Моделирование

РАСПРЕДЕЛЕННЫЕ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА: ОБЗОР ЛИТЕРАТУРЫ И АЛГОРИТМЫ ОПТИМИЗАЦИИ РАБОТ НА ЗАВОДЕ[1]

Егорова Н.Г.,
кандидат технических наук, доцент,
Сотсков Ю.Н.,
доктор физико-математических наук, профессор,
Объединенный институт проблем информатики НАН Беларуси,
Шендерова Ю.А.,
Белорусский государственный университет

Введение. В статье рассматривается двухэтапная задача DJSSP оптимального планирования производственных работ в заводских цехах (это второй этап решения задачи DJSSP) после распределения поступивших заказов по заводам компании (первый этап решения задачи DJSSP).

На втором этапе решения задачи DJSSP требуется определить оптимальные последовательности производственных операций на основе ранее выбранного распределения заказов между однотипными заводами. Предполагается, что при составлении расписания работ в заводском цеху для каждой запланированной работы известны только нижняя граница и верхняя граница возможной длительности выполнения этой работы.

Поиск оптимального порядка выполнения работ на втором этапе решения задачи DJSSP выполняется в условиях неопределенности числовых параметров. Для решения задачи второго этапа в статье предложен эффективный эвристический алгоритм, основой которого является поиск такой перестановки работ в заводском цеху, которая имеет наибольший периметр многогранника оптимальности. Для оценки качества предложенного алгоритма и разработанных программ проведен вычислительный эксперимент на персональном компьютере.

Распределенные задачи календарного планирования работ на множестве заводов (обзор литературы и перспективы исследований)

Как следует из статьи (Mraihi, Driss, EL-Haouzi, 2024), одной из наиболее важных задач управления при принятии производственных решений по планированию производства на множестве заводов является задача оптимального распределения заказов между взаимосвязанными заводами и составление расписаний их реализации. Многие компании рассматривают планирование произ-

[1] Статья поддержана Белорусским Республиканским Фондом Фундаментальных Исследований, проект № Ф23РНФ-017.

водства как жизненно важное звено, которое передает поток заданий (заказов) между оперативным и тактическим уровнями управления внутри завода (предприятия). Функция календарного планирования подразумевает управление всеми доступными ресурсами (машинами, станками и рабочими) и разрешает возникающие производственные конфликты.

Календарное планирование – это раздел исследования операций, который включает оптимизацию использования ресурсов путем организации оптимальной последовательности операций, выделенных каждому ресурсу. Календарное планирование играет фундаментальную роль в планировании производственных систем, особенно в управлении производством товаров и услуг, управлении сложной производственной средой, обеспечивая эффективную координацию доступных ресурсов, материалов и производственных операций. Используя передовые алгоритмы и методы оптимизации, APS-система (т.е. система для пооперационного планирования производства) стремится к созданию эффективных решений для календарного и объемного планирования, которые минимизируют производственные затраты, максимизируют пропускную способность и обеспечивают своевременную доставку продукции потребителям (Liu, Wang, Chu, 2019; Lupeikiene, Dzemyda, Kiss, Caplinskas, 2014).

Далее будем рассматривать задачу с английским названием Distributed Job Shop Scheduling Problem (сокращенно – DJSSP) как обобщение классической задачи планирования работы цехов (английское сокращенное обозначение этой задачи – JSSP). Задача DJSSP состоит в одновременном назначении заданий различным однотипным заводам и их цехам и определении оптимальной последовательности их выполнения.

С развитием глобализации современных предприятий и совместных производств нескольких заводов производственная среда изменяется от традиционных однозаводских компаний к децентрализованным многозаводским компаниям и предприятиям, которые называют распределенным производством. Режим распределенного производства ближе к современному рынку с очевидными преимуществами с точки зрения более низких затрат на рабочую силу и более высокой эффективностью производства. Однако процессы планирования и составления расписаний для распределенных систем оказываются существенно сложнее по сравнению с производственной системой одного завода.

При решении задачи распределенного планирования необходимо учитывать производственные подзадачи как на каждом заводе компании, так и при общем планировании работ на множестве заводов одной компании. Поскольку эти подзадачи тесно связаны, разработка эффективных методов календарного планирования для заводов компании представляет собой очень сложную задачу. Тем не менее в последние годы планирование работ на одном заводе по-прежнему широко изучается, а вот объемное календарное планирование работ на множестве заводов анализируется сравнительно редко.

Обзор литературы по решению задачи DJSSP и ее частных случаев. В статье (Naderi, Ruiz, 2010) была впервые сформулирована перестановочная задача планирования распределенного потока работ (сокращенно – DPFSP) как частный случай задачи DJSSP, в которой работы могут иметь различные маршруты выполнения и порядок работ на разных машинах также может различаться. В статье (Naderi, Ruiz, 2010) сформулировано шесть различных моделей смешанного целочисленного линейного программирования для представления задачи DPFSP и ее решения.

В последние годы эвристики и метаэвристики широко используются для решения задачи DPFSP. Например, в терминах ряда эвристик, мотивированных хорошими показателями, в статье (Fernandez-Viagas, Framinan, 2015) представлен композиционный эвристический алгоритм, состоящий из так называемого итеративного разрушения и жадного алгоритма построения приближенного решения для задачи DPFSP. Для исследования были предложены и другие типы задач распределенного планирования.

Авторы статьи (Hsu, Kao, Ho, Lai, 2016) предложили так называемый агентный механизм переговоров, ориентированный на нечеткие ограничения для задачи распределенного планирования работы взаимосвязанных мастерских (сокращенно – DJSP).

В статье (Zhang, Li, Zhang, Wang, 2020) была разработана эвристика распределенных «колоний муравьев» для решения задачи гибкого планирования сборочных работ с несколькими заданными критериями оптимальности. В статье (Zheng, Wang, Wang, 2020) предложен кооперативный алгоритм коэволюции для многокритериальной задачи планирования нечетко распределенного гибридного потока (сокращенно – DHFSP). В этом алгоритме реализован поиск в режиме EDA (исследовательский анализ данных) и поиск в режиме IG (отношение качества) оптимального целочисленного ответа (по качеству) относительно реально оптимального решения задачи после ее релаксации. Были разработаны схема взаимодействия для переключения режимов на основе информационной энтропии и схема взаимодействия разнообразных эвристических решений задачи DHFSP.

В статье (Shao, Shao, Pi, 2020a) предложен алгоритм IG для оценки отношения качества оптимального целочисленного ответа к качеству оптимального реального решения релаксированной задачи DHFSP. В статье (Shao, Shao, Pi, 2020b) отмечается, что энергетические факторы также должны быть приняты во внимание, чтобы удовлетворить потребности «зеленого» производства.

В статье (Lu, Gao, Yi, Li, 2021) подчеркивается, что развитие мировой экономики традиционно связано с централизованным производством, которое не всегда удовлетворяет потребностям современного рынка быстро меняющегося производства. На этом фоне предлагается множество задач распределенного планирования работы цехов, в том числе цехов с распределенным потоком (Pan, Gao, Li, Wu, 2022a; 2022b; Shao, Shao, Pi, 2020a); планирование работы цехов с распределенным гибридным потоком (Shao, Shao, Pi, 2021); планирование работы цехов с распределенной перестановкой операций (Lu, Huang, Meng, Gao, Zhang, Zhou, 2022; Lu, Gao, Gong, Hu, Yan, Li, 2021); распределенное планирование работы сварочных цехов (Wang, Li, Gao, Li, 2021; Wang, Li, Gao, Li, 2022); распределенное гибкое планирование работы цехов (Meng, Ren, Zhang, Li, Sang, Zhang, 2020); распределенное планирование работы цехов (Jiang, Wang, Peng, 2020); распределенное параллельное планирование работы станков (Zhang, Deng, Zhao, Fan, Liu, Gong, 2022); распределенное групповое планирование работы цехов (Pan, Gao, Wang, 2020) и распределенное гибридное дифференциационное планирование работы цехов (Zhang , Liu, Wang, Yu, Xing, 2022).

