Hindawi

Математические проблемы в технике

Том 2018, ID статьи 3124182, 10 страниц

<https://doi.org/10.1155/2018/3124182>

**Исследовательская статья**

**Планирование пакетной обработки с использованием алгоритма Max – Min Ant System, улучшенного методом локального поиска**

и

1) Школа горного дела, Китайский горно-технический университет, Сюйчжоу 221116, Китай

2) Школа менеджмента, Университет науки и технологий Китая, Хэфэй, 230026, Китай

Переписка должна быть адресована YuWang; YuWangUSTC@126.com

Получено 4 сентября 2017 года; Пересмотрено 13 декабря 2017 года; Принято 1 января 2018 года; Опубликовано 28 января 2018 г.

Академический редактор: Fiorenzo A. Fazzolari

Copyright © 2018 Сяолинь Ли и Ю Ван. Статья в открытом доступе, распространяемая в соответствии с указаниями Creative Commons.

Лицензия, которая разрешает неограниченное использование, распространение и воспроизведение на любом носителе, при условии, что оригинальное произведение правильно процитировано.

В этой статье рассматривается проблема минимизации рабочего времени (рабочего цикла) на одной машине пакетной обработки. Объем задания и время обработки не идентичны, а время обработки каждого пакета определяется по заданию с наибольшим временем обработки в этом пакете. Для решения этой проблемы разработан алгоритм Max – Min Ant System (MMAS). Для повышения производительности алгоритма предлагается локальный метод поиска MJE (Multiple Jobs Exchange), регулирующий задания между пакетами. Для определения параметров MMAS проводится предварительный эксперимент. Производительность предлагаемого алгоритма MMAS посредством численного эксперимента сравнивается с CPLEX, а также с несколько другими алгоритмами, включая алгоритм муравьиного цикла (AC), генетический алгоритм (GA) и две эвристики, первое время обработки с наибольшим соответствием (FFLPT) и наилучшее время наибольшей обработки (BFLPT). Результаты эксперимента показывают, что MMAS превзошел других, особенно для большой численности населения.

**1. Введение**

Планирование машины пакетной обработки является типичной задачей комбинаторной оптимизации. В отличие от традиционных задач планирования, машина пакетной обработки может одновременно обрабатывать несколько заданий. Планирование периодической обработки машины обычно встречается в обрабатывающей промышленности, такой как термообработка в металлургии и экранирование напряжений при производстве интегральных схем. Поскольку эти операции часто являются узким местом в последовательности производства, планирование процесса пакетной обработки будет эффективно продвигать время выполнения работ.

Впервые эта проблема была предложена Икурой и Гимплом [1], которые изучали проблему с одинаковыми размерами заданий и постоянным временем пакетной обработки, а производительность машины определялась количеством одновременно обрабатываемых заданий. Рассматривая операцию горения в производстве полупроводников, Ли и другие [2] представили эффективные алгоритмы на основе динамического программирования для минимизации ряда различных показателей производительности. Эта же проблема была исследована Сунгом и Чоуном [3], которые предложили алгоритм ветвления и связывания, а также несколько эвристик с целью минимизации рабочего времени (рабочего цикла).

Проблема гораздо сложнее, когда рассматриваются неидентичные размеры заданий. Узсой [4] исследовал проблему составления графика обработки отдельных пакетов с целью минимизации рабочего времени (рабочего цикла) (𝐶max) и общего времени обработки (ΣCi) при неидентичных размерах заданий. Обе проблемы оказались NP-трудными, и было предложено несколько эвристических методов, включая время первичной обработки с наибольшим временем обработки (FFLPT), кратчайшее время обработки с первым соответствием (FFSPT) и др. Дюпон и Джолай Газвини [5] предложили две эффективные эвристические задачи: последовательная задача о ранце (SKP) и наилучшее время обработки (BFLPT), а затем одна превзошла FFLPT. Для оптимального решения задачи точный алгоритм, такой как ветвь и граница, был предложен Дюпоном и Даененсом-Флипо [6], они предоставли некоторые свойства доминирования для общей схемы перечисления для критерия рабочего времени (рабочего цикла). Схема нумерации [7] также была разработана в сочетании с существующей эвристикой для решения крупномасштабных задач.

Поскольку задача планирования машины пакетной обработки является NP-трудной [4], для ее решения были разработаны различные метаэвристические алгоритмы. Мелук и соавт. [8] изучили проблему с использованием имитации отжига (SA), и случайные экземпляры были сгенерированы для оценки эффективности алгоритма. Та же проблема была рассмотрена Дамодараном и др. [9] с генетическим алгоритмом (GA). Эксперимент показал, что GA превосходил SA по времени выполнения и качеству решения. Хуссейнзаде Кашан и др. [10] предложили групповую версию алгоритма оптимизации роя частиц (PSO), применение которого было сделано для задачи планирования с одной машиной. Подход GRASP, разработанный Дамодараном и др. [11], был использован для минимизации рабочего времени (цикла) емкостной машины периодической обработки, и экспериментальное исследование пришло к выводу, что GRASP превзошел другие подходы к решению. Для задач, связанных с мультимашинами, Чжоу и др. [12] предложили эффективный дифференциально-эволюционный гибридный алгоритм, чтобы минимизировать продолжительность работы (рабочего цикла) на машинах с параллельной обработкой, и алгоритм был оценен путем сравнения с генетическим алгоритмом случайных ключей (RKGA) и алгоритмом оптимизации роя частиц (PSO). Подобная проблема была изучена Цзян и др. [13] с учетом партийных перевозок. Для решения этой проблемы предлагается гибридный алгоритм, сочетающий преимущества оптимизации роя дискретных частиц (DPSO) и генетического алгоритма (GA). Производительность предложенных алгоритмов была улучшена за счет использования локальной стратегии поиска. Все эти исследования показывают эффективность решения проблем машин пакетной обработки с использованием метаэвристических алгоритмов.

