]. Позже Миркин также определил понятие приближенных бикластеров (вместе с трикластерами и р-кластерами) и предложил два алгоритма бикластеризации на двоичных данных [46]. [18] рассматривает наиболее широко используемые методы и приложения бикластеризации. [49] и [19] предложены модели и методы решения задачи выбора признаков (feature selection) с использованием последовательной бикластеризации.

Исследования по сложности задачи CFP практически не встречаются в научных статьях. По настоящее время формально сложностной статус задачи неопределен, однако многие авторы склонны считать задачу о формировании производ-

ственных ячеек NP-трудной [44, 32, 20, 57, 29].

Многие исследования, в том числе James et al. [36]; Chung et al. [25]; Paydar & Saidi-Mehrabad [52]; Solimanpur et al. [55]; Utkina et al. [58] при обсуждении NP-трудности CFP ссылаются на работу Ballakur & Steudel [4], в которой авторами представляется эвристический алгоритм для различных целевых функций задачи, однако ее сложностной статус явно не указывается. Самая популярная и важная в данной области целевая функция, которая называется эффективностью группировки (grouping efficacy) была представлена позже [42]. Эффективность группировки в настоящее время широко признана исследователями в рассматриваемой области и успешно отражает решение задач минимизации перемещения деталей между производственными ячейками и максимизации загрузки оборудования.

В данной диссертации приводятся строгие доказательства NP-полноты для различных формулировок задачи CFP: с дробно-линейной функцией эффективности группировки и линейной функцией минимизирующей суммарное перемещение деталей и неполноту загрузки машин ($E + V$).

Объект исследования: задача CFP, задаваемая числом имеющихся станков для производства и производимых деталей, а также самим производственным планом. Число производственных ячеек заранее неопределено и может различаться в зависимости от текущей задачи и используемой целевой функции.

Цель кандидатской диссертации: разработка эффективных методов решения задачи CFP по сравнению с современными подходами и нахождение строгих доказательств NP-полноты для различных формулировок задачи CFP.

Научная новизна результатов кандидатской диссертации заключается в следующем:

- доказательство NP-полноты для варианта решения задачи CFP с дробно-линейной целевой функцией эффективности группировки и линейной функцией $E + V$
- два оригинальных точных подхода для решения задачи CFP с дробно-линейной целевой функцией
- эффективный эвристический алгоритм поиска во многих окрестностях для решения задачи CFP

Предложенные алгоритмы и подходы имеют широкий спектр практических применений. Первое - это разработка оптимального или близкого к оптимальному плана размещения оборудования, позволяющего повысить производительность систем серийного производства. Кроме того, формулировка, рассмотренная в этом тезисе, может также использоваться для решения различных проблем биокластеризации (например, задачи экспрессии генов в биологии - gene expression), а также задач кластеризации на двудольных графах (bicluster graph editing problem).

Вклад автора включает разработку и реализацию подходов и алгоритмов, доказательства теорем, лемм и утверждений, тестирование алгоритмов, проведение вычислительных экспериментов и подготовку текстов научных работ.

Автор диссертации имеет свидетельство об официальной регистрации программы для ЭВМ №2014610434 - "Эвристический алгоритм решения задачи о формировании производственных ячеек".

Результаты данной работы опубликованы в 6 научных статьях в международных рецензируемых журналах и материалах конференций.

Публикации первого уровня:

1. Bychkov, I., Batsyn, M. (2018) An efficient exact model for the cell formation problem with a variable number of production cells. *Computers & Operations Research*, 91, 112-120, Q1 (Автор диссертации предложил и реализовал точный подход для решения CFP, провел вычислительные эксперименты и подготовил текст статьи)
2. Bychkov, I., Batsyn, M., Pardalos, P. M. (2014). Exact model for the cell formation problem. *Optimization Letters*, 8(8), 2203-2210, Q2 (Автор диссертации предложил и реализовал точный подход для решения CFP, провел вычислительные эксперименты, доказал приводимые в исследовании теоремы и утверждения и подготовил текст статьи)
3. Batsyn, M., Batsyna E., Bychkov I. (2019) NP-completeness of cell formation problem with grouping efficacy objective, *International Journal of Production Research* (Q1), Published online: 26 Sep 2019, <https://doi.org/10.1080/00207543.2019.1668072> (Автор диссертации принимал участие в обсуждении и доказательстве всех теорем и утверждений)
4. Bychkov, I., Batsyn, M., Sukhov, P., Pardalos, P.M (2013) Heuristic Algorithm for the Cell Formation Problem. In: Goldengorin et al (eds) *Models, Algorithms, and Technologies for Network Analysis*. Springer Proceedings in Mathematics & Statistics 59, 43–69. (Автор диссертации предложил и реализовал эвристический подход для решения CFP, провел вычислительные эксперименты и подготовил текст статьи)

Публикации второго уровня:

5. Bychkov I., Batsyn M., Pardalos P.M. (2017) Heuristic for Maximizing Grouping Efficiency in the Cell Formation Problem. In: Kalyagin et al (eds) *Models, Algorithms, and Technologies for Network Analysis*. Springer Proceedings in Mathematics & Statistics 197, 11–26. (Автор диссертации предложил и реализовал эвристический алгоритм для решения CFP, провел вычислительные эксперименты и подготовил текст статьи)
6. Batsyn, M., Bychkov I., Goldengorin B., Pardalos P., Sukhov P. (2013) Pattern-based heuristic for the cell formation problem in Group Technology. In: Goldengorin et al (eds) *Models, algorithms, and technologies for network analysis*. Springer Proceedings in Mathematics & Statistics 32, 11–50 (Автор диссертации принимал участие в обсуждении идей, реализовал алгоритм, провел вычислительные эксперименты и подготовил часть текста статьи)

Доклады на конференциях и семинарах:

- 8-я Международная конференция по сетевому анализу (NET 2018), 18-19 мая, Яндекс, Москва, Россия. From Cell Formation Problem to Biclustering and Graph Editing.

- Аспирантский семинар Высшей школы компьютерных наук НИУ ВШЭ, 29 сентября 2017 года, факультет компьютерных наук, Высшая школа экономики, Москва, Россия An effective exact model for solving the cell formation problem.
- Научный семинар лаборатории ЛАТАС, 2016, Нижний Новгород, Россия On solving manufacturing cell formation via bicluster editing.
- 20-я конференция Международной федерации исследования операций IFORS-2014, 13-18 июля 2014 г., Барселона, Испания. Multi-start local search heuristic for the cell formation problem.
- Третья Международная конференция по сетевому анализу, 2013, Нижний Новгород, Россия. An exact model for the cell formation problem.
- Вторая Международная конференция по сетевому анализу, 2012, Нижний Новгород, Россия. "Patterns" for solving the Cell Formation Problem.

2 Постановка задачи

2.1 Классическая формулировка

В наиболее часто встречающейся в литературе постановке задача CFP определяется двоичной матрицей A с m строками, представляющими оборудование (станки), и p столбцами, представляющими производимые детали. Элемент этой матрицы $a_{ij} = 1$, если деталь j должна в процессе своего создания обрабатываться на машине i . Цель состоит в том, чтобы сформировать производственные ячейки, которые состоят из находящихся в них машин и из деталей, которые обрабатываются преимущественно в данной ячейке. Эффективные разбиения на производственные ячейки позволяют оптимизировать некоторые производственные показатели, которые были упомянуты выше.

В качестве примера входных данных приведем задачу Waghodekar & Sahu [60], показанную в Таблице 1. Эта задача состоит из 5 машин и 7 частей. В Таблице 2 представлено допустимое решение, состоящее из двух производственных ячеек. Первая производственная ячейка содержит станки m_1, m_4 с деталями p_1, p_7 , а вторая производственная ячейка содержит станки m_2, m_3, m_5 с деталями p_2, p_3, p_4, p_5, p_6 . Некоторые детали в процессе производства должны быть перемещены для обработки из одной ячейки в другую (например деталь p_6 должна быть обработана на станке m_1 , поэтому в некоторый момент ее следует транспортировать из ячейки 2 в ячейку 1). Операции, лежащие вне ячеек, называются исключениями (exceptions). Элементы, которые находятся внутри производственных ячеек, но не являются операциями. Такие элементы называются простоями (voids) и отрицательно влияют на общую загрузку станков внутри производственной ячейки. Цель состоит в том, чтобы свести к минимуму количество исключений и количество простоев одновременно.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
m_1	1	0	0	0	1	1	1
m_2	0	1	1	1	1	0	0
m_3	0	0	1	1	1	1	0
m_4	1	1	1	1	0	0	0
m_5	0	1	0	1	1	1	0

Таблица 1: Исходная матрица станков-деталей 5×7 из Waghodekar & Sahu [60]

	p_7	p_1	p_2	p_3	p_4	p_5	p_6
m_1	1	1	0	0	0	1	1
m_4	0	1	1	1	1	0	0
m_2	0	0	1	1	1	1	0
m_3	0	0	0	1	1	1	1
m_5	0	0	1	0	1	1	1

Таблица 2: Решение для задачи из Waghodekar & Sahu [60] с двумя производственными ячейками

2.2 Целевые функции

Существует несколько различных целевых функций, предложенных исследователями в области для измерения качества бикластеризации деталей и станков. Chandrasekharan & Rajagopalan [24] предложил одну из первых количественных мер для сравнения решений задачи CFP. Эта мера называется производительность группировки (grouping efficiency). Данная метрика представляет собой взвешенную сумму двух значений η_1 и η_2 :

$$\eta = q\eta_1 + (1 - q)\eta_2$$

$$\eta_1 = \frac{n_1 - n_1^{out}}{n_1 - n_1^{out} + n_0^{in}} = \frac{n_1^{in}}{n_1^{in}},$$

$$\eta_2 = \frac{mp - n_1 - n_0^{in}}{mp - n_1 - n_0^{in} + n_1^{out}} = \frac{n_0^{out}}{n_0^{out}},$$

Здесь n_1 – общее число единиц в матрице станков-деталей, n_1^{in} – общее число элементов внутри всех производственных ячеек, n_1^{out} – общее число элементов вне производственных ячеек, n_1^{in} – число операций внутри производственных ячеек, n_1^{out} – число исключений, n_0^{in} – число простоев, n_0^{out} – число нулей вне ячеек. η_1 определяет степень загрузки станков в производственных ячейках, η_2 определяет перемещение деталей из одного цеха в другой, q – коэффициент ($0 \leq q \leq 1$), который определяет вес η_1 в целевой функции. В большинстве публикаций, рассматривающих метрику производительность группировки, устанавливаются равные веса для η_1 и η_2 .

Проведенные сравнения целевых функций для задачи формирования производственных ячеек показывают, что производительность группировки имеет существенные недостатки [54]. В частности, данная мера не всегда способна отличить эффективные разбиения от неэффективных. Производительность группировки дает значения от 0,75 до 1 в большинстве случаев в зависимости от коэффициента q [42]. В результате, некоторые менее эффективные разбиения могут иметь высокое значение целевой функции. Кроме того, с увеличением размера матрицы влияние исключений существенно уменьшается. Еще одним узким местом производительности группировки является необходимость выбора параметра q .