Поскольку производственные затраты в таких крупномасштабных производственных системах, как процессы производства стали, огромны, то оптимизация календарных планов их эксплуатации очень важна с точки зрения снижения затрат и более эффективного использования ресурсов. В ряде статей отмечается, что существует практическая проблема с точки зрения расчетного времени, необходимого для получения оптимального производственного расписания работ и для оптимизации всего производственного процесса.

В ряде статей упоминается: даже если получено оптимальное расписание для всего производственного процесса, уложиться в построенное расписание крайне сложно из-за различий, имеющихся между моделью производственной системы и реальной системой. Также существуют различные переменные и неопределенные факторы, которые трудно учитывать или исключить при реальной эксплуатации производственной системы.

Таким образом, даже если производственный план построен как ориентир для планируемого производства, реальные проблемы оптимизации производственного процесса, как правило, остаются нерешенными. В результате могут быть созданы избыточные рабочие места или наоборот – возникнет нехватка рабочих мест или материала в «узком месте» производственного процесса, остановки оборудования из-за переполнения мест хранения промежуточной продукции (буфера), задержки в доставке комплектующих или дефицит производственных мощностей. С другой стороны, производственная среда существенно улучшается за счет внедрения интернета вещей, в которой информация об оборудовании и логистике на заводе может быть получена в режиме реального времени. Кроме того, на многих предприятиях внедрена производственная среда, в которой инструкции по эксплуатации оборудования и транспортировке изделий могут передаваться непосредственно на используемое для этого оборудование в режиме реального времени.

В статье (Wu, Bai, Chen, Lin, Xing, Lin, Cheng, 2021) сделан вывод о том, что большинство практических производственных сред содержит значительное количество неопределенных факторов и неточных параметров. В классических исследованиях по теории расписаний такие параметры, как время обработки задания, сроки его выполнения, время поступления задания в обслуживающую систему и другие числовые параметры задачи, считаются определенными и фиксированными как в процессе составления расписаний, так и в процессе реализации расписаний. При этом предполагается, что производственные ресурсы всегда доступны. Однако перечисленные предположения не соответствуют реальности многих современных производственных систем, поскольку, как правило, на практике существуют различные виды неопределенностей. Например, рабочие могут стать нетрудоспособны в какие-то моменты времени, машины или станки могут ломаться, рабочая среда может изменяться с течением времени. В реальном производстве возможны случайные поступления новых заданий. Длительности обработки деталей могут быть неизвестны на этапе построения расписаний. Некоторые другие внешние неопределенные факторы могут привести к прерыванию каких-то работ, изменению состава оборудования или снижению качества используемых инструментов.

В статье (Wu, Bai, Chen, Lin, Xing, Lin, Cheng, 2021) отмечается, что ввиду наличия существенных неопределенностей разумно рассматривать время выполнения заданий и другие параметры, связанные с заданиями (например, сроки их выполнения), как случайные параметры. Машины (станки) также подвержены случайным поломкам. По указанным причинам календарному планированию в условиях неопределенности в последние годы стали уделять существенно больше внимания в литературе по исследованию операций.

Большинство исследователей исходят из неких вероятностных распределений для случайных неопределенных факторов и пытаются достичь оптимальности решений в среднем смысле. Однако в действительности сроки обработки деталей часто оцениваются на основе статистических данных. Дисперсия этих данных может быть слишком большой, а базовые распределения вероятностей, полученные на основе статистических данных, могут быть неточными. Более того, довольно часто наихудшая производительность системы оказывается важнее, чем ее средняя производительность. Учитывая эти ситуации, робастный подход во многих случаях является более приемлемым для хеджирования производственных процессов от наихудшего развития событий (Kouvelis, Yu, 1997). Во многих практических случаях неопределенность может быть представлена как часть сценариев, каждый из которых обусловливает свою возможную реализацию параметров заданий (Aloulou, Della Croce, 2008; Aissi, Aloulou, Kovalyov, 2011; Yang, Yu, 2002; Kasperski, Zielinski, 2016).

Как отмечается в статье (Mraihi, Driss, EL-Haouzi, 2024), в связи с глобализацией традиционные производственные системы перешли от обслуживающих систем для одного завода к децентрализованным обслуживающим системам для нескольких однотипных заводов. Задача планирования Flow-shop стала первой в создании простых программ определения оптимальных последовательностей операций с заранее определенными ресурсами, что достигается за счет обмена информацией между ресурсами, заводами, заданиями и машинами, которые сформировали задачу DPFSP-планирования распределенного потока заданий.

В настоящее время автоматизация стала неотъемлемой частью практических производственных систем, при этом многие этапы производства стали автоматизированными. Роль человека на производстве превратилась в контроль производственных операций вместо ручного их выполнения. Функции работника стали ключевым фактором организации и контроля производственных процессов, что необходимо учитывать при назначении работников на рабочие места – так же, как возможности и эффективность их обучения. В статье (Mraihi, Driss, EL-Haouzi, 2024) представлен систематический обзор научной литературы по задаче DPFSP-планирования распределенного потока заданий с оптимизацией последовательностей выполнения работ в заводском цеху с гибкостью работников для квалификации и с определением того, какую важную роль играет человеческий фактор в оптимальном планировании работ. Представлена задача планирования работы цеха с распределенными перестановками выполняемых работ и «гибкостью» квалифицированных работников. В статье (Pan, Gao, Wang, 2020) рассматривается проблема планирования как задача группового планирования распределенных производственных цехов, которая имеет важные приложения в современных производственных системах. В такой задаче определяется возможность организации множества рабочих мест с учетом групповых ограничений на ряде идентичных производственных цехов. Каждый цех имеет специальную структуру с целью минимизации времени изготовления продукции. Авторы статьи (Pan, Gao, Wang, 2020) исследуют вопросы, относящиеся к конкретной задаче. Представлена смешанная модель целочисленного линейного программирования, описан парадокс, противоречащий интуиции, а также две процедуры для ускорения представленного алгоритма и для экономии вычислительных ресурсов и усилий. Из-за сложности задачи рассматривается стратегия декомпозиции и предлагается кооперативный коэволюционный алгоритм (сокращенно – CCEA) для модели совместной работы и схемы повторной инициализации алгоритма. Проводится комплексная и тщательная вычислительная и статистическая оценка проведенного вычислительного эксперимента. Полученные результаты показали, что предлагаемая модель совместной работы и схема повторной инициализации достаточно эффективны. Алгоритм CCEA превосходил ряд эвристик при решении классических задач теории расписаний.

Задачи оптимального планирования относятся к классу NP-трудных задач. В последние годы наблюдается значительное развитие методов оптимизации в многоагентных системах. В статье (Toshiyuki, Toyohiro, Shigemasa, 2020) рассматривается крупномасштабная обслуживающая система как многоагентная система, обсуждаются методы решения задачи оптимального планирования с использованием консенсуса между агентами. Предложен распределенный алгоритм, использующий процедуру переменного направления множителей. Алгоритм протестирован на небольшом примере задачи оптимального планирования производства.