Исследования, рассмотренные выше, в основном решают эту проблему, упорядочивая задания в списке заданий, а затем группируя задания по пакетам. В отличие от существующих исследований, метаэвристический алгоритм MMAS (Max – Min Ant System) был разработан конструктивно, объединяя эти два этапа принятия решений вместе. То есть пакеты создаются напрямую, без учета последовательности заданий, а затем пакеты обрабатываются на машине пакетной обработки. В процессе создания пакета, задания, которые будут добавлены к существующим пакетам, могут быть тщательно отобраны с учетом использования пакета и времени обработки пакета. Чтобы улучшить возможности глобального поиска в MMAS, был предложен локальный метод поиска, основанный на итеративном обмене несколькими заданиями.

Оставшаяся часть этой статьи организована следующим образом. Математическая модель проблемы, изучаемой в этой статье, представлена в Разделе 2. В Разделе 3 мы показываем подробный алгоритм MMAS, используемый для решения рассматриваемой проблемы. Настройка параметров и численные эксперименты приведены в Разделе 4. Статья окончательно завершена в Разделе 5.

**2. Математическая модель**

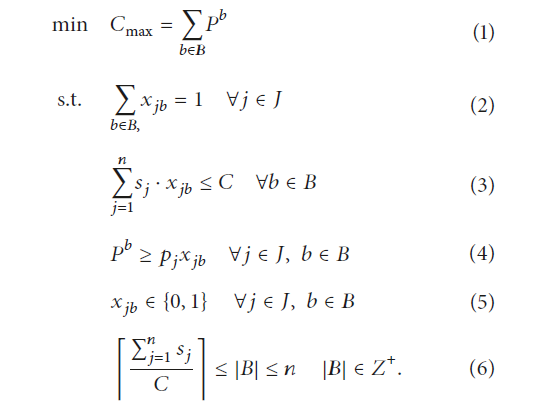
Проблема составления графика обработки отдельных пакетов изучается с целью минимизации продолжительности рабочего времени (рабочего цикла) выполнения. Машина пакетной обработки может обрабатывать несколько заданий в виде пакета, и все задания в этом пакете имеют одинаковое время начала и завершения. Процесс не может быть прерван после начала выполнения, и никакие задания не могут быть добавлены или удалены с машины, пока все задания не будут завершены. Время пакетной обработки определяется заданием с наибольшим временем обработки в пакете. На нулевом моменте все задания доступны. Символы и обозначения, используемые в этой статье, перечислены ниже:

(1) Есть 𝑛 заданий 𝐽 = {1, 2, . . . , 𝑛} для обработки и каждое задание 𝑗 имеет разное время обработки 𝑝𝑗 и размер 𝑠𝑗.

(2) Предполагается, что емкость машины равна 𝐶, а для каждого задания 𝑗 размер 𝑠𝑗 ⩽ 𝐶. Список заданий 𝐽 будет запланирован на пакете 𝑏 ∈ 𝐵 до того, как он будет обработан, где 𝐵 – список пакетов, то есть допустимое решение, а |𝐵| является количеством пакетов в 𝐵. Время обработки каждого пакета 𝑏 равно 𝑃𝑏 = max{𝑝𝑗 | 𝑗 ∈ 𝑏}.

(3) Цель состоит в том, чтобы свести к минимуму продолжительность рабочего цикла (рабочего времени) (𝐶max), которая равна общему времени обработки пакета в решении 𝐵.

Основываясь на допущениях и обозначениях, приведенных выше, мы можем получить следующую математическую модель задачи.



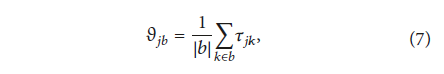
Задача (1) - минимизировать продолжительность работы. Поскольку рассматривается только одна обрабатывающая машина, рабочий период равен общему времени завершения всех сформированных пакетов. Ограничение (2) гарантирует, что каждое задание 𝑗 может быть назначено ровно одному пакету. Ограничение (3) гарантирует, что общий объем заданий в пакете не превышает емкость машины 𝐶. Ограничение (4) объясняет, что время пакетной обработки определяется заданием с наибольшим временем обработки в этом пакете. Ограничение (5) обозначает двоичное ограничение переменной 𝑥𝑗𝑏, равное 1, если задание 𝑗 назначено пакету 𝐵, и 𝑥𝑗𝑏 = 0 в противном случае. Ограничение (6) дает верхнюю и нижнюю границу числа пакетов в допустимом решении 𝐵. Нижняя граница рассчитывается, если предположить, что задания могут быть частично обработаны пакетами [4]. И верхняя граница генерируется, когда каждый пакет вмещает только одно задание.