Kumar & Chandrasekharan [42] предложили еще одну целевую функцию для задачи CFP - эффективность группировки (grouping efficacy). Она определяется как:

$$\tau = \frac{n_1^{in}}{n_1 + n_0^{in}},$$

Эффективность группировки нивелирует недостатки предыдущей рассмотренной метрики. На текущий момент данная мера является самой популярной среди исследователей в области. Наконец, в некоторых работах в качестве целевой функции рассматривается сумма исключений и простоев в итоговом решении $E + V$:

$$E + V = n_1^{out} + n_0^{in}$$

Например, в статье Krushinsky & Goldengorin [39] для целевой функции $E+V$ были решены многие тестовые задачи вплоть до задач средних размеров.

2.3 Математическая модель

В данной работе преимущественно рассматривается постановка задачи CFP с целевой функцией эффективность группировки. В литературе также существуют различные варианты ограничений, связанных с размерами производственных ячеек:

- разрешаются ячейки любых размеров, в том числе размера 0 по одной из размерностей (ячеек, содержащих только машины или детали). Такие ячейки называются пустыми ячейками (residual cells) или зеротонами (zerotones).
- запрещаются ячейки, имеющие размера 0 по одной из размерностей
- запрещаются ячейки, имеющие размера 0 по одной из размерностей и ячейки, имеющие размер 1 по одной (или двум) размерностям (ячейки с одной машиной и одной или несколькими деталями или наоборот). Такие ячейки с одним элементом по одной или обоим размерностям принято называть синглтоны (singleton).

В большинстве научных статей запрещаются только зеротоны. Для классической формулировки мы предполагаем, что синглтоны могут появляться в решениях, а пустые ячейки запрещены. Однако, для оценки алгоритмов и подходов и сравнения с результатами из литературы в данной работе также используются первые два ограничения.

Ниже приводится формулировка задачи СФР в виде задачи целочисленного линейного программирования с целевой функцией эффективность группировки, разрешенными синглтонами и запрещенными зеротонами. Используется следующая нотация: m - число станков, p - число деталей, a_{ij} равно 1, если машина i обрабатывает деталь j , а c - максимально возможное число производственных ячеек. Поскольку каждая производственная ячейка должна содержать по крайней мере один станок и по крайней мере одну деталь $c = \min(m; p)$.

(IFP модель):

$$x_{ik} = \begin{cases} 1, & \text{если станок } i \text{ относится к ячейке } k, \\ 0, & \text{иначе} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{если деталь } j \text{ относится к ячейке } k, \\ 0, & \text{иначе} \end{cases}$$

$$\max \frac{\sum_{i=1}^m \sum_{j=1}^p \sum_{k=1}^c a_{ij} x_{ik} y_{jk}}{\sum_{i=1}^m \sum_{j=1}^p a_{ij} + \sum_{i=1}^m \sum_{j=1}^p \sum_{k=1}^c (1 - a_{ij}) x_{ik} y_{jk}} \quad (1)$$

Ограничения задачи:

$$\sum_{k=1}^c x_{ik} = 1, \quad i = 1, \dots, m \quad (2)$$

$$\sum_{k=1}^c y_{jk} = 1, \quad j = 1, \dots, p \quad (3)$$

$$\sum_{i=1}^m x_{ik} \leq m \cdot \sum_{j=1}^p y_{jk}, \quad k = 1, \dots, c \quad (4)$$

$$\sum_{j=1}^p y_{jk} \leq p \cdot \sum_{i=1}^m x_{ik}, \quad k = 1, \dots, c \quad (5)$$

Целевая функция (1) это метрика эффективность группировки, где числитель - это количество операций внутри ячеек (n_1^{in}), а две суммы в знаменателе - общее количество операций (n_1) и количество простоев внутри ячеек (n_0^{in}) соответственно. Ограничения (2) и (3) требуют, чтобы каждый станок и каждая деталь были назначены только одной производственной ячейке. Следующие два неравенства (4) и (5) запрещают пустые ячейки (без станков или деталей). Левая часть (4) - это общее количество станков, назначенных конкретной ячейке (эта сумма не превышает m), а правая часть - общее количество деталей, назначенных этой ячейке (умноженное на m). Это означает, что если у нас есть хотя бы один станок, назначенный некоторой ячейке, то должна быть хотя бы одна деталь также назначенная этой ячейке. Представленная модель позволяет нам иметь любое количество ячеек

в оптимальном решении не превышающее c . Например, если оптимальное решение имеет только две ячейки, то переменные x_{ik} и y_{jk} будут равны нулю для всех k , за исключением только двух значений k .

3 Существующие модели и подходы

Для решения задачи CFP было предложено множество различных подходов. Большинство из них предоставляют эвристические решения, и только несколькими авторами были предложены точные методы.

King and Nakornchai [38] и Chandrasekaran & Rajagopalan [22] предложили MODRIC и ROC2 - две улучшенные версии алгоритма кластеризации по рангам [37] и применили его к кластеризации матрицы станков-деталей. Chandrasekaran & Rajagopalan [21] представили ideal-seed алгоритм кластеризации с верхней оценкой на число кластеров, полученной с использованием представления задачи CFP в виде графа. Также Chandrasekaran & Rajagopalan [21] ввели численную меру качества решений для CFP, названную продуктивностью группировки. Позднее Chandrasekharan & Rajagopalan [23] также представили алгоритм ZODIAC, где строки и столбцы матрицы станков-деталей сначала разбивались на кластеры, а затем с использованием ideal seeds. Srinivasan & Narendran [56] представил GRAFICS - другой неиерархический кластерный подход для формирования ячеек, который первоначально группирует только станки, решая задачу о назначениях (assignment problem).

Многие авторы использовали теорию графов и графовое представление задачи CFP. Kumar et al. [40] сформулировал его как k -разложение графа, где станки и детали являются вершинами, а ребра - взаимосвязями для фиксированного числа k производственных ячеек. Vannelli & Kumar [59] и Kumar & Vannelli [41] используют графовые модели для определения станков и деталей, необходимых для получения идеальной бикластеризации матрицы станков-деталей, которая также называется блочно-диагональной структурой. Askin et al. [2] сформирована графовая структура с использованием коэффициентов подобия между станкам и деталями, а затем сформулировал задачу бикластеризации матрицы станков-деталей как задачу нахождения Гамильтонова пути. Ng [50] и Ng [51] использовали для решения CFP остовные деревья. Kusiak [43], Wang & Roze [61] и Won [62] предложили различные модель математического программирования, основанные на задаче поиска p -медианы.

3.1 Современные эвристические алгоритмы

Многие современные лучшие результаты для сложных задач комбинаторной оптимизации получены с помощью метаэвристических или гибридных метаэвристических алгоритмов, и задача о формировании производственных ячеек не является исключением. Одна из известных работ принадлежит Gonçalves and Resende [32]. В исследовании Gonçalves and Resende [32] был представлен эффективный генетический алгоритм для кластеризации станков и алгоритм локального поиска для

дальнейшей кластеризации деталей. Авторы собрали и предоставили набор данных 35 GT задач, который внес большой вклад в дальнейшую работу над задачей CFP и появление еще более эффективных алгоритмов.

James et al. [36] представила генетический алгоритм, в котором хромосома кодирует кластер станков и деталей одновременно. Автор использовал локальный поиск из статьи Gonçalves and Resende [32] и добились впечатляющих результатов используя эффективность группировки в качестве целевой функции.

Brusco [8] была предложен простая эвристика итеративного локального поиска с перемещением станков и деталей между производственными ячейками в качестве локального поиска и несколько вариантов случайных перемещений станков и деталей для фазы пертурбации.

В работах Goldengorin et al. [30] и Goldengorin et al. [31] авторам удалось получить хорошие результаты на примерах CFP больших размеров для целевой функции производительность группировки с помощью решения задачи о р-медиане.

3.2 Современные точные алгоритмы и модели

Krushinsky & Goldengorin [39] и Goldengorin et al. [31] разработали две точных MINpCUT модели, которые основаны на известной задаче нахождения наименьшего k-разреза графа. В качестве целевой функции использовалась минимизация числа исключений для фиксированного числа производственных ячеек.

Elbenani & Ferland [29] опубликовал модель смешанного целочисленного линейного программирования, которая максимизирует наиболее популярную целевую функцию - эффективность группировки, впервые сформулированную Kumar & Chandrasekharan [42]. В работе было предложено использовать алгоритм Динкельбаха для дробной целевой функции. Модель Elbenani & Ferland [29] рассчитана на заранее заданное фиксированное число производственных ячеек. Таким образом данный подход не может гарантировать получение глобального оптимума для всех возможных конфигураций производственных ячеек. Также на многих тестовых примерах вычислительные эксперименты были остановлены из-за ограничений по используемому объему памяти или времени, как следствие даже для фиксированного числа производственных ячеек решения не были найдены.

Brusco [9] разработал два точных подхода для решения задачи о формировании производственных ячеек с целевой функцией эффективность группировки. Первая модель представляет собой задачу смешанного целочисленного линейного программирования, которая основана на задаче кластеризации по двум размерностям с некоторыми упрощающими допущениями (например, количество кластеров по строкам и столбцам должно быть одинаковым). Эта модель выглядит интересно, но требует слишком много времени для решения тестовых примеров даже средних размеров. Второй подход это реализация алгоритма ветвей и границ в сочетании с эвристикой перемещения станков и деталей для получения начального решения. По результатам экспериментов метод ветвей и границ способен решить примерно в два раза больше тестовых примеров, также значительно сократилось время вычислений. Автором было заключено, что алгоритм отлично работает на хорошо структурированных данных - задачах среднего размера со значением эффективности группировки более 0.65 - 0.7. Для обоих методов, описанных выше,

были сделаны два основных допущения. Первое - разрешены синглтоны (ячейки, содержащие только один станок или одну деталь), что является довольно распространенной практикой. Второе допущение - разрешается наличие зеротоннов (ячеек, содержащих только станки без деталей или только детали без станков). Кроме того, количество производственных ячеек фиксированно для обоих подходов, но для части тестовых задач в вычислительных экспериментах учитывается несколько возможных значений этого параметра.

Некоторые авторы использовали подходы из области бикластеризации для решения задачи о формировании производственных ячеек.

Boutsinas [7] применил кластеризацию одновременно для двух размерностей (станков и деталей) и минимизировал сумму исключений и простоев.

В [63] представлен точный подход, который находит множество Парето и решает двухкритериальную задачу оптимизации, которая может быть сформулирована как задача о формировании производственных ячеек.

Pinheiro et al. [53] свел задачу формирования ячеек к другой задаче бикластеризации - задаче редактирования двудольных графов. Автором был предложен точный метод и модель линейного программирования. Оба метода максимизируют эффективность группировки и показывают хорошую производительность на широком наборе тестовых примеров.