Распределенное гибкое планирование работ привлекло интерес исследователей в связи с развитием глобального производства. Тем не менее ограничения, включающие транспортировку краном и потребление энергии, должны также рассматриваться исходя из реалистичных требований. Для решения этой более сложной проблемы авторы статьи (Du, Li, Luo, Meng, 2021) смоделировали задачу с помощью метода целочисленного линейного программирования, при котором продолжительность производства и потребление энергии во время движения машины и транспортировки краном оптимизируются одновременно. После этого для решения задачи был предложен гибридный алгоритм, состоящий из алгоритма оценки распределения (сокращенно – EDA) и поиска переменной окрестности (сокращенно – VNS), в котором было разработано правило идентификации четырех условий крана, чтобы сделать расчет пригодности осуществимым. В компоненте САПР параметры в вероятностных матрицах настроены на самоадаптацию для стабильной сходимости и для получения лучшего результата. Кроме того, был применен вероятностный механизм для управления активностью компонентов САПР. В компоненте VNS используются пять проблемно-ориентированных структур соседства, включая глобальные и локальные стратегии, для повышения эксплуатационных возможностей. Результаты имитационных испытаний подтвердили, что предложенный гибридный алгоритм EDA-VNS может решать рассматриваемую задачу с более высокой эффективностью по сравнению с другими конкурирующими алгоритмами, а предложенные стратегии совершенствования имеют большое значение для повышении производительности.

Многие практические постановки задач календарного планировании производства полны существенных неопределенностей. Например, может измениться рабочая среда, могут выйти из строя станки или машины, работники могут стать нетрудоспособными и т.д. В такой среде время обработки задания не может быть заранее известным и фиксированным числом. В свете такой ситуации авторы статьи (Wu, Bai, Chen, Lin, Xing, Lin, Cheng, 2021) исследовали задачу с одним процессором и двумя допустимыми значениями длительностей обработки заданий на основе определенных сценариев. Цель упорядочения заданий для обработки их на процессоре состояла в том, чтобы свести к минимуму максимальное общее время завершения обработки заданий в двух допустимых сценариях.

Когда возникает неопределенность времени обработки задания, робастная версия этой задачи становится NP-трудной, даже для очень ограниченных частных случаев задачи. Для ее решения выводятся некоторые правила доминирования и нижняя граница для разработки алгоритмов ветвей и границ для поиска оптимальных решений. Что касается определения приближенных решений, авторы статьи (Wu, Bai, Chen, Lin, Xing, Lin, Cheng, 2021) предложили пять эвристик, использующих комбинированное время обработки деталей на основе двух заданных сценариев при получении начальных решений и последующего улучшения каждого из них с помощью попарной взаимозаменяемости. Кроме того, авторы предложили эвристическую схему имитационного отжига, включающую семь простых эвристик для приближенного решения этой задачи. В статье (Wu, Bai, Chen, Lin, Xing, Lin, Cheng, 2021) представлены результаты вычислительных экспериментов для оценки эффективности предложенных алгоритмов.

В статье (Hsu, Kao, Ho, Lai, 2016) представлен алгоритм согласования нечетких ограничений на основе агентов (AFCN) для решения задач планирования распределенного цеха (DJSSP). Задача планирования моделируется как набор задач, удовлетворяющих нечетким ограничениям (FCSP), связанным межагентскими ограничениями. Каждая задача FCSP отражает точку зрения участника и управляется автономными агентами. Новизна предлагаемого алгоритма AFCN заключается в том, что концепция нечеткой функции принадлежности используется для представления неточных предпочтений времени начала выполнения задания для агентов в соответствии с доступными ресурсами. Дополнительный обмен информацией имеет решающее значение для эффективности распределенной координации. Это не только ускоряет сходимость алгоритма, но и обеспечивает глобальную согласованность посредством повторяющегося обмена агентов встречными предложениями. Алгоритм AFCN может применять такие стратегии ведения переговоров агентов, как конкурентные, взаимовыгодные и совместные для различных производственных сред. Результаты проведенного вычислительного эксперимента демонстрируют, что предложенная модель может обеспечить не только качественное и экономически эффективное планирование работы цеха (сопоставимое с централизованными алгоритмами), но и высокую производительность с позиций сроков выполнения и среднего времени обработки потока заданий по сравнению с другими моделями согласования для планирования производства на основе агентов. Предложенный алгоритм AFCN гибок и полезен для распределенного планирования производства с непредвиденными нарушениями и неопределенностями производственных процессов.

В статье (Zhang, Li, Zhang, Wang, 2020) исследуется проблема планирования производства в гибкой производственной системе с двумя смежными рабочими зонами, продукция которых объединена гибкими нелинейными технологическими планами и сборочными операциями. Основные детали производятся на одном участке, а затем транспортируются на другой участок для сборки конечных изделий. Монтажные конструкции изделий бывают как плоскими, так и многоуровневыми. Время настройки операций в зависимости от последовательности и время перехода заданий между станками рассматриваются отдельно от времени их обработки. Потоковая передача лотов учитывается заранее, чтобы каждое задание представляло собой базовую деталь, а не партию идентичных деталей. Идентичные узлы сборки являются общими для всех возможных операций сборки, а не предварительно связанными с каким-либо продуктом. Длина расписания, общее запаздывание изготовления продукции и общая рабочая нагрузка считаются целевыми функциями, которые необходимо минимизировать.

Авторы статьи (Zhang, Li, Zhang, Wang, 2020) предлагают распределенную систему «колоний муравьев» для решения проблемы и исследования фронта Парето. Подход сначала сравнивается с другими методами; при этом используются несколько наборов гипотетических тест-кейсов с разными размерами и сложностью. Затем метод применяется для решения задачи планирования производства шаровых кранов при различных возможных сценариях. Показано, что предложенный подход превосходит по качеству решений большинство других методов для тестируемых задач, особенно для примеров большой размерности, что делает его конкурентоспособным при решения практических задач оптимального планирования производства.

Задача построения расписания выполнения работ в заводском цеху в условиях неопределенности продолжительности работ

При планировании работ в заводском цеху сложно заранее определить точное время, которое потребуется для выполнения той или иной планируемой работы. Между тем можно заранее дать вполне реалистичную оценку временного интервала, который будет заведомо содержать длительность выполнения запланированной работы. В силу этого на этапе составления расписания будем предполагать, что для каждой работы $J_i \in G$ из заданного множества работ $G = \{J_1, J_2, ..., J_n\}$ известны нижняя граница $p_i^L > 0$ и верхняя граница $p_i^U \geq p_i^L$ длительности ее выполнения. Пусть для каждой работы $J_i \in G$ задан вес (ее приоритет) $w_i > 0$, характеризующий важность более раннего завершения работы J_i.

Если множество работ на горизонте планирования не меняется (иными словами, в процессе выполнения работ множества $G=\{J_1, J_2, ..., J_n\}$ в цех не поступают новые работы и не изменяется само множество $G=\{J_1, J_2, ..., J_n\}$ запланированных работ), то такую задачу можно интерпретировать как задачу $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ составления оптимального расписания (т.е. перестановки всех запланированных работ) для одного станка с интервальными длительностями выполнения работ и критерием $\sum w_iC_i$ минимизации суммы взвешенных моментов завершения выполнения работ множества $G=\{J_1, J_2, ..., J_n\}$.

В соответствии с введенной в статье (Graham, Lawler, Lenstra, Rinnooy Kan, 1979) трехпозиционной формой $\alpha|\beta|\gamma$ классификации всех задач теории расписаний, позиция α характеризует обслуживающую систему и определяет число станков, позиция β определяет характеристики выполняемых работ, а в позиции γ указывается целевая функция, которая определяет оптимальность искомого расписания выполнения запланированных работ.