**3. Max–Min Ant System**

MMAS [14] является одним из наиболее успешных вариантов в рамках оптимизации подражанием муравьиной колонии (муравьиного алгоритма) (ACO) [15, 16], которые были применены ко многим задачам комбинаторной оптимизации, таким как задачи планирования [17], задачи назначения трафика [18] и проблемы коммивояжера [15]. В MMAS интервал ограничения следа феромона [𝜏min, 𝜏max] используется для предотвращения преждевременной конвергенции и использования лучших решений, найденных во время поиска решения. На производительность алгоритма существенно влияют значения параметров; таким образом, выполняется настройка параметров для оптимизации производительности алгоритма. Локальный метод поиска также разработан для улучшения поисковых возможностей MMAS.

MMAS - это конструктивный метаэристический алгоритм, который способен строить решение шаг за шагом. Он может быть адаптирован к различным проблемам комбинаторной оптимизации с несколькими модификациями. При наличии списка заданий MMAS группирует задания в пакеты, добавляя задания в существующие или новые пакеты по одному. Последовательность выбора заданий для формирования в пакеты зависит от вероятности перехода состояния, вычисленной по плотности феромонного тракта и эвристической информации каждого элемента решения. Решение создается после того, как все задания объединены в пакет.

*3.1. Феромоновые тропы.* При решении задачи TSP (задача коммивояжера) с использованием муравьиных алгоритмов [15] ,следы феромонов 𝜏𝑖𝑗 определяются ожиданием выбора города 𝑗 из города 𝑖, то есть количеством феромонов дуги (𝑖 𝑗). Город с более высокой плотностью следа феромонов будет выбран с большей вероятностью. Однако в задаче планирования машины пакетной обработки каждое решение представляет собой набор пакетов, и последовательность заданий в пакете не влияет на время пакетной обработки; таким образом, следы феромонов подразумевают ожидание организации работы в пакете. В этом исследовании мы измеряем ожидание добавления задания в пакет, используя средние феромонные следы между заданием 𝑗 и каждым заданием в пакете 𝑏 следующим образом.



где 𝜏𝑗𝑘 означает след феромона между заданием 𝑗 и существующим заданием 𝑘 в пакете 𝑏. 𝜗𝑗𝑏 обозначает ожидание добавления задания 𝑗 в текущий пакет 𝑏. | 𝑏 | обозначает количество заданий в пакете 𝑏. И это ожидание будет использовано в качестве следов феромонов при расчете вероятности перехода состояния.

*3.2. Эвристическая информация*. Поскольку цель 𝐶max равна общему времени обработки всех сформированных партий, качество решения зависит как от количества партий, так и от времени обработки каждой партии в решении. Таким образом, в данном исследовании рассматриваются два вида эвристической информации, а именно: использование емкости машины и эффективность времени пакетной обработки.

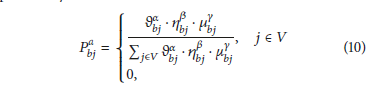
Обычно решения с меньшим количеством пакетов дают лучшие результаты. Чтобы уменьшить свободную емкость каждого пакета, мы добавляем задание 𝑗 в пакет 𝑏 с наиболее выполнимой емкостью в приоритете в соответствии с FFD (первый подходящий по убыванию) алгоритмом задачи об упаковке в контейнеры (bin packing problem) [19]. Эвристика для добавления выполнимой задачи 𝑗 в пакет 𝑏 определяется следующим образом:



Время обработки пакета определяется заданием с наибольшим временем обработки в пакете. Очевидно, что задания с одинаковым временем обработки должны быть объединены вместе, чтобы повысить эффективность времени пакетной обработки. Таким образом, мы даем еще одну эвристическую информацию для добавления задания 𝑗 в пакет 𝑏 следующим образом:

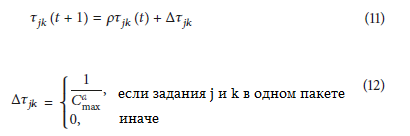


*3.3. Построение решения.* Для каждого муравья 𝑎 решение строится путем выбора незапланированного задания 𝑗 и добавления его в существующий пакет 𝑏 в соответствии с вероятностью перехода состояния . Если ни один из существующих пакетов не может вместить задание 𝑗, будет создан новый пустой пакет, в который задание 𝑗 будет добавлено. Решение будет сгенерировано, когда все задания запланированы в пакеты. Поскольку построение решения зависит от последовательности выбранных заданий, качество решения значительно зависит от вероятности перехода состояния. Вероятность определяется следами феромонов и эвристической информацией между текущим пакетом и заданием, которое необходимо запланировать. Вероятность перехода состояния определяется следующим образом:



где 𝑉 обозначает выполнимые задания, которые могут быть добавлены к текущему пакету 𝑏, так что емкость машины не нарушается. 𝛼, 𝛽 и 𝛾 показывают относительную важность феромоновых троп и двух видов эвристической информации. Для каждого муравья 𝑎 допустимое задание 𝑗 в 𝑉 будет выбрано с вероятностью и добавлено к текущему или новому пакету, пока все задания не будут запланированы.