4 Содержание работы

К основным результатам текущего исследования можно отнести следующие:

- доказательство NP-полноты для задачи разрешимости CFP с целевой функцией эффективность группировки и линейной функцией $E + V$
- точный подход для решения задачи CFP, основанный на оригинальной целочисленной модели линейного программирования и алгоритме Динкельбаха
- точный подход для решения задачи CFP, основанный на фиксации значения знаменателя целевой функции эффективность группировки и оригинальной модели целочисленного линейного программирования
- эвристический алгоритм поиска во многих окрестностях с рестартами для целевой функции производительность группировки

Многие исследователи в этой области используют для вычислительных экспериментов набор данных 35 GT, предоставленный Gonçalves and Resende [32]. Эти примеры взяты из различных исследовательских работ по задаче CFP (ссылки на источники приведены в разделе 4). Некоторые примеры тестовых задач в этом наборе данных имеют ошибки и отличаются от тех, которые представлены в оригинальных статьях. Часть исследователей, включая Elbenani & Ferland [29] и Pinheiro et al. [53], провели свои вычислительные эксперименты, используя именно этот набор тестовых задач из Gonçalves and Resende [32]. Однако, есть и авторы, использовавшие в той или иной степени скорректированную версию набора данных. С точки зрения проведения вычислительных экспериментов такие различия приводят к невозможности сравнения подходов и их оценке. В текущей работе были рассмотрены все источники и исправлены ошибки, имеющиеся в наборе данных

35 GT. Скорректированные данные (Testset A) доступны по ссылкам в библиографии, что является одним из вкладов в область, которые вносит данная работа. Помимо исправленного набора 35 GT в текущей работе было сформировано еще одно множество тестовых задач из различных источников, реже используемых в вычислительных экспериментах (Testset B). Все данные можно загрузить с сайта opt-hub.com или researchgate.net (полные URL-адреса можно найти в ссылках).

4.1 Сложность CFP

Несмотря на то, что популярность задачи о формировании производственных ячеек в последнее время связана именно с трудностью проведения вычислений для больших тестовых примеров, формальное доказательство принадлежности задачи к классу NP-трудных или NP-полных задач автору работы найти не удалось. В настоящем разделе данный пробел восполняется для целевых функций $E + V$ и эффективность группировки. Задача CFP в данном разделе рассматривается как задача разрешимости. Все результаты связанные со сложностью задачи CFP изложены в работе Batsyn et al. [5].

Сначала мы рассмотрим так называемую задачу редактирования двудольного графа (Bicliaster Graph Editing Problem, BGER). BGER состоит в определении минимального числа ребер, которые должны быть добавлены или удалены из данного двудольного графа, чтобы он превратился в набор изолированных биклик (bicliques). NP-полнота BGER была доказана в [1].

Теорема 1. [1] *Задача редактирования двудольного графа является NP-полной за счет сведения задачи о 3-точном 3-покрытии множества.*

Используя этот результат, в данной работе доказывается, что задача о формировании производственных ячеек с целевой функцией $E + V$ также является NP-полной.

Теорема 2. *Задача о формировании производственных ячеек с линейной целевой функцией $E + V$ является NP-полной так как она эквивалентна задаче о редактировании двудольного графа.*

Между задачей CFP как задачей разрешимости и задачей BGER существует взаимно однозначное соответствие. Каждый станок в CFP соответствует вершине одной из долей двудольного графа в BGER, и каждая деталь в CFP соответствует вершине в оставшейся доле этого графа. Матрица операций в CFP совпадает с матрицей смежности двудольного графа в BGER. Каждое исключение в решении CFP соответствует ребру, которое должно быть удалено из двудольного графа в BGER, чтобы преобразовать его в набор изолированных биклик. И каждая пустота в решении CFP соответствует ребру, которое должно быть добавлено к двудольному графу в BGER. Целевая функция $E + V$ эквивалентна целевой функции BGER - минимизации количества добавленных/удаленных ребер, необходимых для преобразования входного двудольного графа в набор изолированных биклик. Каждая биклика соответствует прямоугольной производственной ячейке в CFP.

Чтобы доказать NP-полноту задачи CFP с целевой функцией эффективность группировки в данной работе предлагается ее сведение к задаче CFP с целевой

$$\left[\begin{array}{c} \left[\begin{array}{ccc} & & \\ & A & \\ \left[\begin{array}{ccc} 0 & \dots & 0 \end{array} \right] & & \\ \vdots & & \vdots \\ \left[\begin{array}{ccc} 0 & \dots & 0 \end{array} \right] & & \end{array} \right] \left[\begin{array}{ccc} \left[\begin{array}{ccc} 0 & \dots & 0 \end{array} \right] \\ \vdots & & \vdots \\ \left[\begin{array}{ccc} 0 & \dots & 0 \end{array} \right] \\ 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{array} \right] \end{array} \right]$$

Рис. 1: Исходная матрица операций A расширенная с помощью большого блока из единиц

функцией $E + V$. Функция эффективность группировки может быть представлена как:

$$\frac{n_1^{in}}{n_1 + n_0^{in}} = 1 - \frac{e + v}{n_1 + v} \rightarrow \max \iff \frac{e + v}{n_1 + v} \rightarrow \min$$

Это выражение практически соответствует линейной функции $E + V$ за исключением значения V в знаменателе. Оригинальная идея данной работы - нивелировать влияние значения V в знаменателе за счёт существенного увеличения общего числа единиц в матрице операций - n_1 . Задача CFP с целевой функцией $E + V$ сводится к CFP с функцией групповой эффективности за счёт расширения изначальной матрицы операций с помощью большого блока единиц. Иллюстрация идеи отображена на Рисунке 1.

Предложение 1. Если в матрице операций задачи CFP с целевой функцией эффективность группировки имеются одинаковые строки, то существуют оптимальные решения, в которых эти строки принадлежат одной производственной ячейке.

Предложение 2. Если число единиц в расширенной матрице операций равно $(tr)^2$ то максимальное значение функции эффективность группировки для исходной подматрицы расширенной матрицы и минимальное значение функции $E + V$ для исходной матрицы A будут получены на одном и том же решении.

Два результата выше использованы в доказательстве следующей теоремы:

Теорема 3. Задача CFP с целевой функцией эффективность группировки является NP-полной, так как между ней и задачей CFP с целевой функцией $E + V$ существует полиномиальное сведение.

Так как задача CFP в виде задачи разрешимости с целевой функцией эффективность группировки является NP-полной, то оптимизационная версия задачи принадлежит к классу NP-трудных задач.

4.2 Точный подход для решения задачи CFP с целевой функцией эффективность группировки, основанный на фиксации значения знаменателя

Эффективность группировки является дробной целевой функцией. В качестве одного из методов работы с ней в данной работе предлагается идея оценки возможных нижних и верхних границ на значение знаменателя и дальнейший их перебор. В особенности, необходимо перебрать значения переменной n_0^{in} , так как вторая составляющая знаменателя n_1 является константой. Решается множество задач CFP для каждого возможного значения n_0^{in} посредством добавления дополнительного ограничения, которое в явном виде фиксирует количество нулей внутри производственных ячеек. Таким образом, оригинальная целевая функция эффективность группировки сводится к линейной максимизации переменной n_1^{in} . Однако, при таком подходе приходится решать множество задач CFP для каждого возможного фиксированного значения n_0^{in} . Глобальный оптимум в дальнейшем определяется максимальным значением эффективности группировки среди всех решений, полученных на предыдущем шаге. Для ограничения множества значений n_0^{in} предлагается использовать любое имеющееся допустимое решение задачи. Предложение 3 позволяет получить верхнюю оценку на возможное число нулей внутри решения CFP. Предложение 4 в свою очередь позволяет получить нижнюю границу для величины n_1^{in} , чтобы исключить значения n_1^{in} при которых значение эффективности группировки на искомом решении задачи будет не больше чем известное допустимое значение целевой функции.

Предложение 3. Пусть τ значение эффективности группировки для некоторого допустимого решения задачи CFP. Тогда n_0^{in} в оптимальном решении не превосходит $\lfloor \frac{1-\tau}{\tau} \cdot n_1 \rfloor$.

Предложение 4. Пусть τ значение эффективности группировки для некоторого допустимого решения задачи CFP. Тогда значение n_1^{in} в оптимальном решении должно быть не менее $\lceil \tau \cdot (n_1 + n_0^{in}) \rceil$.

Предложение 3 таким образом используется для сокращения возможного количества нулей внутри искомого оптимального решения CFP, а именно уменьшает количество подзадач, которые нужно решить для каждого фиксированного значения n_0^{in} .

Трехиндексная модель:

$$x_{ik} = \begin{cases} 1, & \text{если станок } i \text{ принадлежит к ячейке } k, \\ 0, & \text{иначе} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{если деталь } j \text{ принадлежит к ячейке } k, \\ 0, & \text{иначе} \end{cases}$$

$$z_{ijk} = \begin{cases} 1, & \text{если станок } i \text{ и деталь } j \text{ принадлежат к ячейке } k, \\ 0, & \text{otherwise} \end{cases}$$

$$\max \sum_{i=1}^m \sum_{j=1}^p \sum_{k=1}^c a_{ij} z_{ijk} \quad (6)$$

Ограничения:

$$z_{ijk} \leq x_{ik}, \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, p, \quad \forall k = 1, \dots, c \quad (7)$$

$$z_{ijk} \leq y_{jk}, \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, p, \quad \forall k = 1, \dots, c \quad (8)$$

$$z_{ijk} \geq x_{ik} + y_{jk} - 1, \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, p, \quad \forall k = 1, \dots, c \quad (9)$$

$$\sum_{k=1}^c x_{ik} = 1, \quad \forall i = 1, \dots, m \quad (10)$$

$$\sum_{k=1}^c y_{jk} = 1, \quad \forall j = 1, \dots, p \quad (11)$$

$$\sum_{i=1}^m \sum_{j=1}^p z_{ijk} \geq \sum_{i=1}^m x_{ik}, \quad \forall k = 1, \dots, c \quad (12)$$

$$\sum_{i=1}^m \sum_{j=1}^p z_{ijk} \geq \sum_{j=1}^p y_{jk}, \quad \forall k = 1, \dots, c \quad (13)$$

$$\sum_{i=1}^m \sum_{j=1}^p \sum_{k=1}^c (1 - a_{ij}) z_{ijk} = n_0^{in} \quad (14)$$

$$\lceil \tau(n_1 + n_0^{in}) \rceil \leq \sum_{i=1}^m \sum_{j=1}^p \sum_{k=1}^c a_{ij} z_{ijk} \leq n_1 \quad (15)$$

$$x_{ik}, y_{jk}, z_{ijk} \in \{0, 1\} \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, p, \quad \forall k = 1, \dots, c \quad (16)$$

Модель описывает задачу CFP с помощью бинарных переменных x_{ik} и y_{jk} . Индекс станков i принимает значения от 1 до m , индекс деталей j от 1 до значения p . Индекс производственной ячейки k принимает значения 1 до $c = \min(m, p)$ так как каждая производственная ячейка должна содержать не менее одного станка и одной детали, соответственно эта величина не может превышать m и p . Число единиц внутри всех ячеек равно $\sum_{k=1}^c \sum_{i=1}^m \sum_{j=1}^p a_{ij} x_{ik} y_{jk}$, а число нулей внутри всех ячеек равно $\sum_{k=1}^c \sum_{i=1}^m \sum_{j=1}^p (1 - a_{ij}) x_{ik} y_{jk}$. Произведение $x_{ik} y_{jk}$ линеаризуется стандартной процедурой - вводом новой бинарной переменной $z_{ijk} = x_{ik} y_{jk}$ и дополнительными ограничениями (7), (8), (9).