В качестве критерия оптимальности будем рассматривать минимизацию суммы $\sum_{i=1}^{n} w_iC_i$ взвешенных моментов C_i завершения выполнения работ $J_i \in G$. Такую задачу можно интерпретировать и как задачу максимизации суммарной прибыли цеха (Сотсков, Егорова, Матвейчук, 2020). Действительно, пусть выполнение работы $J_i \in G$ приносит прибыль, оцениваемую величиной $H - w_iC_i$, где H – это достаточно большое число. Тогда чем раньше будет выполнена работа, тем больше будет полученная заводом прибыль. Как показано в статье (Сотсков, Егорова, Матвейчук, 2020), задача минимизации величины $\sum_{i=1}^{n} w_iC_i$ равносильна задаче максимизации суммарной прибыли $\sum_{i=1}^{n}(H - w_iC_i) = nH - \sum_{i=1}^{n} w_iC_i$, поскольку произведение nH не зависит от порядка выполнения работ множества G.

В задаче $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ предполагается, что каждая работа $J_i \in G$ должна быть выполнена без прерываний за время $p_i \in [p_i^L, p_i^U]$, которое становится известным в момент завершения этой работы. Если границы допустимых длительностей выполнения каждой работы $J_i \in G$ совпадают, т.е. $p_i^L = p_i^U$, то полученная детерминированная задача $1||\sum w_iC_i$ может быть решена за время $O(n\log n)$, как было доказано в статье (Smith, 1956).

Для неопределенной задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ в общем случае не существует такой перестановки работ из множества S всех $n!$ перестановок выполнения работ множества G, которая гарантированно являлась бы оптимальной при всех фиксированных длительностях $p=(p_1, p_2, ..., p_n)$ выполнения работ множества G (т.е. при всех сценариях из заданного множества сценариев $T=\{p=(p_1, p_2, ..., p_n): p \in R_+^n: p_i^L \le p_i \le p_i^U, i \in \{1,2,...,n\}\}$). Здесь R_+^n обозначает множество неотрицательных действительных векторов размерности n.

В связи с этим в качестве близкого к оптимальному эвристического решения задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ в статье (Егорова, Сотсков, Вернер, 2017) было предложено использовать эффективную перестановку π_k с максимальным значением $PerOB(\pi_k)$ относительного периметра многогранника оптимальности.

Построенные с помощью приведенного в статье (Lai, Sotskov, Egorova, Werner, 2018) алгоритма эффективные перестановки можно использовать для решения задачи планирования на длительный период времени (Сотсков, Егорова, Матвейчук, 2020). Перестановка $\pi_e = (\pi_{e_1}, \pi_{e_2}, ..., \pi_{e_p})$ выполнения работ в заданном интервале планирования определяется как конкатенация частичных перестановок π_{e_k} выполнения работ, запланированных на *k*-й период горизонта планирования. Под периодом горизонта планирования понимается временной промежуток, в течение которого множество запланированных работ не меняется.

В свою очередь перестановка π_k выполнения работ, запланированных на *k*-й период времени, может включать только те работы, которые поступили для выполнения на момент составления расписания. Иными словами, перестановка π_k включает работы, поступившие в *k*-й период времени для последующего их выполнения, а также работы, поступившие и не выполненные в предыдущие периоды планирования.

В качестве периода планирования при построении эффективной перестановки работ в проведенном вычислительном эксперименте рассматривался один рабочий день. В течение *k*-го рабочего дня все работы выполнялись в соответствии с расписанием (т.е. в соответствии с перестановкой π_k) до тех пор, пока начало s_{k_r} выполнения очередной работы J_{k_r}, согласно перестановке $\pi_k = (J_{k_1}, J_{k_2}, ..., J_{k_d})$, не выходило за пределы рабочего времени *k*-го рабочего дня.

Алгоритмы построения перестановки работ с наибольшим периметром многогранника оптимальности

Детерминированную задачу $1||\sum w_i C_i$ с фиксированным сценарием $p \in T$ обозначим $1|p|\sum w_i C_i$. Пусть $M = (k_{i_1}, k_{i_2}, ..., k_{i_{|M|}})$, $k_{i_1} < k_{i_2} < ... < k_{i_{|M|}}$, обозначает упорядоченное подмножество множества натуральных чисел $\{1, 2, ..., n\}$.

Определение 1. Максимальный (по включению) параллелепипед $OB(\pi_k, T) = [l^*_{k_{i_1}}, u^*_{k_{i_1}}] \times [l^*_{k_{i_2}}, u^*_{k_{i_2}}] \times ... \times [l^*_{k_{i_{|M|}}}, u^*_{k_{i_{|M|}}}] \subseteq T$ будем называть многогранником оптимальности перестановки $\pi_k = (J_{k_1}, J_{k_2}, ..., J_{k_n}) \in S$ работ из множества $G = \{J_1, J_2, ..., J_n\}$, если для любого сценария $p = (p_1, p_2, ..., p_n) \in T$, при котором перестановка π_k является оптимальной для задачи $1|p|\sum w_i C_i$, эта перестановка π_k остается оптимальной и для задачи $1|p'|\sum w_i C_i$, $p' \in [p_1, p_1] \times [p_2, p_2] \times ... \times [p_{i_g - 1}, p_{i_g - 1}] \times [l^*_{i_g}, u^*_{i_g}] \times [p_{i_g + 1}, p_{i_g + 1}] \times ... \times [p_n, p_n]$, для любого индекса $i_g \in M = (k_{i_1}, k_{i_2}, ..., k_{i_{|M|}})$. Если же не существует сценария $p \in T$, при котором перестановка π_k оптимальна для задачи $1|p|\sum w_i C_i$, то имеет место равенство $OB(\pi_k, T) = \varnothing$.

Непустой отрезок $[l^*_{k_i}, u^*_{k_i}]$, $l^*_{k_i} < u^*_{k_i}$, будем называть отрезком оптимальности работы J_{k_i} в перестановке π_k. Если не существует непустого отрезка $[l^*_{k_i}, u^*_{k_i}]$ для работы $J_{k_i} \in J$ в перестановке π_k, то работа $J_{k_i} \in J$ не имеет отрезка оптимальности в перестановке π_k.

В качестве эвристического решения задачи $1|p_i^L \le p_i \le p_i^U|\sum w_i C_i$ будем использовать перестановку π_k с максимальным взвешенным периметром $PerOB(\pi_k)$ многогранника оптимальности. Величина $PerOB(\pi_k)$ взвешенного периметра многогранника оптимальности перестановки π_k

равна сумме отрезков оптимальности $[u^*_{k_i}/w_{k_i}, l^*_{k_i}/w_{k_i}]$ для всех работ $J_{k_i} \in J$, для которых имеется непустой отрезок оптимальности в перестановке π_k. Перестановку $\pi_k \in S$ с максимальным значением $PerOB(\pi_k)$ будем называть эффективной перестановкой для задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$.

В статье (Lai, Sotskov, Egorova, Werner, 2018) были доказаны свойства многогранника оптимальности $OB(\pi_k, T)$ и на основе доказанных свойств разработан метод динамического программирования для построения эффективной перестановки с использованием следующих понятий блоков и работ, не фиксированных в блоке.

Определение 2. Блоком будем называть максимальное подмножество $B_r = \{J_{r_1}, J_{r_2}, ..., J_{r_{|B_r|}}\} \subseteq G$ множества работ G, для которых выполняется соотношение $\min_{J_{r_i} \in B_r} \left\{ \frac{w_{r_i}}{p^L_{r_i}} \right\} \ge \max_{J_{r_i} \in B_r} \left\{ \frac{w_{r_i}}{p^U_{r_i}} \right\}$.

Отрезок $[b_r^L, b_r^U]$, для которого справедливы равенства $b_r^L = \max_{J_{r_i} \in B_r} \left\{ \frac{w_{r_i}}{p^U_{r_i}} \right\}$ и $b_r^U = \min_{J_{r_i} \in B_r} \left\{ \frac{w_{r_i}}{p^L_{r_i}} \right\}$, будем называть ядром блока B_r.

