**Использование эвристического поиска для решения задач по обработке одной машиной**

Thanh-Tung Dang, Baltaz ́ar Frankovicˇ, Ivana Budinska ́

Institute of Informatics

Slovak Academy of Sciences

D ́ubravsk ́a cesta 9

845 07 Bratislava, Slovakia

e-mail: utrrtung@savba.sk, dang.tung@ul.ie

Ben Flood

Center for Telecomunications and Value-Chain-Driven Research

Trinity College Dublin

Dublin, Ireland

Con Sheahan

Enterprise Research Centre

University of Limerick

Limerick, Ireland

Bao-Lam Dang

Department of Design of Machines and Robots

Hanoi University of Technology

Hanoi, Vietnam

Manuscript received 6 June 2005; revised 12 May 2006

Communicated by Imre J. Rudas

**Аннотация.** В этой статье рассматривается планирование для пакетной обработки одной машиной, в частности, задача транспортировки с одним транспортным средством. Рассмотрены ограничения по мощности машины, и главная цель – найти назначение заданий для достижения минимального времени обработки всех партий, учитывая эти мощности. Предложен полиномиальный алгоритм для решения случая, когда задания являются не перегрузочными, неидентичными и известны до создания графика. Предложенный алгоритм реализован и показал, что даёт лучшие результаты, чем альтернативы из литературы.

**Ключевые слова:** Планирование, пакетная обработка, оптимизация, эвристический поиск.

**1 Введение**

В машине пакетной обработки наборы заданий обрабатываются одновременно, как пакеты. Время обработки каждого пакета – это максимальное время обработки всех заданий, включенных в пакет. Каждое задание имеет набор весов, и машина для пакетной обработки имеет ограниченную производительность.

Одним из применений, в которых используются машины для обработки пакетов, является проблема транспортировки на производстве. Транспортное средство с автоматическим управлением (AGV) должно доставлять продукты или пакеты материалов, или компонентов на ряд рабочих станций в разных местах в кратчайшие сроки. Рабочие станции расположены по кольцу или по прямой. Транспортное средство – это машина для пакетной обработки, а задание – это доставка материалов из магазина на рабочую станцию. Время, необходимое для выполнения пакета, не зависит от количества заданий в пакете и здесь относится только к самому дальнему местоположению, Транспортные средства имеют ограниченный размер и могут нести ограниченный вес. В результате каждое транспортное средство может доставлять только комбинации продуктов или упаковок с общими размером и весом, не превышающими вместимость транспортного средства. Августин и Сейден [3], а также Каруно и Нагамочи [7] описывают примеры, когда транспортное средство посещает множество мест для выполнения работ. Ограничения транспортного средства в этих случаях не учитываются. поскольку транспортное средство, например робот, выполняет задачи и не доставляет определенное количество.

Существует много других практических применений, в которых используются машины для периодической обработки, например, для испытания материалов, электрических цепей и термообработанных печей. Интересным примером является печь для обжига на заводах по производству керамики, используемая для производства ряда промышленных компонентов, таких как бусы, мосты, опоры, соединительные блоки и др. (http://www.ceramicx.com/). Эти компоненты затем используются в различных продуктах. Все компоненты обрабатываются в печи, но каждый тип компонентов требует разного давления и температуры. Каждый тип компонентов имеет разные размеры и вес, а сушильная машина имеет физические ограничения, в том числе ограниченный размер и общий вес. Следовательно, компоненты должны быть сгруппированы в одинаковые партии, чтобы общее время обработки было сведено к минимуму.

В этой статье рассматривается проблема планирования для одной машины, которая в целом может обрабатывать задания партиями, где будут рассматриваться различные ограничения, такие как ограниченный размер машины, ограниченная пропускная способность и т. д. Основная цель - найти график для назначения неидентичных и не вытесняющих заданий отдельным партиям, чтобы минимизировать общее время обработки.

**2 Связанные задания**

Если существует n заданий, которые нужно обработать на m машинах, то существует (n!)m возможных расписаний, что подразумевает, что проблема NP-сложная [8]. Большая работы была проделана в задачах пакетной обработки. аналогично представленной здесь. [2] показывают, что проблема может быть решена за полиномиальное время, когда последовательность заданий предопределена. В [5] представлен полиномиальный алгоритм для решения задачи планирования для особого случая, когда веса каждого задания равны.