Равенства (10), (11) определяют принадлежность каждого станка и каждой детали к ровно одной производственной ячейке. Ограничения (12) и (13) запрещают производственные ячейки в которых есть только станки или только детали. Равенство (14) фиксирует число нулей внутри искомого решения в виде константы n_0^{in} .

Ограничения (15) определяют возможное число единиц внутри ячеек в искомом решении. Предложенная модель (6) - (16) решается для всех возможных n_0^{in} с помощью CPLEX.

#	Источник	Размер	ЭГ	Время, с
1	King and Nakornchai (1982) - Figure 1a	5x7	0.8235	0.63
2	Waghodekar and Sahu (1984) - Problem 2	5x7	0.6957	2.29
3	Seifoddini (1989b)	5x18	0.7959	5.69
4	Kusiak and Cho (1992)	6x8	0.7692	1.86
5	Kusiak and Chow (1987)	7x11	0.6087	9.14
6	Boctor (1991) - Example 1	7x11	0.7083	5.15
7	Seifoddini and Wolfe (1986)	8x12	0.6944	13.37
8	Chandrasekaran and Rajagopalan (1986a)	8x20	0.8525	18.33
9	Chandrasekaran and Rajagopalan (1986b)	8x20	0.5872	208.36
10	Mosier and Taube (1985a)	10x10	0.75	6.25
11	Chan and Milner (1982)	10x15	0.92	2.93
12	Askin and Subramanian (1987)	14x23	0.7206	259.19
13	Stanfel (1985)	14x24	0.7183	179.21
14	McCormick et al. (1972)	16x24	0.5161*	20829.38*
15	Srinivasan et al. (1990)	16x30	0.6900*	13719.99*
16	King (1980)	16x43	0.5753*	24930.93*
17	Carrie (1973)	18x24	0.5773*	13250.01*
18	Mosier and Taube (1985b)	20x20	0.3871*	43531.77*
19	Kumar et al. (1986)	20x23	0.4672*	33020.13*
20	Carrie (1973)	20x35	0.7785*	11626.98*
21	Boe and Cheng (1991)	20x35	0.4675*	33322.08*
22	Chandrasekharan and Rajagopalan (1989) - 1	24x40	1	1.64
23	Chandrasekharan and Rajagopalan (1989) - 2	24x40	0.8511*	6916.24*
24	Chandrasekharan and Rajagopalan (1989) - 3	24x40	0.5649*	14408.88*
25	Chandrasekharan and Rajagopalan (1989) - 5	24x40	0.4656*	34524.47*
26	Chandrasekharan and Rajagopalan (1989) - 6	24x40	0.4351*	41140.94*
27	Chandrasekharan and Rajagopalan (1989) - 7	24x40	0.4122*	44126.76*
28	McCormick et al. (1972)	27x27	0.54.02*	22627.28*
29	Carrie (1973)	28x46	0.2465*	71671.08*
30	Kumar and Vannelli (1987)	30x41	0.4844*	22594.2*
31	Stanfel (1985) - Figure 5	30x50	0.5065*	31080.82*
32	Stanfel (1985) - Figure 6	30x50	0.3832*	48977.0*
33	King and Nakornchai (1982)	30x90	0.3941*	99435.64*
34	McCormick et al. (1972)	37x53	0.5960*	47744.04*
35	Chandrasekharan and Rajagopalan (1987)	40x100	0.8403*	24167.76*

Таблица 3: Результаты трехиндексной модели с фиксированием знаменателя

Для вычислительных экспериментов в работе используется популярный набор данных 35 GT из литературы. Результаты приведены в Таблице 3 (столбец ЭГ - значение эффективности группировки). Для задач с 1 по 13 и задачи 22 глобально

оптимальные решения были найдены за сравнительно небольшое время вычислений. Для этих задач найденные точные решения соответствуют лучшим эвристическим решениям из разных источников. Таким образом, используя подход с фиксированным знаменателем удалось доказать оптимальность для 14 решений рассматриваемого набора, что не было сделано авторами предыдущих работ по задаче CFP. Значения эффективности группировки для таких задач выделены в таблице. Решение остальных задач набора 35 GT потребовало использования слишком большого количества памяти или не уложилось в ограничение по времени. Для таких задач CPLEX был запущен с ограничением в 300 секунд на каждую подзадачу и в качестве результата представлены лучшие из полученных значений. Все вычисления были проведены на компьютере с процессором Intel Core i7, частотой в 2.2 ГГц и 8 гигабайтами оперативной памяти.

4.3 Точный подход, основанный на двухиндексной модели и алгоритме Динкельбаха

Из-за большого количества переменных и ограничений в трехиндексной модели [15] CPLEX зачастую тратит очень много вычислительных ресурсов даже для решения задач небольших размеров (глобальный оптимум с помощью трехиндексной модели был найден только для 14 задач из 35 в наборе данных 35 GT). Также, число подзадач, которые необходимо решить, могло быть очень большим в примерах с низким значением эффективности группировки. В качестве более перспективного подхода в данной работе предлагается двухиндексная модель целочисленного линейного программирования и подход, основанный на алгоритме Динкельбаха, для линеаризации целевой функции. Ключевая идея новой модели - удаление отношений вида станок-деталь-ячейка. Вместо соответствия элементов конкретным ячейкам в новой модели используется простой факт. Если станки находятся в одной производственной ячейке, то они входят туда с одинаковым набором деталей, также входящих в эту ячейку.

Двухиндексная модель целочисленного линейного программирования - это оригинальная модель, которая позволяет получать глобально оптимальные решения задачи CFP с любым допустимым количеством ячеек и целевой функцией эффективности группировки. В отличие от большинства моделей математического программирования, предложенных для задачи CFP в литературе, двухиндексная модель не содержит прямого назначения машин или деталей на ячейки. Используются отношения вида станок-станок и станок-деталь вместо широко используемого назначения станок-деталь-ячейка. Это приводит к более компактной и элегантной формулировке, учитывающей только те ограничения, которые обеспечивают блочно-диагональную структуру решений, что в свою очередь позволяет значительно сократить количество переменных и ограничений в модели и получить глобально оптимальные решения даже для некоторых тестовых примеров больших размеров.

Вместе с двухиндексной моделью используется отличный от подхода с трехиндексной моделью способ линеаризации целевой функции - алгоритм Динкельбаха [26]. Этот параметрический подход был предложен В.Динкельбахом и является

одним из самых популярных подходов для решения задач дробно-линейного программирования. Алгоритм Динкельбаха сводит задачу дробно-линейного программирования к решению серии более простых задач. Пусть есть задача дробно-линейного программирования со следующей целевой функцией:

$$Q(x) = \frac{P(x)}{D(x)}, \quad (17)$$

тогда алгоритм Динкельбаха выглядит следующим образом:

- **Шаг 1** рассмотрим некоторое допустимое решение x^0 , вычислим $\lambda_1 = \frac{P(x^0)}{D(x^0)}$ and let $k = 1$
- **Шаг 2** решим исходную задачу с целевой функцией $Q(x)$ замененной на $F(\lambda_k) = P(x) - \lambda_k D(x) \rightarrow \max$ и обозначим оптимальное решение для этой задачи x^k
- **Шаг 3** Если $F(\lambda_k)$ равна 0 (или меньше чем некоторая заданная заранее точность) тогда процедура останавливается и x^k возвращается как оптимальное решение
Иначе $k = k + 1, \lambda_k = \frac{P(x^k)}{D(x^k)}$ и переходим к Шагу 2.

Elbenani & Ferland [29] также использует алгоритм Динкельбаха для линеаризации целевой функции эффективность группировки. Несмотря на достаточно быстрый работающий подход Elbenani & Ferland [29] рассматривает только одно возможное число производственных ячеек.

Двухиндексная модель:

$$x_{ik} = \begin{cases} 1, & \text{если станки } i \text{ и } k \text{ принадлежат одной ячейке,} \\ 0, & \text{иначе} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{если станок } i \text{ и деталь } j \text{ принадлежат одной ячейке} \\ 0, & \text{otherwise} \end{cases}$$

$$\max \sum_{i=1}^m \sum_{j=1}^p a_{ij} y_{ij} - \lambda \cdot \left(\sum_{i=1}^m \sum_{j=1}^p (1 - a_{ij}) y_{ij} + \sum_{i=1}^m \sum_{j=1}^p a_{ij} \right) \quad (18)$$

Subject to:

$$2x_{ik} - y_{ij} - y_{kj} \geq -1 \quad i, k = 1, \dots, m \quad j = 1, \dots, p \quad (19)$$

$$y_{ij} - y_{kj} - x_{ik} \geq -1 \quad i, k = 1, \dots, m \quad j = 1, \dots, p \quad (20)$$

$$y_{kj} - y_{ij} - x_{ij} \geq -1 \quad i, k = 1, \dots, m \quad j = 1, \dots, p \quad (21)$$

$$\sum_{j=1}^p y_{ij} \geq 1 \quad i = 1, \dots, m \quad (22)$$

$$\sum_{i=1}^m y_{ij} \geq 1 \quad j = 1, \dots, p \quad (23)$$

Целевая функция (18) это функция эффективность группировки, линеаризованная с помощью подхода Динкельбаха. Ограничения (19), (20), (21) определяют отношения между станками и деталями, а именно блочно-диагональную структуру допустимого решения. Последние два неравенства(22) и (23) запрещают появление в решениях зеротонов (ячеек без деталей или без станков).

Предлагаемый метод начинается с задания значения λ равным лучшему известному допустимому значению эффективности группировки для каждого конкретного тестового примера (также для этого можно использовать предложенный в данной работе эвристический алгоритм, описанный в следующих главах). Далее решается серия из задач в двухиндексной постановке и согласно алгоритму Динкельбаха обновляется значение параметра λ (пока параметр больше 0). Для вычислительных экспериментов двухиндексной модели с алгоритмом Динкельбаха были использованы два набора тестовых задач. Первый - исправленный набор 35GT, названный в текущей работе Testset A. Второй набор задач, собранный для дополнительных тестов из литературы назван Testset B. Все ссылки на тестовые задачи могут быть найдены в статье Vuchkov & Batsyn [17].

Testset A - Классический набор. Первый тестовый набор - набор 35GT, считающийся классическим тестовым набором в области, был предложен Gonçaves and Resende [32]. Он содержит 35 тестовых примеров с размерами от 5×7 до 40×100 (нотация - станки \times детали). Данный тестовый набор очень популярен и именно для него получено огромное число результатов для задачи CFP различными подходами (в основном эвристическими и метаэвристическими алгоритмами). Как было отмечено ранее в работе, некоторые тестовые примеры в 35 GT наборе отличаются от тех статей, из которых они были адаптированы. В текущей работе набор Testset A является исправленной версией 35 GT.