*3.4. Обновление феромонов.* Плотность феромоновых троп является важным фактором в процессе построения решения, поскольку он указывает на качество компонентов решения. Когда все муравьи создадут свое выполнимое решение, следы феромонов на каждом компоненте решения будут обновляться за счет депонирования и испарения феромонов. После каждой итерации следы феромонов каждого компонента решения будут уменьшаться с увеличением скорости испарения 𝜌, в то время как компонент решения итеративного наилучшего решения или глобального наилучшего решения будет увеличиваться на величину Δ𝜏𝑗𝑘. Обновление феромона для каждого муравья 𝛼 на t-й итерации между компонентами решения задания 𝑗 и задания 𝑘 выполняется согласно (11) следующим образом.



где 𝜌 (0 ≤ 𝜌 <1) контролирует скорость испарения феромона. обозначает решение, которое получит муравей 𝑎.

Следы феромонов ограничены интервалом [𝜏min, 𝜏max] в алгоритме MMAS. согласно Штутцле и Хусу [14] в этом исследовании мы полагаем 𝜏max = 1/[] и 𝜏min = 𝜏max/2𝑛. это наилучшее рабочее время, которое нашли муравьи.

3.5. Алгоритм локального поиска. Когда в метаэвристике применяются методы локального поиска [20], качество решения может быть эффективно улучшено. Это связано с тем, что в методах локального поиска обычно применяются жадные стратегии, и локальный оптимум можно легко найти в окрестности заданного решения. Возможность метаэвристики в глобальном оптимальном поиске может быть улучшена путем объединения преимуществ в исследовании пространства решений с локальными методами поиска.

Поскольку время пакетной обработки определяется заданием с наибольшим временем обработки в пакете, задания с меньшим временем обработки не влияют на время пакетной обработки. Локальный метод поиска может применяться для уменьшения времени пакетной обработки путем совместной пакетной обработки заданий с одинаковым временем обработки.

*Определение 1.* Задание 𝑖 с наибольшим временем обработки в пакете 𝑏 называется доминирующим заданием . Общее время обработки заданий без доминирующего задания обозначается как доминирующее время обработки заданий, отмеченное как 𝐷𝑇.

**Теорема 2**. Рабочая область будет минимизирована путем максимизации 𝐷𝑇.

*Доказательство.* Имеем для заданного решения 𝐵. Согласно определению 1, ; таким образом, мы имеем . Поскольку общее время обработки задания является постоянным для данного экземпляра, рабочее время будет минимизировано, когда DT максимизируется.

*Определение 3.* Для каждого пакета 𝑏 сумма оставшейся емкости и размера доминирующей работы называется сменной емкостью ; мы обозначим .

**Теорема 4.** Учитывая ⩾ , больший DT будет получен путем обмена пакета 𝑏 с 𝑚 заданиями в пакете 𝑞, где и , пока не увеличивается.

*Доказательство.* Дано два пакета 𝑏 и 𝑞, ⩾ , задания в пакете 𝑏 можно разделить на три набора , 𝑚 заданий 𝑀 и другие задания 𝑂, где и . Задания в пакете 𝑞 могут быть разделены на наборы и другие задания 𝑂’. Таким образом, мы имеем . После того, как обмен применяется к пакетам, мы имеем , где . Так как , отношение выполнено. Согласно Теореме 2, существует ; меньший указывает на больший . Теорема 4 верна.

Согласно Теореме 4, многократные задания могут итеративно обмениваться между пакетами, чтобы уменьшить время пакетной обработки. Для данного решения 𝑆 количество пакетов ограничено общим количеством заданий, где каждое задание сгруппировано как один пакет. Подробная процедура предложенного алгоритма локального поиска MJE (Multiple Jobs Exchange) приведена ниже:

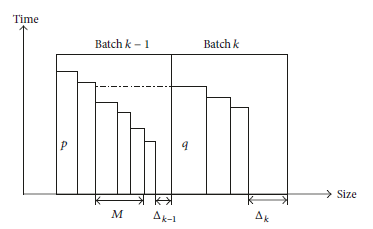


Рисунок 1 – Описание алгоритма MJE.

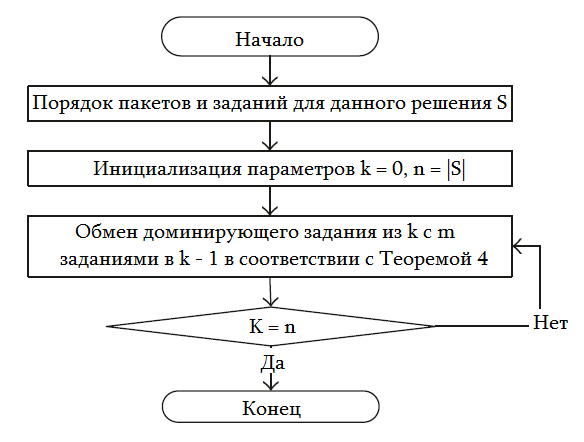


Рисунок 2 - Блок-схема алгоритма MJE.