Работы, принадлежащие одному и тому же блоку, будем называть фиксированными в блоке, а принадлежащие нескольким блокам – нефиксированными.

В разработанных в статье (Lai, Sotskov, Egorova, Werner, 2018) алгоритмах сложности $O(n\log n)$ для нахождения всех блоков B для задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ предполагается, что блоки $\{B_1, B_2, ..., B_m\} = B$ пронумерованы в порядке убывания левых границ их ядер. После выделения блоков в статье (Lai, Sotskov, Egorova, Werner, 2018) предлагается выполнить декомпозицию исходной задачи на несколько подзадач.

Для декомпозиции задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ достаточно выполнить $O(n)$ операций. Результирующая перестановка π_k является конкатенацией перестановок, полученных при решении всех подзадач, полученных после декомпозиции исходной задачи.

При решении задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$ с одним блоком эффективную перестановку $\pi_k \in S$ с максимальным значением $PerOB(\pi_k)$ можно построить за время $O(n)$. При построении эффективной перестановки для задачи $1|p_i^L \le p_i \le p_i^U|\sum w_iC_i$, когда работы содержатся в нескольких блоках и необходимая декомпозиция множества работ невозможна, в статье (Егорова, Сотсков, 2018) предложен алгоритм динамического программирования (алгоритм ДП).

Алгоритм ДП строит дерево решений высотой $m = |B|$, в котором ранг вершины соответствует номеру анализируемого блока работ. Пусть j обозначает номер вершины в дереве решений, B_r – это номер блока, а G^j – это множество нефиксированных работ, принадлежащих блоку B_r. На каждой итерации t алгоритма ДП для каждой висячей вершины $\pi^j(B_r, G^j) \in V$ дерева решений, выбранной из множества уже построенных вершин, строится множество всех потомков вершины $\pi^j(B_r, G^j) \in V$ следующим образом.

Пусть B^t_{r+1} обозначает множество всех работ, фиксированных в блоке B_{r+1}, а также нефиксированных работ блока B_{r+1}, не распределенных в свои блоки на предыдущих итерациях алгоритма ДП. Рассмотрим множество $G[B^t_{r+1}] \subseteq B^t_{r+1}$, состоящее из нефиксированных работ, принадлежащих блоку B_{r+2}. На следующих итерациях работы множества $G[B^t_{r+1}]$ могут быть распределены как в блок B_{r+1}, так и в один из следующих за блоком B_{r+1} блоков. Поэтому для таких работ строится множество всех его подмножеств (т.е. булеан $P(G[B^t_{r+1}])$ множества $G[B^t_{r+1}]$) $P(G[B^t_{r+1}]) = \{\varnothing, ..., G[B^t_{r+1}]\}$, которые порождают $2^{|G[B^t_{r+1}]|}$ вершин дерева решений (это потомки вершины $\pi^j(B_r, G^j) \in V$ в дереве решений).

Пусть G_i, $|G_i| = m_i$, обозначает множество нефиксированных работ, распределенных в блок B_{r+1} в вершине $\pi^i(B_{r+1}, G^i)$, $G^i \in P(G[B^t_{r+1}])$ дерева решений, а также фиксированных работ в блоке B_{r+1}. В каждой построенной вершине $\pi^i(B_{r+1}, G^i)$, $G^i \in P(G[B^t_{r+1}])$, дерева решений упорядочим работы множества G_i и вычислим для упорядоченных работ $G_i' = (J_{i_1}, J_{i_2}, ..., J_{i_{m_i-1}}, J_{i_{m_i}})$ значение штрафа $F(\pi^i)$. Величина штрафа $F(\pi^i)$ определяет разницу между вычисленным значением относительного периметра многогранника оптимальности в построенной перестановке работ и максимально возможным значением относительного периметра многогранника оптимальности.

Очевидно, что вершины $\pi^j(B_r, G^j)$ дерева решений с одинаковыми номерами блока B_r и наборами фиксированных работ в блоке B_r и предшествующих блоках, для которых выполняется неравенство $\frac{p^U_{j_{m_j}}}{w_{j_{m_j}}} \le \min_{J_i \in B_{r+1}} \left\{ \frac{p^L_i}{w_i} \right\}$, определяют одно и то же состояние в пространстве решений задачи $1 | p^L_i \le p_i \le p^U_i | \sum w_i C_i$.

Из множества висячих вершин построенного дерева решений для дальнейшего ветвления выбирается вершина $\pi^i(B_{r+1}, G^i)$ с наименьшим штрафом $F(\pi^i)$. Если таких вершин окажется несколько, выбирается висячая вершина $\pi^i(B_{r+1}, G^i)$ с наименьшим штрафом $F(\pi^i)$ из построенного дерева решений с наибольшим номером $r+1$ блока B_{r+1}.

Условия окончания работы алгоритма ДП следующие: (1) должна быть построена перестановка, содержащая все работы множества G; (2) для построенной перестановки работ множества G должен быть получен минимальный штраф $F(\pi^i)$ среди штрафов еще нерассмотренных (висячих) вершин построенного дерева решений.

Если оба условия (1) и (2) выполнены, следует восстановить искомую эффективную перестановку обратным ходом по построенному дереву решений.

Были проведены вычислительные эксперименты на персональном компьютере для оценки эффективности описанных выше алгоритмов. В вычислительном эксперименте моделировался 30-дневный период составления расписаний. Проведенные вычислительные эксперименты показали эффективность такого подхода при составлении расписаний для задачи $1 | p^L_i \le p_i \le p^U_i | \sum w_i C_i$ (Сотсков, Егорова, Матвейчук, 2020).

Модификация метода динамического программирования

При ветвлении вершины $\pi^j(B_r, G^j) \in V$ вместо всех $2^{|G[B^t_{r+1}]|}$ подмножеств множества G^j незафиксированных ранее нефиксированных работ в этой вершине предлагается сформировать не более $1 + C^1_{|G[B^t_{r+1}]|} + C^2_{|G[B^t_{r+1}]|} + 2 \cdot C^3_{|G[B^t_{r+1}]|}$ подмножеств. Такое действие возможно в силу того, что на величину относительного периметра многогранника оптимальности $PerOB(\pi_k)$ влияют только работы в первой, второй, предпоследней ($m_i - 1$) и последней (m_i) позиции частичной перестановки работ в каждом из блоков множества $B = \{B_1, B_2, ..., B_m\}$ (Lai, Sotskov, Egorova, Werner, 2018). Следовательно, нефиксированные в вершине $\pi^i(B_{r+1}, G^i)$ работы после их упорядочения можно разделить на два класса: зафиксированные работы (они влияют на многогранник оптимальности, т.е. при упорядочивании оказались в одной из четырех указанных выше позиций) и «плавающие» работы (они не влияют на многогранник оптимальности). «Плавающие работы» либо могут быть распределены в последующие блоки, либо могут оставаться в любом из блоков, которым они принадлежат.

После нахождения наибольшего многогранника оптимальности при построении результирующей перестановки с наибольшим многогранником оптимальности такие нераспределенные работы можно распределить в любой из блоков, в котором они являются «плавающими». Незафиксированная ранее работа множества G^i будет «плавающей», если для нее выполняются следующие два условия: (3) в дереве решений существует вершина $\pi^p(B_q, G^p)$, $q < r+1$, в которой эта работа принадлежит соответствующему этой вершине блоку B_q; (4) в вершине $\pi^p(B_q, G^p)$ мощность объединения множества фиксированных работ и множества зафиксированных в этой вершине работ – не менее четырех. Такую работу можно возвратить в блок B_q в вершине $\pi^p(B_q, G^p)$.