ID	Источник	m	p
A1	King and Nakornchai (1982) - Figure 1a	5	7
A2	Waghodekar and Sahu (1984) - Problem 2	5	7
A3	Seifoddini (1989b)	5	18
A4	Kusiak and Cho (1992)	6	8
A5	Kusiak and Chow (1987)	7	11
A6	Boctor (1991) - Example 1	7	11
A7	Seifoddini and Wolfe (1986)	8	12
A8	Chandrasekaran and Rajagopalan (1986a)	8	20
A9	Chandrasekaran and Rajagopalan (1986b)	8	20
A10	Mosier and Taube (1985a)	10	10
A11	Chan and Milner (1982)	15	10
A12	Askin and Subramanian (1987)	14	24
A13	Stanfel (1985)	14	24
A14	McCormick et al. (1972)	16	24
A15	Srinivasan et al. (1990)	16	30
A16	King (1980)	16	43
A17	Carrie (1973)	18	24
A18	Mosier and Taube (1985b)	20	20
A19	Kumar et al. (1986)	23	20
A20	Carrie (1973)	20	35
A21	Boe and Cheng (1991)	20	35
A22	Chandrasekharan and Rajagopalan (1989) - 1	24	40
A23	Chandrasekharan and Rajagopalan (1989) - 2	24	40
A24	Chandrasekharan and Rajagopalan (1989) - 3	24	40
A25	Chandrasekharan and Rajagopalan (1989) - 5	24	40
A26	Chandrasekharan and Rajagopalan (1989) - 6	24	40
A27	Chandrasekharan and Rajagopalan (1989) - 7	24	40
A28	McCormick et al. (1972)	27	27
A29	Carrie (1973)	28	46
A30	Kumar and Vannelli (1987)	30	41
A31	Stanfel (1985) - Figure 5	30	50
A32	Stanfel (1985) - Figure 6	30	50
A33	King and Nakornchai (1982)	30	90
A34	McCormick et al. (1972)	37	53
A35	Chandrasekharan and Rajagopalan (1987)	40	100

Таблица 4: Testset A - Задачи

Testset B - Дополнительные задачи. Второй набор данных *Testset B* состоит из более редких тестовых задач. Было проанализирован ряд работ по тематике CFP и использованные там тестовые примеры, которые практически не встречаются в других работах. Из данных задач был сформирован новый тестовый набор. Он включает 32 тестовых задачи размерами от 6×6 до 50×150 . Несколько примеров были адаптированы из более сложных вариаций задачи CFP.

Для вычислительных экспериментов были рассмотрены две наиболее популярных версии ограничений на размеры ячеек:

1. Зеротоны запрещены, синглтоны (ячейки с одним станком и/или с одной деталью) и ячейки больших размеров разрешены

2. Зеротоны разрешены

Для сравнения предложенных в данной работе подходов были выбраны несколько лучших существующих точных методов решения задачи CFP из литературы. Аппаратные ресурсы для экспериментов: компьютер на базе процессора Intel Xeon с частотой 3.4 ГГц и 16 гигабайтами оперативной памяти. Программные ресурсы - CPLEX версии 12.4.0.

ID	Источник	m	p
B1	Adil (1996)	6	6
B2	Parkin and Li (1997)	6	7
B3	Brown and Sumichrast (2001)	6	11
B4	Chan and Milner (1982)	7	5
B5	Kusiak and Chow (1987)	7	8
B6	Zolfaghari and Liang (2002)	7	8
B7	Won and Kim (1997)	7	10
B8	Sarker and Khan (2001)	8	8
B9	Nair (1999)	8	10
B10	Islam and Sarker (2000)	8	10
B11	Kumar et al. (1986)	9	15
B12	Ham et al. (1985)	10	8
B13	Viswanathan (1996)	10	12
B14	Shargal et al. (1995)	10	38
B15	Won and Kim (1997)	11	10
B16	Seifoddini (1988)	11	22
B17	Moon and Chi (1992)	12	19
B18	Li (2003)	14	14
B19	Chan and Milner (1982) - Fig.3a	15	10
B20	Yang and Yang (2008) - Fig.6b	15	15
B21	Yang and Yang (2008) - Fig.6c	15	15
B22	Yang and Yang (2008) - Fig.6d	15	15
B23	Harhalakis et al. (1994)	17	20
B24	Seifoddini and Djassemi (1991)	18	24
B25	Sandbothe (1998)	20	10
B26	Nagi et al. (1990)	20	51
B27	Won and Kim (1997)	26	28
B28	Yang and Yang (2008) - Fig.7	28	35
B29	Seifoddini and Djassemi (1996)	35	15
B30	Seifoddini and Djassemi (1996)	41	50
B31	Yang and Yang (2008) - Fig.12	46	105
B32	Zolfaghari and Liang (1997)	50	150

Таблица 5: Testset B - Задачи

Тестовые задачи из Таблицы 4 широко используются в литературе. Результаты представлены отдельно для формулировки с минимальным размером ячейки 1×1

(Таблица 6 и Рисунок 2) и для формулировки с разрешенными зеротонами (Таблица 7 и Рисунок 3). В первом случае было произведено сравнение с двумя подходами:

1. Точный подход из Elbenani & Ferland [29]
2. Точный подход из Vychkov et al. [15]

#	Время, сек			ЭГ		
	Elbenani & Ferland (2012)	Vychkov et al. (2014)	two-index model	Elbenani & Ferland (2012) (cells)	Vychkov et al. (2014)	two-index model
A1	2.3	0.63	0.00	0.8235(2)	0.8235	0.8235
A2	1.6	2.29	0.00	0.6957(2)	0.6957	0.6957
A3	3.1	5.69	0.00	0.7959(2)	0.7959	0.7959
A4	2.0	1.86	0.09	0.7692(2)	0.7692	0.7692
A5	30.6	9.14	0.17	0.6087(5)	0.6087	0.6087
A6	4.3	5.15	0.01	0.7083(4)	0.7083	0.7083
A7	9.6	13.37	0.02	0.6944(4)	0.6944	0.6944
A8	3.1	18.33	0.01	0.8525(3)	0.8525	0.8525
A9	3.5	208.36	0.45	0.5872(2)	0.5872	0.5872
A10	1.1	6.25	0.00	0.7500(5)	0.7500	0.7500
A11	1.6	2.93	0.02	0.9200(3)	0.9200	0.9200
A12	2188.7	259.19	0.19	0.7206(7)	0.7206	0.7206
A13	593.2	179.21	0.23	0.7183(7)	0.7183	0.7183
A14	15130.5	*	4.24	0.5326(8)	*	0.5326
A15	252.5	*	0.25	0.6953(6) ^E	*	0.6899
A16	183232.5	*	4.80	0.5753(8)	*	0.5753
A17	2345.6	*	3.82	0.5773(9)	*	0.5773
A18	*	*	32243.10	*	*	0.4345
A19	131357.5	*	245.59	0.5081(7)	*	0.5081
A20	31.1	*	0.22	0.7791(5)	*	0.7791
A21	14583.6	*	24.34	0.5798(5)	*	0.5798
A22	11.3	1.64	0.14	1.0000(7)	1.0000	1.0000
A23	230.7	*	0.12	0.8511(7)	*	0.8511
A24	1101.1	*	0.16	0.7351(7)	*	0.7351
A25	*	*	1026.96	*	*	0.5329
A26	*	*	178182.24	*	*	0.4895
A27	*	*	*	*	*	*
A28	958714.1	*	1964.00	0.5482(5)	*	0.5482
A29	*	*	*	*	*	*
A30	378300.0	*	8.72	0.6331(14)	*	0.6331
A31	*	*	136.00	0.6012(13) ^E	*	0.5977
A32	*	*	*	*	*	*
A33	*	*	*	*	*	0.4800
A34	268007.6	*	16323.71	0.6064(3)	*	0.6064
A35	7365.3	*	1.34	0.8403(10)	*	0.8403

Таблица 6: Testset A - Результаты сравнения подходов (зеротоны запрещены, синглтоны разрешены)

В Elbenani & Ferland [29] рассмотрена упрощенная формулировка - задача решается только для одного заранее выбранного числа ячеек. Авторы выполнили свои эксперименты на компьютере с процессором AMD частотой 2.2 ГГц и 4 гигабайтами оперативной памяти. Для Testset A в качестве начальных значений

для параметра λ были использованы имеющиеся решения из литературы.

В случае с любыми допустимыми размерами ячеек представленный подход сравнивался со следующими работами:

1. Алгоритм ветвей и границ Brusco [9]
2. Модель линейного программирования Pinheiro et al. [53]
3. Итеративный точный метод Pinheiro et al. [53]

#	Время, сек				ЭГ		
	Brusco (2015)	Pinheiro et al. (2016) IM	Pinheiro et al. (2016) ILP	two-index	Brusco (2015) (cells)	Pinheiro et al. (2016)	two-index
A1	0.01	0.16	0.01	0.01	0.8235(2,3,4)	0.7500 ^E	0.8235
A2	0.01	0.07	0.01	0.01	0.6957(2,3,4)	0.6956	0.6957
A3	0.02	0.09	0.03	0.01	0.8085(2,3,4)	0.8085	0.8085
A4	0.01	0.02	0.01	0.01	0.7916(2,3,4)	0.7917	0.7917
A5	0.6	0.29	0.06	0.17	0.6087(2,3,4,5,6)	0.6087	0.6087
A6	0.04	0.14	0.01	0.01	0.7083(2,3,4,5)	0.7083	0.7083
A7	0.08	0.18	0.03	0.01	0.6944(2,3,4,5)	0.6944 ^E	0.6944
A8	0.01	2.06	0.04	0.01	0.8525(2,3,4)	0.8525	0.8525
A9	35.86	81.46	4.94	0.45	0.5872(2,3,4)	0.5872	0.5872
A10	0.06	0.03	0.01	0.01	0.7500(2,3,4,5,6)	0.7500	0.7500
A11	0.01	0.01	0.02	0.02	0.9200(2,3,4)	0.9200	0.9200
A12	633.91	0.49	0.09	0.03	0.7424(6,7,8)	0.7424	0.7424
A13	2631.76	0.49	0.11	0.03	0.7285(6,7,8)	0.7286	0.7286
A14	24716.34	600.98	144.91	4.88	0.5385(8)	0.5333 ^E	0.5385
A15	1279.93	7.24	0.54	0.16	0.6992(5,6,7)	0.6992 ^E	0.6992
A16	-	1156.23	125.62	4.24	-	0.5804	0.5804
A17	20840.55	87.13	42.32	3.84	0.5773(9)	0.5773 ^E	0.5773
A18	-	*	*	52810.10	-	*	0.4397
A19	1375608.66	23928.70	1771.99	249.52	0.5081(7)	0.5081	0.5081
A20	4830.00	1.78	14.55	0.09	0.7888(5,6,7)	0.7938 ^E	0.7888
A21	-	2145.24	305.48	22.60	-	0.5879 ^E	0.5860
A22	0.01	0.02	0.15	0.14	1.0000(7)	1.0000	1.0000
A23	42.29	10.08	0.44	0.14	0.8511(7)	0.8511	0.8511
A24	208158.02	17.46	0.78	0.20	0.7351(7)	0.7351	0.7351
A25	-	371233.00	48743.90	759.70	-	0.5329 ^E	0.5329
A26	-	*	*	134418.65	-	*	0.4895
A27	-	*	*	*	-	*	*
A28	-	*	*	46361.97	-	*	0.5482
A29	-	*	*	*	-	*	*
A30	-	183.71	41.53	8.00	-	0.6304 ^E	0.6331
A31	-	13807.50	2622.06	64.82	-	0.5977	0.5977
A32	-	*	*	234055.90	-	*	0.5084
A33	-	*	*	*	-	*	0.4829
A34	-	*	*	14212.57	-	*	0.6131
A35	-	325.53	18.22	1.61	-	0.8403	0.8403