*Алгоритм MJE (обмен несколькими заданиями)*

*Шаг 1*. Для лучшего итеративного решения 𝑆 расположите пакеты в порядке убывания времени их обработки и упорядочите задания для каждого пакета в порядке убывания размера задания.

*Шаг 2*. Инициализируйте параметр 𝑘 = 0, установите 𝑛 = | 𝑆 |, - оставшаяся емкость пакета 𝑘, а 𝑝 и 𝑞 обозначают доминирующее задание пакета 𝑘 - 1 и пакета 𝑘 соответственно.

*Шаг 3*. Если 𝑘 = 𝑛, выйти; иначе 𝑘 = 𝑘 + 1.

*Шаг 4*. Замените задание 𝑞 пакета 𝑘 на 𝑘 (𝑚> 0) заданий 𝑀 в пакете 𝑘 - 1, если max{𝑝𝑖 | 𝑖 ∈ 𝑀} ≤ ,

и ; перейдите к шагу 3.

Соответствующие обозначения показаны на рисунке 1.

Блок-схема алгоритма MJE представлена как на рисунке 2.

*3.6. Структура алгоритма MMAS с локальным поиском.*Согласно базовой структуре муравьиных алгоритмов [16], структура MMAS для решения исследуемой задачи задается следующим образом.

*Алгоритм ММАS*

*Шаг 1.* Инициализируйте параметры в MMAS, включая 𝛼, 𝛽, 𝛾 и 𝜏(0).

*Шаг 2.* Инициализируйте популяцию муравьев.

*Шаг 3.* Выберите незапланированное задание для каждого муравья 𝑎 и добавьте задание в новый пакет.

*Шаг 4.* Подготовьте следующуее незапланированное выполнимое задание с максимальной вероятностью перехода состояния, рассчитанной по (10), в текущий пакет.

*Шаг 5.* Если существуют задания, которые можно добавить в текущий пакет, перейдите к шагу 4.

*Шаг 6.* Если есть незапланированные задания, перейдите к шагу 3.

*Шаг 7.* Применение локального метода поиска MJE к итерационным решениям.

*Шаг 8.* Обновите феромоновые тропы в соответствии с формулой (11).

*Шаг 9.* Если условие завершения выполнено, выведите решение. Иначе перейдите к шагу 2.

Чтобы более логично проиллюстрировать процедуру MMAS, приведена блок-схема алгоритма MMAS, как показано на рисунке 3.

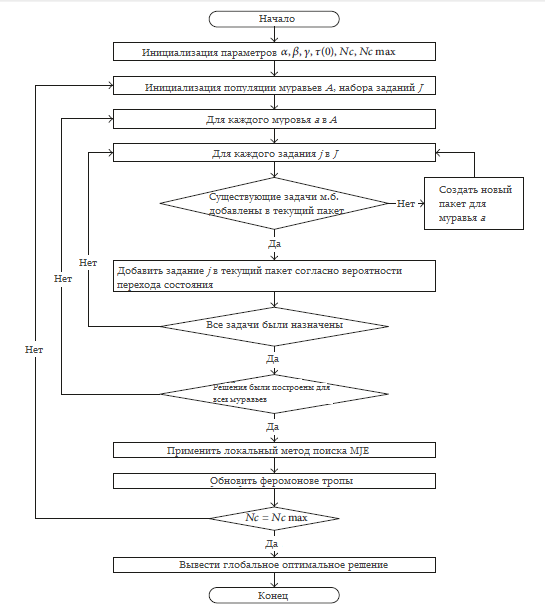


Рисунок 3 - Блок-схема алгоритма MMAS

**4. Эксперимент**

*4.1. Экспериментальное проектирование.* Случайные экземпляры были сгенерированы для проверки эффективности предложенного алгоритма. В численном эксперименте были рассмотрены три фактора, включая количество заданий 𝐽, время обработки заданий 𝑃 и размер заданий 𝑆. 24 (4 ∗ 2 ∗ 3) категории задач были сгенерированы и обозначены в виде 𝐽𝑖𝑃𝑗𝑆𝑘 (𝑖 = 1, 2, 3, 4; 𝑗 = 1,2; 𝑘 = 1,2,3). Коэффициенты и уровни показаны в таблице 1. Предполагается, что емкость машины равна 10 для всех проблемных случаев.

Например, 𝑃1𝑃2𝑆3 представлял категорию экземпляров с 10 заданиями, где время обработки заданий было случайно сгенерировано из дискретного равномерного распределения [1, 20], а размеры заданий были случайным образом сгенерированы из дискретного равномерного распределения [4, 8].

*4.2. Настройка параметров.* Поскольку каждый муравей будет строить возможные решения с вероятностью, большая группа муравьев расширит возможности алгоритма по исследованию пространства решений, в то время как большее количество итераций поможет лучше использовать феромоновые тропы и эвристическую информацию. Однако алгоритм требует гораздо больше времени при большем количестве муравьев и итераций. Были проведены предварительные тесты, и результаты показали, что увеличение количества итераций дает лучшие результаты в заданное время выполнения. Это легко понять, поскольку алгоритм склонен выполнять случайный поиск на начальных этапах, и наряду с увеличением числа итераций учитываются более эмпирические результаты. Чтобы получить компромисс между качеством решения и временными затратами, в этом исследовании была установлена численность муравьев, равная 30, и количество итераций, равное 80.