Поскольку последняя работа J_{i_m} блока B_{r+1} удовлетворяет условию $J_{i_m} \notin B_{r+2}$, то на величину относительного периметра многогранника оптимальности $PerOB(\pi_k)$ может влиять не более трех незафиксированных ранее работ, принадлежащих блоку B_{r+1} и не принадлежащих блоку B_{r+2}. Поскольку на многогранник оптимальности влияют только работы, которые при упорядочении в частичной перестановке $G_i' = (J_{i_1}, J_{i_2}, ..., J_{i_{m_i-1}}, J_{i_{m_i}})$ оказались в одной из четырех позиций, то в дереве решений в каждой вершине $\pi^i(B_{r+1}, G^i)$ достаточно сохранять только те нефиксированные работы, которые оказались в одной из этих четырех указанных выше позиций.

При этом для каждого множества зафиксированных в вершине $\pi^i(B_{r+1}, G^i)$ работ, которые ранее были нефиксированными, содержащих плавающие в этой вершине работы, в дереве решений найдется вершина $\pi^q(B_{r+1}, G^q)$, $|G^q| = m_q$, не содержащая эти плавающие работы. Нетрудно убедиться в том, что в первой, второй, предпоследней ($m_i - 1$) и последней (m_i) позициях в вершине $\pi^i(B_{r+1}, G^i)$ и в первой, второй, предпоследней ($m_q - 1$) и последней (m_q) позициях в вершине $\pi^q(B_{r+1}, G^q)$ окажется одинаковое множество работ. В силу этого для дальнейшего рассмотрения можно оставить только одну вершину $\pi^q(B_{r+1}, G^q)$. Более того, порядок работ в четырех указанных выше позициях будет тоже одинаковым. Следовательно, вершины с одинаковым набором нефиксированных работ в рассмотренных четырех позициях определяют одно и то же состояние процесса решения задачи. Поэтому в дерево решений следует добавлять вершины только в том случае, если в нем еще нет вершин с уже найденным набором зафиксированных работ.

Сохранение в дереве решений нефиксированных работ в порядке их следования в перестановке ускорит сравнение этих множеств. Если в вершине $\pi^i(B_{r+1}, G^i)$ мощность объединения множества фиксированных работ и множества зафиксированных в ней работ не более четырех, то такую вершину можно добавить в дерево решений без дополнительных сравнений вершин дерева решений.

С целью сокращения количества рассмотренных ветвей при построении дерева решений можно вместо фиксирования всех подмножеств плавающих работ мощностью не более четырех рассматривать только те подмножества, в которых количество нефиксированных работ, принадлежащих блоку B_{r+1} и не принадлежащих блоку B_{r+2}, не более трех. В противном случае либо одна из них будет плавающей вершиной, либо все такие вершины будут образовывать «виртуальный» блок. Предложенная модификация алгоритма ДП позволила сократить время построения эффективной перестановки и, следовательно, сократить время составления расписаний при планировании работ завода.

План вычислительного эксперимента. Целью вычислительного эксперимента было сравнение времени построения перестановки выполнения запланированных работ множества $G=\{J_1, J_2, ..., J_n\}$ на основе построенных перестановок с наибольшим относительным периметром многогранника оптимальности при условии, что вместо опубликованного ранее метода динамического программирования (Сотсков, Егорова, Матвейчук, 2020) используется модифицированный метод динамического программирования. В вычислительном эксперименте моделировался 30-дневный период составления расписаний на каждый из дней горизонта планирования. Для каждого рабочего дня были заданы его продолжительность и количество вновь поступивших работ. Схема формирования множества работ для выполнения в *k*-й рабочий день та же, что и в статье (Сотсков, Егорова, Матвейчук, 2020).

В проведенном на персональном компьютере вычислительном эксперименте из частичных перестановок $\pi_k=(J_{k_1}, J_{k_2}, ..., J_{k_r}, ..., J_{k_{10}})$, $\pi_h=(J_{h_1}, J_{h_2}, ..., J_{h_u}, ..., J_{h_{10+|G(1)|}})$, $\pi_t=(J_{t_1}, J_{t_2}, ..., J_{t_v}, ..., J_{t_{10+|G(2)|}})$, ... выполнения работ в каждый день из интервала планирования строилась перестановка $\pi_e=(J_{k_1}, J_{k_2}, \ldots, J_{k_r}, J_{h_1}, J_{h_2}, \ldots, J_{h_u}, J_{t_1}, J_{t_2}, \ldots, J_{t_v}, \ldots)$ выполнения всех работ в течение всех дней рассматриваемого горизонта планирования.

При этом фиксировалось время $t_{исх}$ построения перестановки π_e при условии, что для построения частичных перестановок π_k, π_h, π_t, ... использовался описанный в статье (Сотсков, Егорова, Матвейчук, 2020) алгоритм, а также время $t_{модерн}$ получения перестановки π_e при условии, что для построения частичных перестановок π_k, π_h, π_t, ... использовался описанный модифицированный алгоритм ДП.

Результаты вычислительного эксперимента. В вычислительном эксперименте были сгенерированы и решены приближенно серии примеров, содержащих случайное число k нефиксированных работ, $(n-k)$ фиксированных работ и случайное количество блоков работ. Целые веса работ выбирались равновероятно из отрезка [1, 10]. Фиксированные работы распределялись поровну между m блоками работ. Генерация фиксированных работ, содержащихся в блоке B_j, проводилась следующим образом.

По заданному ядру блока $[b_j^L, b_j^U]$ генерировались отрезки $[\bar{p}_i^L, \bar{p}_i^U]\supset[b_j^L, b_j^U]$, $\bar{p}_i^L\in[b_j^L-D, b_j^L]$, $\bar{p}_i^U\in[b_j^U, b_j^U-D]$, $D\in\{2,3,5,7,10,15\}$, по которым определялись следующие отрезки допустимых длительностей выполнения работ: $[p_i^L, p_i^U]=[w_i\bar{p}_i^L, w_i\bar{p}_i^U]$.

Генерация нефиксированных работ производилась аналогичным образом. Для заданного отрезка $[\bar{b}_i^L, \bar{b}_i^U]\supset[b_i^L, b_m^U]$ генерировались отрезки $[\bar{p}_i^L, \bar{p}_i^U]\supset[\bar{b}_i^L, \bar{b}_i^U]$, $\bar{p}_i^L\in[\bar{b}_j^L-D, \bar{b}_j^L]$, $\bar{p}_i^U\in[\bar{b}_j^U, \bar{b}_j^U-D]$, $D\in\{1,2,3,4,5,10\}$, по которым определялись следующие отрезки допустимых длительностей выполнения работ: $[p_i^L, p_i^U]=[w_i\bar{p}_i^L, w_i\bar{p}_i^U]$. Каждая серия состояла из 10 примеров задачи $1|p_i^L\le p_i\le p_i^U|\sum w_iC_i$. В таблице (см. таблицу) представлены результаты вычислительного эксперимента. Серии задач в вычислительном эксперименте отличаются одна от другой количеством фиксированных работ и количеством блоков.

Поскольку не существует различий между алгоритмом ДП и модифицированным алгоритмом динамического программирования в количестве нефиксированных работ, которые были зафиксированы в каждой вершине дерева решений, то различия по времени решения одинаковых примеров могут возникать, только если в вершине рассматривается не менее пяти нефиксированных работ.

В эксперименте для каждой задачи суммарно генерировалось 300 работ. Поскольку каждый день поступает на выполнение 10 новых работ, причем они случайно выбираются из 300 уже сгенерированных работ, то при общем количестве нефиксированных работ менее 50 вероятность появления более пяти нефиксированных требований в подзадаче дневного планирования будет небольшой. Вследствие этого время решения таких задач модифицированным алгоритмом ДП практически не уменьшалось. Если же количество нефиксированных работ превышало 70, то немодифицированный алгоритм ДП не справлялся с построением расписания для однодневных задач $1|p_i^L\le p_i\le p_i^U|\sum w_iC_i$ в разумные сроки.