Таблица 7: Testset A - Результаты сравнения подходов (зеротоны разрешены)

Так как Elbenani & Ferland [29] и Brusco [9] рассматривают либо одно фиксированное число ячеек, либо несколько вариантов число ячеек при сравнении с этими подходами указывается в скобках. Brusco [9] рассматривает несколько значений для возможного количества производственных ячеек, в этих случаях время

работы сравнивается с суммой времен для всех рассматриваемых вариантов количества ячеек. В качестве аппаратной платформы Brusco [9] заявляет систему с процессором Intel Core i7-2600 частотой 3.4 ГГц и 8GB оперативной памяти и Pinheiro et al. [53] такой же процессор Intel Core i7-2600 частотой 3.4 GHz с 32 GB оперативной памяти.

Результаты для набора данных Testset A отражены в Таблицах 6 и 7. Для каждого алгоритма представлено значение функции эффективности группировки (столбец ЭГ) на полученных оптимальных решениях. Время работы алгоритмов отражено в секундах. Так как используемый в данной работе набор данных чуть шире, чем набор используемый Brusco [9], отсутствующие результаты отмечены как ”-”.

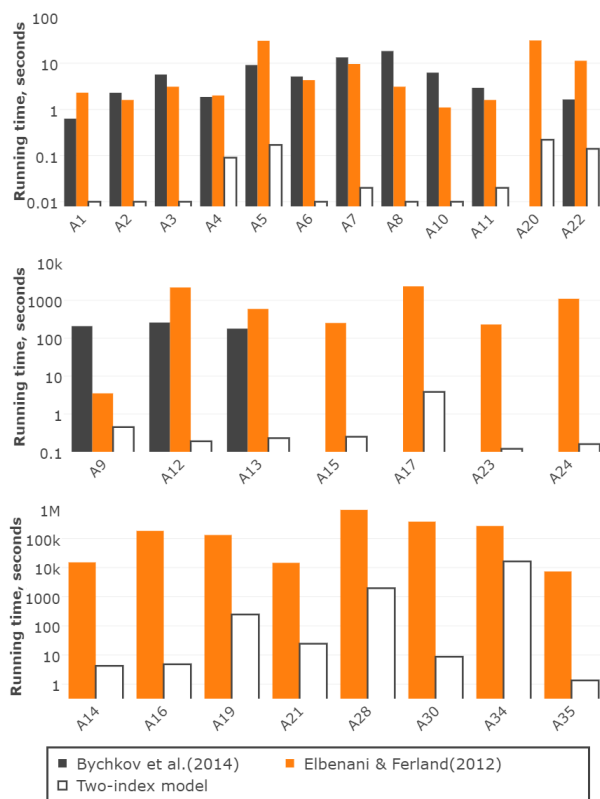


Рис. 2: Testset A - Сравнение времени работы алгоритмов. Формулировка без зеротонов.

Для нескольких тестовых задач глобальные оптимумы не были найдены, такие результаты помечены как ”*”. Таблица 6 показывает результаты для случая где зеротоны запрещены в решении. Подход, основанный на трехиндексной модели, также берет во внимание любое допустимое количество производственных ячеек,

но из-за большого количества переменных и ограничений способен решить лишь 14 задач из 35. Точный подход предложенный в Elbenani & Ferland [29] решает 27 тестовых задач, но только для одного фиксированного числа ячеек на каждом примере. Подход, основанный на двухиндексной модели находит глобальные оптимумы (с учетом любого возможного числа ячеек) для 31 задачи из 35. Для тестовой задачи A33 новым подходом было найдено решение со значением эффективности группировки 0.48 неизвестным ранее.

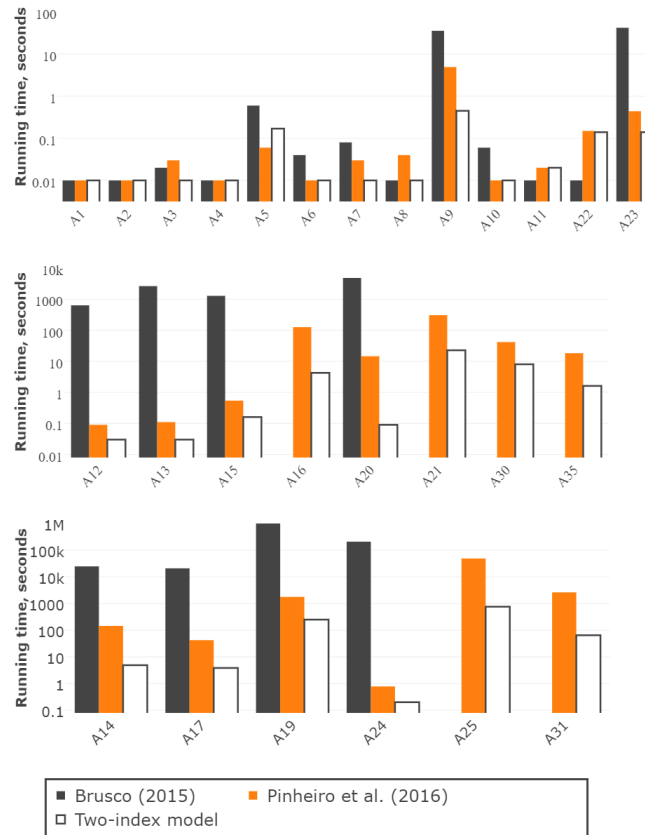


Рис. 3: Testset A - Разрешены любые размеры ячеек. Сравнение времени работы.

Для 17 тестовых задач A14-A21, A23-A26, A28, A30, A31, A34 и A35 представленный в диссертации алгоритм первым доказал глобальную оптимальность для существующих решений (принимая во внимание любое допустимое число ячеек) из различных источников, полученных в большей степени эвристическим путем.

Время работы алгоритмов из Таблицы 6 отражено на Рисунке 2. По этому показателю подход с двухиндексной моделью демонстрирует очень хорошие результаты. Он работает от 7 до 43383 раз быстрее чем подход Elbenani & Ferland [29] и от 11 до 1833 раз быстрее чем подход с трехиндексной моделью Vuchkov

et al. [15]. Нужно подчеркнуть, что несмотря на превосходство в плане аппаратного обеспечения над Elbenani & Ferland [29] двухиндексная модель решает более сложную и объемную задачу - найти оптимальное решение с учетом всех возможных количеств ячеек.

#	Время		Heuristic bound	ЭГ	
	two-index (no residual cells)	two-index (allow residual cells)		two-index (no residual cells)	two-index (allow residual cells)
B1	0.01	0.01	0.8095	0.8095	0.8095
B2	0.01	0.01	0.7222	0.7222	0.7222
B3	0.25	0.03	0.6071	0.6071	0.6071
B4	0.01	0.01	0.8889	0.8889	0.8889
B5	0.01	0.01	0.7500	0.7500	0.7500
B6	0.01	0.01	0.7391	0.7391	0.7391
B7	0.01	0.01	0.8148	0.8148	0.8148
B8	0.01	0.01	0.7222	0.7222	0.7222
B9	0.01	0.01	0.7576	0.7576	0.7576
B10	0.01	0.01	0.9000	0.9000	0.9000
B11	0.01	0.02	0.7273	0.7273	0.7297
B12	0.01	0.01	0.8276	0.8276	0.8276
B13	0.36	0.80	0.5962	0.5962	0.6042
B14	0.25	0.30	0.6405	0.6405	0.6405
B15	0.01	0.01	0.8333	0.8333	0.8333
B16	0.16	0.06	0.7391	0.7391	0.7444
B17	0.98	0.26	0.6552	0.6552	0.6842
B18	1.82	1.65	0.6027	0.6129	0.6129
B19	0.03	0.06	0.8000	0.8000	0.8113
B20	0.05	0.03	0.8710	0.8710	0.8710
B21	0.03	0.04	0.8333	0.8333	0.8333
B22	0.05	0.01	0.7258	0.7258	0.7258
B23	0.05	0.06	0.8111	0.8111	0.8111
B24	4.79	7.80	0.5673	0.5673	0.5728
B25	0.20	0.10	0.7600	0.7600	0.8000
B26	13.81	25.75	0.6068	0.6068	0.6078
B27	0.25	0.28	0.7248	0.7248	0.7248
B28	0.83	1.04	0.6729	0.6729	0.6729
B29	33.82	51.76	0.5730	0.5730	0.5745
B30	4.76	8.67	0.7308	0.7308	0.7325
B31	19.69	17.50	0.6799	0.6799	0.6799
B32	*	*	0.6193	*	*

Таблица 8: Testset B - Вычислительные эксперименты

Результаты для формулировки без ограничений на размеры ячеек проиллюстрированы в Таблице 7. Алгоритм, предложенный Pinheiro et al. [53] нашел оптимальные решения для 27 тестовых задач. Подход с двухиндексной моделью решил 32 задачи из 35, более того для тестовых задач A18, A33 и A34 были найдены решения ранее.

Так как тестовые задачи из набора Testset B менее популярны в литературе нашей целью было лишь получить оптимальные решения для этого набора. Для получения хороших начальных границ был использован оригинальный эвристический алгоритм поиска во многих окрестностях с рестартами из [14], который

описывается в следующих главах.

Результаты для набора Testset В показаны в Таблице 8. Оптимальные решения были найдены для 31 из 32 тестовых задач. Еще один результат - великолепная производительность эвристического алгоритма: лишь один решенный эвристическим алгоритмом тестовый пример отличается от найденного точного решения (пример В18). Из-за больших размеров представленным подходам не удалось получить точное решение для примера В32 (50×150).

4.4 Алгоритм поимка во многих окрестностях с рестартами

В настоящей главе будет представлен эвристический алгоритм для нахождения решений с высоким значением как целевой функции производительность группировки (grouping efficiency), так и эффективности группировки (grouping efficacy) для задачи CFP. Предлагаемый алгоритм итеративно применяет процедуру улучшения для случайно сгенерированных стартовых конфигураций ячеек. Вычислительные эксперименты показывают, что для целевой функции производительность группировки алгоритм превосходит имеющиеся лучшие решения из литературы на популярных наборах данных с тестовыми задачами размерами от 10×20 до 50×150 . Лучшие решения найдены в 23 из 24 случаев.