Из формулы (10) видно, что существует экспоненциальная связь между вероятностью перехода состояния, феромоновыми тропами и эвристической информацией. Таким образом, вероятность более чувствительна к факторам с большими параметрами 𝛼, 𝛽 и 𝛾. Чтобы проверить влияние этих параметров, а также выбрать их уровни, были проведены предварительные испытания. 𝛼 = 1 использовался в качестве эталонного уровня для изучения влияния параметров 𝛽 и 𝛾 на рабочее время в интервале [1, 10]. Результаты показаны на рисунке 4.

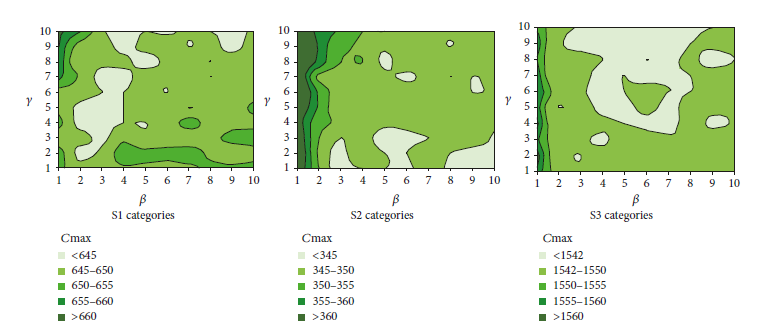
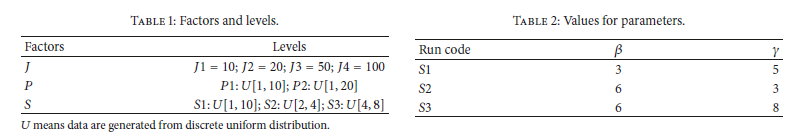


Рисунок 4 - Контурные линии 𝐶max под влиянием параметров 𝛼, 𝛽 и 𝛾 для разных категорий задач.



Как видно из рисунка 4, следует использовать средние 𝛽 и меньшие 𝛾 для экземпляров категории 𝑆2 с небольшими заданиями, чтобы получить лучшее время выполнения, в то время как больший 𝛾 следует использовать для экземпляров категории 𝑆3 с большими заданиями. Для экземпляров категории 𝑆1 со смешанными размерами заданий меньшие значения 𝛽 и средние 𝛾 кажутся лучше по сравнению с предыдущими двумя категориями. Все экземпляры дают плохие результаты, когда 𝛽 слишком близко к 1. Крупные задания обычно имеют меньшую гибкость при их размещении в пакетах по сравнению с заданиями малого размера. Столь высокий уровень 𝛾 используется для организации в первую очередь заданий большого размера с высокой вероятностью для экземпляров с большими заданиями. Точно так же, большой 𝛽 используется для пакетирования вместе заданий с одинаковым временем обработки, чтобы повысить эффективность времени пакетной обработки. Согласно приведенному выше анализу, в этом исследовании выбраны три уровня для каждого параметра, как показано в таблице 2.

Феромоновые тропы испаряются на каждой итерации, а скорость регулируется параметром 𝜌. (1 - 𝜌) является скоростью испарения. Высокая скорость испарения приводит к постоянному изменению феромоновых троп на каждом элементе решения, в то время как низкая скорость приводит к таким феромоновым тропам на элементах решения, которые не могут испаряться во времени.

Феромонные тропы каждого элемента решения ограничены в интервале [, ] в алгоритме MMAS. Феромонные тропы обычно устанавливаются на высокое значение в начале работы, чтобы улучшить способность поиска в пространстве решения. Если элемент решения всегда находится в состоянии испарения и его феромонная тропа уменьшается до , сразу после 𝑁𝑐 итераций, то мы имеем , = ,. Поскольку = / 2𝑛, как упомянуто выше,то можно получить, что = 1 / 2𝑛; то есть , 0 <𝜌 <1, где 𝑛 - количество заданий. Отношения между 𝑁𝑐 и 𝜌 для 100 рабочих мест описаны на рисунке 5.

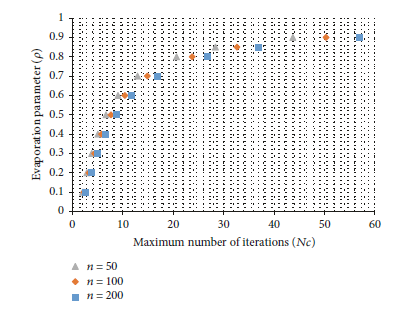
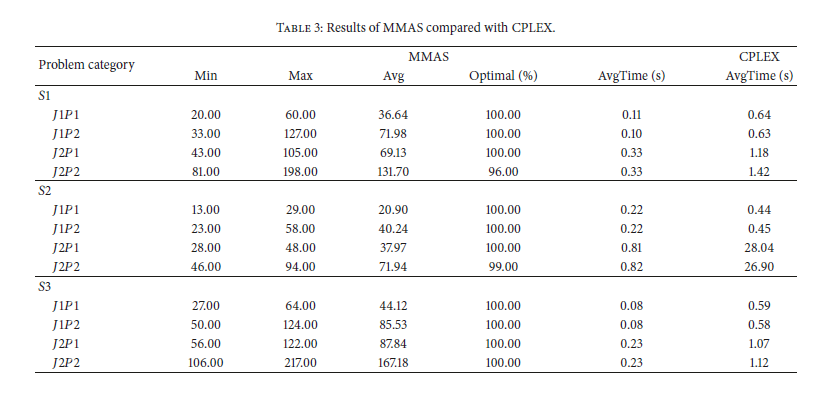