Таблица

Время решения задач алгоритмом ДП и модифицированным алгоритмом динамического программирования

Количество нефиксированных работ	Количество блоков	Среднее время решения задач алгоритмом ДП (сек), $t_{исх}$	Среднее время решения задач модифицированным алгоритмом (сек), $t_{модерн}$	Значение отношения $t_{исх} / t_{модерн}$
50	2	22,9513468	21,7334041	1,05604
60	2	23,9101514	19,9741454	1,197055
65	2	47,3619990	21,5972379	2,192966
70	2	> 900	24,5065172	>36,72492
50	3	22,8582041	22,6008109	1,011389
60	3	44,8727562	21,6313606	2,074431
65	3	43,3271949	21,9040574	1,978044
70	3	> 900	26,7315266	>33,66811
50	4	35,7220989	18,5920702	1,921362
60	4	55,7881936	24,4355724	2,283073
65	4	121,6912888	24,5681774	4,953208
70	4	> 900	26,0561913	>34,54074
Среднее значение				>10,30011

Источник: авторская разработка.

В проведенном вычислительном эксперименте время решения немодифицированным алгоритмом некоторых задач одного рабочего дня превышало 15 мин. При этом время работы модифицированного алгоритма ДП изменялось незначительно.

Перспективы исследования задач DJSSP и $1 | p_i^L \le p_i \le p_i^U | \sum w_i C_i$. В дальнейшем представляется перспективным исследовать задачу DJSSP, ее различные частные случаи и методы решения задач $1 | p_i^L \le p_i \le p_i^U | \sum w_i C_i$, которые возникают после распределения поступивших заказов на заводы компании. Поскольку задача DJSSP является NP-трудной в сильном смысле, то практический интерес представляет выделение таких частных случаев задачи DJSSP, которые являются полиномиально разрешимыми.

Представляет интерес проведение компьютерных экспериментов с задачами DJSSP большой размерности и исследование алгоритмов, обеспечивающих сходимость процессов решения к оптимальным расписаниям. В общем случае задачи DJSSP важно определять степень оптимальности получаемых приближенных расписаний.

В дальнейшем предлагается исследовать обобщения алгоритмов оптимизации с декомпозицией исходной задачи на подзадачи для однокритериальных и многокритериальных задач оптимального планирования. Следует рассматривать сочетание различных целей оптимизации производственных процессов, разрабатывать правила, основанные на знаниях для конкретных задач DJSSP, и включать эти правила в эвристические алгоритмы поиска решений на основе популяций для дальнейшего расширения возможностей оптимизации расписаний для задач DJSSP. Предлагается исследовать задачи DJSSP с неопределенными числовыми параметрами, как это сделано для задачи $1 | p_i^L \le p_i \le p_i^U | \sum w_i C_i$, возникающей на последнем этапе решения общей задачи DJSSP после распределения поступивших заказов по заводам компании.

Для практических приложений задач DJSSP целесообразно рассматривать два критерия оптимальности решений. Первый критерий (например, минимизация длины расписания) следует рассматривать для подзадачи первого этапа решения задачи DJSSP, который связан с оптимальным распределением заказов по однотипным заводам компании. Второй критерий (например, критерий $\sum w_i C_i$) целесообразно рассматривать для задачи второго этапа решения общей задачи DJSSP.

Этот этап связан с построением оптимального расписания выполнения заказов на конкретном заводе компании, поскольку для завода более важной является прибыль, а не длина построенного расписания.

Практический интерес представляет исследование задачи DJSSP, которую представим, используя трехпозиционное обозначение $\alpha|\beta|\gamma$, как это принято с 1979 года (Graham, Lawler, Lenstra, Rinnooy Kan, 1979) для классификации всех задач, относящихся к теории расписаний. Ранее в публикациях по задаче DJSSP не было введено трехпозиционное обозначение этой задачи, поэтому обозначим такую систему обслуживания как DfJ. Это сокращение английского названия обслуживающей системы Distributed Job-shop, где буква *f* определяет число фабрик (заводов) компании. Получаем, таким образом, обозначение $DfJm|p_i^L \le p_i \le p_i^U|\sum_{i=1}^{f}\sum w_iC_i$ для задачи DJSSP, которую рекомендуем исследовать в дальнейшем. Буква *m* в обозначении $DfJm|p_i^L \le p_i \le p_i^U|\sum_{i=1}^{f}\sum w_iC_i$ определяет максимальное число машин (станков), которые имеются на одной из фабрик.

Критерий $\sum_{i=1}^{f}\sum w_iC_i$ определяет максимизацию общей прибыли компании, которая складывается из прибылей всех ее фабрик (заводов). Из неравенств $p_i^L \le p_i \le p_i^U$ следует, что для каждой работы $J_i \in G$, которые требуется распределить между *f* фабриками на момент решения задачи $DfJm|p_i^L \le p_i \le p_i^U|\sum_{i=1}^{f}\sum w_iC_i$, известны только нижняя граница и верхняя граница возможной продолжительности выполнения этой работы.

Заключение. Статья содержит обзор публикаций по решению задачи DJSSP и ее частных случаев. Показано, что DJSSP – важная задача управления при принятии оптимальных решений планирования производства на заводах компании. В задаче DJSSP требуется оптимально распределить заказы между однотипными заводами и составить расписание реализации заказов на заводах компании. На втором этапе задачи DJSSP требуется решить задачи $1|p_i^L \le p_i \le p_i^U|\sum C_i$ для каждого завода, получившего заказы.

Исследована задача $1|p_i^L \le p_i \le p_i^U|\sum C_i$ составления расписания выполнения множества работ с неопределенными длительностями с целью минимизации взвешенной суммы моментов завершения выполнения всех работ. Для задачи $1|p_i^L \le p_i \le p_i^U|\sum C_i$ рассматривается область оптимальности перестановки выполнения всех работ. В статье модифицирован алгоритм ДП динамического программирования для поиска перестановки $\pi_k \in S$ выполнения работ с наибольшим взвешенным периметром многогранника оптимальности. Предложенная модернизация алгоритма ДП существенно сокращает время составления расписаний при планировании работ заводского цеха.

В дальнейшем предлагается исследовать задачу $DfJm|p_i^L \le p_i \le p_i^U|\sum_{i=1}^{f}\sum w_iC_i$. Приводится обоснование необходимости решения распределенных задач теории расписаний.

Литература

Егорова Н.Г., Сотсков Ю.Н. 2018. Минимизация суммарного времени обслуживания требований с неопределенными длительностями операций. *Экономика, моделирование, прогнозирование*. Сборник научных трудов. Минск: НИЭИ Министерства экономики Республики Беларусь. Выпуск 12. С. 121–131.

Сотсков Ю.Н., Егорова Н.Г., Вернер Ф. 2017. Перестановка с максимальным параллелепипедом оптимальности обслуживания требований с интервальными длительностями. *Экономика, моделирование, прогнозирование*. Сборник научных трудов. Минск: НИЭИ Министерства экономики Республики Беларусь. Выпуск 11. С. 104–116.

Сотсков Ю.Н., Егорова Н.Г., Матвейчук Н.М. 2020. Алгоритмы планирования рабочего времени в условиях неопределенности. *Информатика*. Т. 17. № 2. С. 86–102.

Aissi H., Aloulou M.A., Kovalyov M.Y. 2011. Minimizing the number of late jobs on a single machine under due date uncertainty. *Journal of Scheduling*. Vol. 14. No 4. P. 351–360.