Ход предложенного эвристического алгоритма отражен в Algorithm 1.

Algorithm 1 Основная функция

```
function Solve( )
    FindOptimalCellRange(MinCells, MaxCells)
    ConfigsNumber = 2000
    AllConfigs = GenerateConfigs(MinCells, MaxCells, ConfigsNumber)
    return CMHeuristic(AllConfigs)
end function
```

Сначала вызывается функция FindOptimalCellRange, результатом выполнения которой является небольшой интервал с потенциально оптимальным значением количества ячеек - от *MinCells* до *MaxCells*. Данные значения и параметр *ConfigsNumber* - количество случайных конфигураций, которые будут сгенерированы передаются в дальнейшем в GenerateConfigs, где происходит само создание начальных конфигураций ячеек. Затем каждая конфигурация используется в функции CMHeuristic(*AllConfigs*), которая запускает процедуру поиска во многих окрестностях и пытается тем самым улучшить сгенерированное решение.

В реализации FindOptimalCellRange(*MinCells*, *MaxCells*) (Algorithm 2) перебираются все возможные количества ячеек от 2 до $\min(m, p)$. Для каждого числа ячеек в данном интервале генерируется фиксированное небольшое число конфигураций (для экспериментов использовалось эмпирически полученное значение 500), а затем проводится серия коротких тестовых запусков CMHeuristic(*Configs*). Главная задача этого этапа - найти потенциально перспективный небольшой интервал для общего числа ячеек в решении и в дальнейшем запустить на нем более тщательный поиск. Такой интервал выбирается относительно лучшего числа ячеек на коротких запусках - берем его и вместе с 10% окрестностями.

Algorithm 2 Процедура нахождения интервала для количества ячеек

```
function FindOptimalCellRange( MinCells, MaxCells)  
  if (m > p) then  
    minDimension = p  
  else  
    minDimension = m  
  end if  
  ConfigsNumber = 500  
  Configs = GenerateConfigs(2, minDimension, ConfigsNumber)  
  Solution = CMHeuristic(Configs)  
  BestCells = GetCellsNumber(Solution)  
  MinCells = BestCells - [minDimension * 0,1 ]      ▷ [ ] - integer part  
  MaxCells = BestCells + [minDimension * 0,1 ]  
end function
```

Algorithm 3 Configurations generation

```
function GenerateConfigs(MinCells, MaxCells, ConfigsNumber)  
  Configs =  $\emptyset$   
  for cells = MinCell, MaxCells do  
    Generated = GenerateConfigs(cells, ConfigsNumber)  
    Configs = Configs  $\cup$  Generated  
  return Configs  
end for  
end function
```

Функция $\text{GenerateConfigs}(\text{MinCells}, \text{MaxCells}, \text{ConfigsNumber})$ (Algorithm 3) возвращает множество конфигураций сгенерированных для текущих вариантов числа ячеек используя равномерное распределение. Конфигурация ячеек для k ячеек может быть представлена как разложение натуральных чисел m и p на суммы из k слагаемых. Такие конфигурации генерируются для каждого числа ячеек из интервала, а затем комбинируются в единый набор конфигураций.

Algorithm 4 CMHeuristic

```

function CMHeuristic(Configs)
    Best = 0
    for all config  $\in$  Configs do
        Solution = ImproveSolution(config)
        if Solution > Best then
            Best = Solution
        end if
    end for
    return Best
end function

```

CMHeuristic(Configs) (Algorithm 4) получает набор конфигураций в качестве входного параметра и для каждой запускает процедуру улучшения. Затем в качестве ответа возвращается решение с лучшим значением целевой функции.

Процедура улучшения решения основана на идее поиска во многих окрестностях. Используются две окрестности - перемещение станка в отличную от текущей ячейку и перенос детали. $\eta_{part, cell}$ это производительность группировки в текущем решении, где деталь с индексом $part$ перемещена в ячейку $cell$. Данная величина вычисляется для всех возможных пар деталь-ячейка. Пара с наибольшим увеличением по сравнению с текущим решением $\eta_{current}$ выбирается в качестве лучшего кандидата в окрестности деталей и значение изменения записывается в Δ_{parts} . Аналогичная процедура проводится со станками посредством сравнения двух значений Δ_{parts} и $\Delta_{machines}$ определяется изменение, которое будет применено к решению.

Вычислительные эксперименты для эвристического алгоритма показаны в Таблице 9. Ссылки на все тестовые задачи могут быть найдены в статье Vuchkov et al. [16]. Во всех экспериментах для определения интервала количеств ячеек использовался набор из 500 случайно сгенерированных конфигураций, а для полноценного поиска более широкий набор из 2000 конфигураций.

Алгоритм поиска во многих окрестностях с рестартами был запущен на 24 тестовых задачах из литературы. Размеры задач варьировались от 10×20 до 50×150 . Для каждой задачи производилось 50 запусков алгоритма и в качестве результата сохранялись данные о запуске с минимальным, максимальным и средним значением целевой функции производительность группировки. Данные значения сравнивались с лучшими известными решениями из Goldengorin et al. [30] и Bhatnagar & Saddikuti [6]. Новые глобальные оптимумы были найдены для 23 из 24 тестовых примеров. Для задачи 6 было найдено оптимальное решение с

Algorithm 5 Процедура улучшения решения

```
function ImproveSolution(config,  $\eta_{current}$ )
   $\eta_{current}$  = GroupingEfficiency(config)
  repeat
    PartFrom = 0
    PartTo = 0
    for part = 1, partsNumber do
      for cell = 1, cellsNumber do
        if ( $\eta_{part,cell} > \eta_{current}$ ) then
           $\Delta_{parts}$  = ( $\eta_{part,cell} - \eta_{current}$ )
          PartFrom = GetPartCell(part)
          PartTo = cell
        end if
      end for
    end for
    MachineFrom = 0
    MachineTo = 0
    for machine = 1, machinesNumber do
      for cell = 1, cellsNumber do
        if ( $\eta_{machine,cell} > \eta_{current}$ ) then
           $\Delta_{machines}$  = ( $\eta_{machine,cell} - \eta_{current}$ )
          MachineFrom = GetMachineCell(machine)
          MachineTo = cell
        end if
      end for
    end for
    if  $\Delta_{parts} > \Delta_{machines}$  then
      MovePart(PartFrom, PartTo)
    else
      MoveMachine(MachineFrom, MachineTo)
    end if
  until  $\Delta > 0$ 
end function
```

максимально возможным значением целевой функции как и в работе Goldengorin et al. [30].

5 Заключение

В текущей работе удалось добиться следующих важных результатов по рассмотренной задаче о формировании производственных ячеек:

1. Определение сложности статуса задачи CFP. Несмотря на долгую историю формального доказательства принадлежности оптимизационной версии задачи к классу NP-трудных и принадлежности задачи разрешимости к классу NP-полных до сих пор опубликовано не было. В текущей работе удалось доказать что задача разрешимости CFP с целевой функцией $E + V$ и целевой функцией эффективность группировки являются NP-полными, а оптимизационная версия CFP - NP-трудной.

2. Эффективные точные подходы. Лишь несколько авторов предложили точные подходы для ее решения в самой популярной постановке - с использованием дробной целевой функции эффективность группировки (grouping efficacy). Большая часть таких работ имела дополнительное ограничение в виде фиксированного числа производственных ячеек. В текущей работе предложено два точных подхода: трехиндексная модель с перебором значений знаменателя и двухиндексная модель с использованием алгоритма Динкельбаха. Первый подход на момент публикации стал первым, кому удалось доказать глобальную оптимальность части решений популярного набора данных 35 GT. Второй подход позволил найти глобальные оптимумы почти для всех задач этого набора, найти несколько новых решений и показать существенный выигрыш в производительности по сравнению с самыми быстрыми методами решения из литературы.

3. Эффективный эвристический алгоритм. Также удалось предложить эффективный алгоритм поиска во многих окрестностях с рестартами. При достаточно небольших вычислительных затратах алгоритм показал себя эффективным для целевой функции производительность группировки и нашел лучшие решения для 23 из 24 тестовых примеров.

Таблица 9: Вычислительные эксперименты для алгоритма поиска во многих окрестностях с рестартами

#	Source	mxp	Efficiency value			Time, sec	Cells	
			Bhatnagar & Saddikuti	Goldengorin et al.	Our			
			Min	Avg	Max			
1	Sandbothe (1998)	10x20	0.9540	0.9593 ^a	0.9566	0.9566	0.36	7
2	Ahi et al. (2009)	20x20	0.9262	0.9385	0.9599	0.9599	0.62	9
3	Mosier & Taube (1985)	20x20	0.8563	0.8871	0.9016	0.9022	0.88	9
4	Boe & Cheng (1991)	20x35	0.8831	0.8805	0.9334	0.9347	1.62	10
5	Carrie (1973)	20x35	0.9076	0.9564	0.9543	0.9578	1.54	10
6	Ahi et al. (2009)	20x51	0.8786	0.9411	0.9536	0.9545	3.1	12
7	Chandrasekharan & Rajagopalan (1989)	20x40	0.9882	1	1	1	1.8	7
8	Chandrasekharan & Rajagopalan (1989)	20x40	0.9533	0.9748	0.977	0.9775	2.42	12
9	Chandrasekharan & Rajagopalan (1989)	20x40	0.9378	0.9636	0.9684	0.9688	2.56	12
10	Chandrasekharan & Rajagopalan (1989)	20x40	0.8792	0.9432	0.9611	0.9616	3.3	15
11	Chandrasekharan & Rajagopalan (1989)	20x40	0.8495	0.9421	0.9594	0.9603	2.84	15
12	Chandrasekharan & Rajagopalan (1989)	20x40	0.8506	0.9232	0.9585	0.9595	2.76	15
13	Nair & Narendran (1996)	20x40	0.9644	0.9739	0.9778	0.9778	2.12	10
14	Nair & Narendran (1996)	20x40	0.9235	0.9574	0.974	0.974	2.2	14
15	Nair & Narendran (1996)	20x40	0.9325	0.9570	0.9581	0.9603	2.48	12
16	Nair & Narendran (1996)	20x40	0.9111	0.9640	0.9698	0.9698	2.78	14
17	Ahi et al. (2009)	25x40	0.9109	0.9552	0.9648	0.9648	2.58	14
18	Yang & Yang (2008)	28x35	0.9343	0.9382	0.9481	0.9485	2.46	10
19	Kumar & Vannelli (1987)	30x41	0.9066	0.9722	0.9738	0.9753	3.54	18
20	Stanfel (1985)	30x50	0.8817	0.9648	0.9677	0.9683	5.02	18
21	King & Nakomchai (1982)	30x90	0.8318	0.9462	0.9537	0.9584	13.1	25
22	Chandrasekharan & Rajagopalan (1987)	40x100	0.9475	0.9591	0.9806	0.9813	16.88	17
23	Yang & Yang (2008)	46x105	0.9098	0.9520	0.961	0.9618	23.9	18
24	Liang & Zolfaghari (1999)	50x150	0.9305	0.9292	0.9608	0.9617	51.66	24