Рисунок 5 – Отношения между 𝑁𝑐 и 𝜌

Из рисунка 5 видно, что 𝑁𝑐 резко возрастает, когда 𝜌> 0,6; таким образом, в этом исследовании значение 𝜌 установлено на 0,6, чтобы гарантировать, что феромонные тропы испаряются не слишком быстро или медленно, и пространство решения может быть исследовано за разумное время.

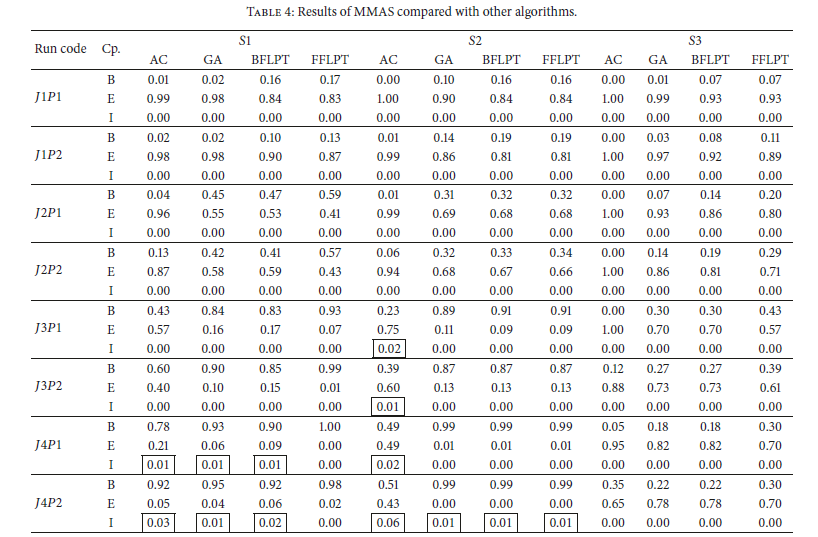
*4.3. Сравнение производительности для экземпляров малого масштаба.* Как показано в таблице 3, результаты, полученные из MMAS, сравниваются с результатами из CPLEX (коммерческий решатель для линейных и смешанных целых задач). Из-за NP-сложности задачи, только мелкие экземпляры задач категорий 𝐽1 и 𝐽2 решаются с помощью CPLEX. Минимальные, максимальные и средние значения 𝐶max, полученные из MMAS, перечислены в столбцах с 3 по 5. Среднее время выполнения MMAS и CPLEXT приведены в столбцах 7 и 8. В столбце 6 указан процент оптимальных значений, полученных с использованием MMAS по сравнению с результатами CPLEX.



Почти все мелкомасштабные экземпляры могут быть оптимально решены с помощью MMAS, за исключением нескольких экземпляров из 𝐽2𝑃2𝑆1 и 𝐽2𝑃2𝑆2 с относительно большим пространством решения. MMAS является конкурентоспособной в вычислительном отношении по сравнению с CPLEX для всех случаев. Вычислительное время всех экземпляров из MMAS составляет менее 1 секунды. Время вычислений CPLEX сильно варьируется в зависимости от пространства решений экземпляра.

*4.4. Оценка эффективности алгоритма*. Чтобы оценить производительность алгоритма для всех категорий проблем, MMAS, разработанный в этой статье, сравнивался с GA [9]. Чтобы хорошо решить проблемы был закодирован базовый муравейный алгоритм (AC), а параметры использованные в нем были такими, как предписано в Чэн и др. [21]. Две известные эвристики FFLPT и BFLPT также сравнивали с MMAS в экспериментальном исследовании. Для каждой проблемной категории было сгенерировано 100 экземпляров и в итоге был использован лучший результат из 10 прогонов для каждого экземпляра.

Сравнение различных алгоритмов обобщено в таблице 4. В столбце 1 таблицы 4 представлен код выполнения. Столбец 2 показывает результаты MMAS по сравнению с другими алгоритмами, в которых B, E и I означают, что интервал времени, полученный с помощью MMAS, лучше, равен или ниже, чем у других алгоритмов, соответственно. Столбцы 3–6 сообщают о пропорции, которую составной период, полученный с помощью MMAS, по сравнению с другими алгоритмами в 100 экземплярах категории проблем 𝑆1. Столбцы 7–10 и 11–14 аналогичны столбцам 3–6. Например, первое число 0,01 указывает, что в MMAS доля результатов, о которых сообщается в 0,01, лучше, чем в AC. Поскольку исследуемая проблема NP-сложна, пространство решения должно взорваться вместе с увеличением численности населения от 𝐽1 до 𝐽4. С другой стороны, меньшие размеры работы дают больше комбинаций пакетов; таким образом, пространство решения становится меньше для 𝑆3 категорий задач по сравнению с 𝑆1 для заданного числа заданий.