Aloulou M.A., Della Croce F. 2008. Complexity of single machine scheduling problems under scenario-based uncertainty. *Operations Research Letters*. Vol. 36. No 3. P. 338–342.

Graham R.E., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*. Vol. 5. P. 287–326.

Du Y., Li J.-Q., Luo C., Meng L.-l. 2021. A hybrid estimation of distribution algorithm for distributed flexible job shop scheduling with crane transportations. *Swarm and Evolutionary Computation*. Vol. 62. No 100861. P. 1–24.

Fernandez-Viagas V., Framinan J. M. 2015. A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research*. Vol. 53. P. 1111–1123.

Hsu C., Kao B., Ho V.L., Lai K.R. 2016. Agent-based fuzzy constraint-directed negotiation mechanism for distributed job shop scheduling. *Engineering Applications of Artificial Intelligence*. Vol. 53. P. 140–154.

Jiang E.-D., Wang L., Peng Z.-P. 2020. Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition. *Swarm and Evolutionary Computation*. Vol. 58. No 100745. P. 1–16.

Jin X., Liang G., Quan-ke P., Tasgetiren M.F. 2019. An effective multi-objective artificial bee colony algorithm for energy efficient distributed job shop scheduling. *25-th International Conference on Production Research Manufacturing Innovation*: Cyber Physical Manufacturing. August 9–14, 2019. Chicago, Illinois (USA). Vol. 39. P. 1194–1203.

Kasperski A., Zielinski P. 2016. Robust discrete optimization under discrete and interval uncertainty: a survey. *Robustness analysis in decision aiding, optimization, and analytics*. Springer, Cham. P. 113–143.

Kouvelis P., Yu G. 1997. *Robust discrete optimization and its applications*. Springer New York, NY. 358 p.

Lai T.-C., Sotskov Y.N., Egorova N.G., Werner F. 2018. The optimality box in uncertain data for minimizing the sum of the weighted job completion times. *International Journal of Production Research*. Vol. 56. No 19. P. 6336–6362.

Liu J.L., Wang L.C., Chu P.C. 2019. Development of a cloud-based advanced planning and scheduling system for automotive parts manufacturing industry. *Procedia Manufacturing*. Vol. 38. P. 1532–1539.

Lu C., Gao L., Gong W., Hu C., Yan X., Li X. 2021. Sustainable scheduling of distributed permutation flow-shop with non-identical factory using a knowledge-based multiobjective memetic optimization algorithm. *Swarm and Evolutionary Computation*. Vol. 60. No 100803.

Lu C., Gao L., Yi J., Li X. 2021. Energy-efficient scheduling of distributed flow shop with heterogeneous factories: a real-world case from automobile industry in China. *IEEE Transactions on Industrial Informatics*. Vol. 17. No 10. P. 6687–6696.

Lu C., Huang Y., Meng L., Gao L., Zhang B., Zhou J. 2022. A Pareto-based collaborative multi-objective optimization algorithm for energy-efficient scheduling of distributed permutation flow-shop with limited buffers *Robotics and Computer-Integrated Manufacturing*. Vol. 74. No 102277.

Lupeikiene A., Dzemyda G., Kiss F., Caplinskas A. 2014. Advanced planning and scheduling systems: Modeling and implementation challenges. I*nformatica*. Vol. 25. P. 581–616.

Meng L., Ren Y., Zhang B., Li J.Q., Sang H., Zhang C. 2020. MILP modeling and optimization of energy-efficient distributed flexible job shop scheduling problem. I*EEE Access*. Vol. 8. P. 191191–191203.

Mraihi T., Driss O.B., EL-Haouzi H.B. 2024. Distributed permutation flow shop scheduling problem with worker flexibility: Review, trends and model proposition. *Expert Systems with Applications*. Vol. 238. No 121947. P. 1–13.

Naderi B., Ruiz R. 2010. The distributed permutation flowshop scheduling problem. *Computers & Operations Research*. Vol. 37. P. 754–768.

Pan Y., Gao K., Li Z., Wu N. 2022a. Improved meta-heuristics for solving distributed lot-streaming permutation flow shop scheduling problems. *IEEE Transactions on Automation Science and Engineering*. P. 1–11.

Pan Y., Gao K., Li Z., Wu N. 2022b. Solving biobjective distributed flow-shop scheduling problems with lot-streaming using an improved Jaya algorithm. *IEEE Transactions on Cybernetics*. P. 1–11.

Pan Q.-K., Gao L., Wang L. 2020. An effective cooperative co-evolutionary algorithm for distributed flowshop group scheduling problems. *IEEE Transactions on Cybernetics*. Vol. 52. No 7. P. 1–14.

Rui Li, Wenyin G., Ling W., Chao Lu, Shuning J. 2022. Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm and Evolutionary Computation*. Vol. 74. No 101139. P. 1–12.

Shao Z., Shao W., Pi D. 2020a. Effective heuristics and metaheuristics for the distributed fuzzy blocking flow-shop scheduling problem. *Swarm and Evolutionary Computation*. Vol. 59. No 100747.

Shao W., Shao Z., Pi D. 2020b. Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem. *Knowledge-Based Systems*. Vol. 194. No 105527.

Shao W., Shao Z., Pi D. 2021. Multi-objective evolutionary algorithm based on multiple neighborhoods local search for multi-objective distributed hybrid flow shop scheduling problem. *Expert Systems with Applications*. Vol. 183. No 115453.

Smith W.E. 1956. Various optimizers for single-stage production. *Naval Research and Logistics Quarterly*. Vol. 3. No 1. P. 59–66.

Toshiyuki M., Toyohiro U., Shigemasa T. 2020. Distributed job shop scheduling using consensus alternating direction method of multipliers. *IFAC Papers OnLine*. Vol. 53. No 2. P. 10785–10790.

Wang G., Li X., Gao L., Li P. 2022. An effective multi-objective whale swarm algorithm for energy-efficient scheduling of distributed welding flow shop. *Annals of Operations Research*. Vol. 310. No 1. P. 223–255.

Wang G., Li X., Gao L., Li P. 2021. Energy-efficient distributed heterogeneous welding flow shop scheduling problem using a modified MOEA/D. *Swarm and Evolutionary Computation*. Vol. 62. No 100858.

Wu C.-C., Bai D., Chen J.-H., Lin W.-C., Xing L., Lin J.-C., Cheng S.-R. 2021. Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times. *Swarm and Evolutionary Computation*. Vol. 60. No 100765. P. 1–17.

Yang J., Yu G. 2002. On the robust single machine scheduling problem. *Journal of Combinatorial Optimization*. Vol. 6. No 1. P. 17–33.

Zhang L., Deng Q., Zhao Y., Fan Q., Liu X., Gong G. 2022. Joint optimization of demand- side operational utility and manufacture-side energy consumption in a distributed parallel machine environment. *Computers and Industrial Engineering*. Vol. 164. No 107863.

Zhang G., Liu B., Wang L., Yu D., Xing K. 2022. Distributed co-evolutionary memetic algorithm for distributed hybrid differentiation flowshop scheduling problem. *IEEE Transactions on Evolutionary Computation*. Vol. 26. P. 1043–1057.

Zhang S., Li X., Zhang B., Wang S. 2020. Multi-objective optimization in flexible assembly job shop scheduling using a distributed ant colony system. *European Journal of Operational Research*. Vol. 283. No 2. P. 441–460.

Zheng J., Wang L., Wang J. 2020. A cooperative convolution algorithm for multi-objective fuzzy distributed hybrid flow shop. *Knowledge-Based Systems*. Vol. 194. No 105536. P. 1–11.

Статья поступила 02.02.2024 г.