^aОшибка в решении

6 Список литературы

- [1] Amit N. (2004). The bicluster graph editing problem. Master thesis. Tel Aviv University, 50 p.
- [2] Askin, R.G., Cresswell, S.H., Goldberg, J.B. and Vakharia, A.J., 1991. A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research*, 29(6), pp.1081-1100.
- [3] Ballakur A. (1985). An Investigation of Part Family/Machine Group Formation for Designing Cellular Manufacturing Systems. Ph.D. Thesis, University Wisconsin, Madison.
- [4] Ballakur, A., & Steudel, H. J. (1987). A within cell utilization based heuristic for designing cellular manufacturing systems. *International Journal of Production Research*, 25, 639–655.
- [5] Batsyn, M., Batsyna E., Bychkov I. (2019) NP-completeness of cell formation problem with grouping efficacy objective, *International Journal of Production Research (Q1)*, Published online: 26 Sep 2019, <https://doi.org/10.1080/00207543.2019.1668072>
- [6] Bhatnagar, R., V. Saddikuti. (2010). Models for cellular manufacturing systems design: matching processing requirements and operator capabilities. *J. Opl. Res. Soc.* 61, 827-839
- [7] Boutsinas, B. , 2013. Machine-part cell formation using biclustering. *Eur. J. Oper. Res.* 230, 563–572 .
- [8] Brusco, M. J. (2015). An iterated local search heuristic for cell formation. *Computers & Industrial Engineering*, 90, 292-304.
- [9] Brusco, M.J. (2015). An exact algorithm for maximizing grouping efficacy in part-machine clustering. *IIE Trans.* 47 (6), 653–671 .
- [10] Burbidge, J. L. (1961). The "new approach"to production. *Production Engineer*, 40(12), 769-784.
- [11] Burbidge, J. L. (1963). Production flow analysis. *Production Engineer*, 42(12), 742-752.
- [12] Burbidge, J. L. (1971). Production flow analysis. *Production Engineer*, 50(4.5), 139-152.
- [13] Burbidge, J. L. (1975). The introduction of group technology. Halsted Press.

- [14] Bychkov, I., Batsyn, M., Sukhov, P., Pardalos, P.M.(2013) Heuristic Algorithm for the Cell Formation Problem. In: Goldengorin B. I., Kalyagin V. A., Pardalos P. M. (eds.) Models, Algorithms, and Technologies for Network Analysis. Springer Proceedings in Mathematics & Statistics 59, 43–69.
- [15] Bychkov, I., Batsyn, M., Pardalos, P. (2014). Exact model for the cell formation problem. Optimization Letters 8(8), 2203–2210.
- [16] Bychkov I., Batsyn M., Pardalos P.M. Heuristic for Maximizing Grouping Efficiency in the Cell Formation Problem (2017), in: Models, Algorithms, and Technologies for Network Analysis. Springer Proceedings in Mathematics & Statistics / Ed. by V. A. Kalyagin, A. I. Nikolaev, P. M. Pardalos, O. Prokopyev . Vol. 197. Springer, 2017.
- [17] Bychkov, I. and Batsyn, M.(2018). An efficient exact model for the cell formation problem with a variable number of production cells. Computers & Operations Research, 91, pp.112-120.
- [18] Busygin S., Prokopyev, O., Pardalos, P. M. (2008). Biclustering in data mining. Computers Operations Research 35(9), 2964–2987
- [19] Busygin S., Prokopyev, O.A. and Pardalos, P.M., 2005. Feature selection for consistent biclustering via fractional 0–1 programming. Journal of Combinatorial Optimization, 10(1), pp.7-21.
- [20] Chan F. T. S., Lau K. W., Chan L. Y., Lo V. H. Y. (2008). Cell formation problem with consideration of both intracellular and intercellular movements. International Journal of Production Research 46(10), 2589-2620.
- [21] Chandrasekaran, M.P., Rajagopalan, R., (1986a). An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. International Journal of Production Research 24, 451-463.
- [22] Chandrasekaran, M.P., Rajagopalan, R., (1986b). MODROC: An extension of rank order clustering of group technology. International Journal of Production Research 24 (5), 1221-1233.
- [23] Chandrasekharan, M. P., & Rajagopalan, R. (1987). ZODIAC - An algorithm for concurrent formation of part families and machine cells. International Journal of Production Research, 25(6), 835-850.
- [24] Chandrasekharan, M. P., & Rajagopalan, R. (1989). Groupability: An analysis of the properties of binary data matrices for group technology. International Journal of Production Research , 27(6), 1035-1052.
- [25] Chung S.-H., Wu T.-H., Chang C.-C. (2011). An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. Computers & Industrial Engineering 60(1), 7-15.

- [26] Dinkelbach, W. , 1967. On nonlinear fractional programming. *Manage. Sci.* 13 (7), 492–498 .
- [27] Dimopoulos C., Zalzalá A. (2000). Recent developments in evolutionary computation for manufacturing optimization: problems, solutions and comparisons. *IEEE Transactions on Evolutionary Computation* 4(2), 93-113.
- [28] Dolgui, A. and Proth, J.M., 2010. *Supply chain engineering: useful methods and techniques*. Springer Science & Business Media.
- [29] Elbenani, B., Ferland, J. A. (2012). An exact method for solving the manufacturing cell formation problem. *International Journal of Production Research*, 50(15), 4038–4045.
- [30] Goldengorin, B., Krushinsky, D., Slomp, J. (2012). Flexible PMP approach for large size cell formation. *Operations Research*, 60(5), 1157–1166
- [31] Goldengorin B. , Krushinsky D., Pardalos P.M. *Cell formation in industrial engineering: theory, algorithms and experiments*, Springer Optimization and Its Applications, v.79, Springer-Verlag New York, 2013.
- [32] Gonçalves J.F., Resende M.G.C(2004). An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*, 47(2-3),247-273.
- [33] Hansen, P., Mladenović, N., Todosijević, R. and Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3), pp.423-454.
- [34] Hartigan, J.A.(1972). Direct clustering of a data matrix. *Journal of the American Statistical Association* 67, 123–129
- [35] Hartigan, J.A.(1975). *Clustering Algorithms*. Wiley, Chichester
- [36] James, T. L., Brown, E. C., Keeling, K. B. (2007). A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research*, 34(7), 2059–2079.
- [37] King, J. R. (1980). Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research* , 18(2), 213-232.
- [38] King, J. R., & Nakornchai, V. (1982). Machine-component group formation in group technology: Review and extension. *International Journal of Production Research* , 20(2), 117-133.
- [39] Krushinsky, D., & Goldengorin, B. (1984). An exact model for cell formation in group technology. *Computational Management Science*, 9(3), 323-338.

- [40] Kumar K. R., Kusiak A., Vannelli A. (1986). Grouping of parts and components in flexible manufacturing systems. *European Journal of Operations Research*, 24, 387-397.
- [41] Kumar K.R., Vannelli A. (1987). Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research*, 25(12), 1715-1728.
- [42] Kumar, K. R., Chandrasekharan, M. P. (1990). Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, 28(2), 233–243.
- [43] Kusiak, A. (1987). The generalized group technology concept. *International Journal of Production Research*, 25(4), 561–569.
- [44] Mak K. L., Wong Y. S., Wang X. X. (2000). An Adaptive Genetic Algorithm for Manufacturing Cell Formation. *The International Journal of Advanced Manufacturing Technology* 16(7), 491-497.
- [45] Mirkin, B. (1996). *Mathematical Classification and Clustering*, p. 448. Kluwer, Dordrecht.
- [46] Mirkin, B. (2011). Approximate bicluster and tricluster boxes in the analysis of binary data. *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, p 248-256, Springer, Berlin, Heidelberg
- [47] Mitrofanov S.P.(1959). *Nauchniye osnovy gruppovoi technologii*.
- [48] Mitrofanov, S.P.(1966). *The Scientific Principles of Group Technology*. Boston Spa, Yorkshire: National Lending Library Translation. Translation of Mitrofanov (1959).
- [49] Mucherino, A. and Cafieri, S., 2010. A new heuristic for feature selection by consistent biclustering. *arXiv preprint arXiv:1003.3279*.
- [50] Ng, S. (1993). Worst-case analysis of an algorithm for cellular manufacturing. *European Journal of Operational Research*, 69(3), 384–398.
- [51] Ng, S. (1996). On the characterization and measure of machine cells in group technology. *Operations Research*, 44(5), 735–744.
- [52] Paydar, M. M., Saidi-Mehrabad, M. (2013). A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy. *Computers & Operations Research*, 40(4), 980–990.
- [53] Pinheiro R. G. S., Martins I. C., Protti F., Ochi, L. S.; Simonetti L.G., & Subramanian A. (2016), On solving manufacturing cell formation via Bicluster Editing, *European Journal of Operational Research*, 254(3), 769 - 779
- [54] Sarker, B. R. (2001). Measures of grouping efficiency in cellular manufacturing systems. *European Journal of Operational Research*, 130(3), 588–611.

- [55] Solimanpur M., Saeedi S., Mahdavi I. (2010). Solving cell formation problem in cellular manufacturing using ant-colony-based optimization. *The International Journal of Advanced Manufacturing Technology* 50, 1135-1144.
- [56] Srinivasan, G., Narendran, T.T. (1991). GRAFICS—a nonhierarchical clustering algorithm for group technology. *The International Journal of Production Research*, 29(3), pp.463-478.
- [57] Tunnukij T., Hicks C. (2009). An Enhanced Grouping Genetic Algorithm for solving the cell formation problem. *International Journal of Production Research* 47(7), 1989-2007.
- [58] Utkina, I., Batsyn, M., Batsyna, E. (2016b). A branch and bound algorithm for a fractional 0-1 programming problem. *Lecture Notes in Computer Science*, 9869, 244–255.
- [59] Vannelli, A., Ravi Kumar, K., 1986. A method for finding minimal bottle-neck cells for grouping part-machine families. *International Journal of Production Research*, 24(2), pp.387-400.
- [60] Waghodekar, P. H., & Sahu, S. (1984). Machine-component cell formation in group technology MACE. *International Journal of Production Research*, 22, 937-948.
- [61] Wang, J., and Roze, C., 1997, Formation of machine cells and part families in cellular manufacturing: an experimental study. *International Journal of Production Research*, 35, 463-478
- [62] Won, Y. (2000). Two-phase approach to GT cell formation using efficient p-median formulations. *International Journal of Production Research*, 38(7), 1601–1613.
- [63] Zilinskas, J., Goldengorin, B., Pardalos, P. M. (2015). Pareto-optimal front of cell formation problem in group technology. *Journal of Global Optimization* 61(1), 91–108.
- [64] <http://opt-hub.com/problems/cfp>
- [65] https://www.researchgate.net/publication/316648108_Test_instances_and_solutions_for_the_cell_formation_problem