Из таблицы 4 видно, что MMAS превосходит другие алгоритмы в целом. Несколько результатов, полученных из MMAS, уступающих результатам, полученным другими алгоритмами, были выделены прямоугольником. Для категорий задач 𝐽1 результаты, полученные из MMAS, в основном равны результатам других алгоритмов. Около 16% результатов MMAS лучше, чем у алгоритмов BFLPT и FFLPT для проблемных категорий 𝑆1 и 𝑆2. Для категорий задач 𝑆3 результат составляет 7%, потому что большинство результатов одинаковы для всех алгоритмов в небольшом пространстве решений. MMAS сообщает, что до 14% лучше, чем GA для проблемной категории 𝑃1𝑃2𝑆3. Преимущество MMAS для таких алгоритмов, как GA, BFLPT и FFLPT, гораздо более заметно для 𝐽2 проблемных категорий. MMAS дает лучшие результаты, чем AC, и этот процент составляет от 6% до 13% для категорий проблем 𝐽2𝑃2. Более 80% результатов, полученных в MMAS, лучше, чем в GA, BFLPT и FFLPT для проблемных категорий 𝐽3𝑆1 и 𝐽3𝑆2. Около 23% до 43% процентов результатов от MMAS лучше, чем AC, в то время как последний показывает менее хорошие результаты. Для проблемных категорий 𝐽4 более 90% результатов, генерируемых MMAS, лучше, чем результаты алгоритмов AC, GA, BFLPT и FFLPT для экземпляров больших заданий. Почти все другие алгоритмы показывают несколько лучшие результаты, чем MMAS для проблемных категорий 𝐽4.

На основании анализа, приведенного выше, MMAS обладает более высокой способностью в исследовании пространства решения. Он превосходит все остальные алгоритмы почти во всех случаях, особенно для большого числа заданий. Поскольку MMAS исследует пространство решений в вероятности, оптимальное решение не может быть гарантировано с ограниченной совокупностью и итерациями. Поэтому некоторые результаты, полученные из MMAS, уступают другим алгоритмам, даже эффективной эвристике, такой как BFLPT и FFLPT.

Время выполнения увеличивается для 𝑃2 проблемных категорий по сравнению с 𝑃1 проблемными категориями, но нет существенного изменения производительности для различных алгоритмов.

Чтобы проиллюстрировать разрыв между MMAS и другими алгоритмами для разных размеров популяции, средний процент результатов с лучшим рабочим периодом, полученных MMAS по сравнению с другими алгоритмами, представлен на рисунке 6.

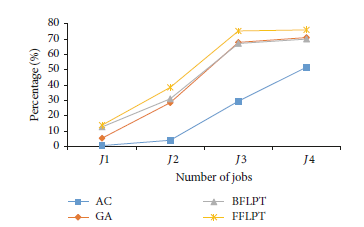


Рисунок 6 - Средний процент результатов, полученных из MMAS по сравнению с другими алгоритмами.

В среднем, от 10% результатов, полученных из MMAS, лучше, чем у других алгоритмов для категорий проблем 𝐽1, в то время как для 𝐽4 это число составляет 70%. Алгоритм AC является вторым лучшим алгоритмом по сравнению с GA и двумя другими эвристиками. BFLPT и FFLPT эффективны в решении проблемы для небольшого размера населения. BFLPT обычно лучше, чем FFLPT почти для всех категорий проблем, а BFLPT также используется в GA для генерации первоначальных решений.

**5. Выводы и будущая работа**

В этой статье была изучена проблема составления плана обработки отдельных пакетов. Для решения проблемы был разработан и применен улучшенный муравьиный алгоритм MMAS (Max – Min Ant System). Параметры MMAS были определены предварительным экспериментом, а также был представлен метод локального поиска MJE для улучшения производительности MMAS.

Оптимальные решения могут быть получены с помощью MMAS за меньшее вычислительное время по сравнению с CPLEX для небольших задач. Чтобы оценить производительность MMAS, ее сравнили с базовым алгоритмом муравья AC (Ant Cycle), GA (генетический алгоритм) и двумя другими известными эвристиками планирования пакетной обработки, то есть FFLPT (First Fit Longest Time Processing Time) и BFLPT (Best Fit Longest Processing Time). Численный эксперимент показал, что MMAS превзошел другие алгоритмы почти во всех случаях в различных категориях задач, особенно для больших групп населения.

Учитывая хорошую производительность MMAS, может быть разработан улучшенный механизм эвристики и применен к другим задачам планирования машины пакетной обработки. Исследуемая проблема может быть изучена с другими целями и ограничениями, включая время выпуска и сроки выполнения.

**Конфликт интересов**

Авторы заявляют, что у них нет конфликта интересов.

**Подтверждения**

Эта работа поддерживается Фондом фундаментальных исследований для центральных университетов (№ 2014QNA48) и Национальным фондом естественных наук Китая (№ 71401164)