Во многих статьях обсуждаются задачи, при которых время обработки варьируется для разных партий и зависит от заданий, используемых для формирования партий. [14] обсуждают проблему планирования для технологической цепочки машин пакетной обработки. Предполагается, что задания обрабатываются в одном и том же порядке на каждом компьютере. То есть, если два задания i и j обрабатываются в двух разных пакетах на машине Mk, и i обрабатывается до j, то задание i всегда будет обрабатываться не позднее, чем задание j на каждой последующей машине в потоковом обработчике. Такое допущение уменьшает сложность задачи планирования, поскольку при расчете необходимо сосредоточиться только на машинах, ограничения емкости которых являются узкими местами потокового обработчика. Динамическое программирование (DP) используется для эвристической минимизации времени завершения. DP гарантирует, что будет найдено оптимальное решение. Тем не менее, это требует изучения всех возможных графиков. В результате DP имеет очень высокую сложность и может применяться только для небольших задач. Когда задания не являются приоритетными, то есть когда задания обрабатываются, они должны быть завершены целиком, обработка задания не может быть приостановлена ​​или возобновлена, а задания могут быть обработаны в любом порядке, тогда применение DP невозможно для больших наборов заданий.

Другими эвристическими решениями задачи планирования являются генетические алгоритмы (GA) [6] и моделируемая нормализация (SA) [9]. Качество решений, полученных этими методами, сильно зависит от времени работы алгоритмов. Важным шагом в алгоритмах, основанных на эволюционном принципе, является создание соседних решений, что занимает много времени из-за процесса пересчета параметров графика. В алгоритме, представленном в [9], два случайно выбранных задания из разных партий обмениваются друг с другом, чтобы создать новое решение, если производительность машины не будет нарушена. Одно из выбранных заданий является самым длинным заданием обработки в его пакете. Это называется мутационным процессом. Процесс мутации останавливается, когда соотношение между улучшением времени завершения и параметром «температура» T, которое регулярно уменьшается на предварительно определенную постоянную в каждом цикле, равно или больше, чем предварительно определенная постоянная.

Другой подход к задаче планирования - программирование ограничений (CP) [4]. Проблема рассматривается как набор переменных (то есть время начала каждого задания, соответствует каждому заданию в пакете) и набор ограничений (то есть все ограничения пакета). Задача состоит в том, чтобы найти присваивание переменных, которые удовлетворяют всем ограничениям. Основным принципом этих методов, основанных на CP, является объединение распространения ограничений и обратного отслеживания для поиска решения.

[17] обзор методов решения задач планирования машин пакетной обработки, основанный на принципе Лагранжевой релаксации (LR). LR используется для разложения задачи на непересекающиеся подзадачи. Графики получаются путем итеративного точного решения этих подзадач, но обычно приводят к неоптимальному общему решению. В LR есть два метода: динамическое программирование в обратном направлении (BDP) и динамическое программирование в прямом направлении (FDP). [16] использует динамическое программирование в обратном направлении, чтобы найти решение для задачи планирования машины пакетной обработки с одной машиной, предполагая, что каждая партия может обрабатывать максимум b заданий одновременно, независимо от их возможностей. Подобный подход обсуждался в [12], рассматривая проблему стохастического пакетного обслуживания, в которой входы и выходы каждого пакета взаимозависимы.

В этой статье рассматривается задача планирования для одной пакетной машины, но с некоторыми существенными отличиями в том, что рабочие места не являются приоритетными и не идентичны по размеру или весу. То есть, пакетный автомат имеет ограниченный размер и грузоподъемность. Цель состоит в том, чтобы найти расписание, которое минимизирует сумму времени завершения всех партий.

**3 Формулировка проблемы**

Есть n пакетов {Pi | i = 1,…, n}, каждый пакет Pi, представлен тремя параметрами (si, wi, ti), где si и wi соответственно размер и вес упаковки Pi, и ti обозначает время доставки из магазина на рабочую станцию. Время доставки включает в себя время, необходимое для загрузки и выгрузки работ, все рабочие станции обрабатываются с одинаковым приоритетом.

Рассмотрим автомобиль с размером s и грузоподъемностью c. Цель состоит в том, чтобы запланировать задания, которые берёт автомобиль, партии, так, чтобы общее время доставки всех работ было минимизировано. Транспортное средство может принимать новые пакеты только после завершения доставки всех пакетов, включенных в его текущую партию. Время обработки каждой партии определяется наибольшим временем доставки среди всех пакетов в партии, в результате предположения, что рабочие станции расположены по кольцу или по прямой линии. Пусть k будет количеством пакетов, {Bi | i = 1,…, k} будет набором пакетов, включенных в партию i, а Ti будет временем обработки пакета i. Следующие уравнения описывают проблемные ограничения.

∀i ≠ j ∈ [1, k] : Bi ∩ Bj = 0 (1)

∀i ≠ j ∈ [1, k]:

и

Цель состоит в том, чтобы найти распределение пакетов по отдельным партиям с небольшим значением общего времени обработки всех партий T,

Уравнения (1) и (2) гарантируют, что каждый пакет назначен только одной партии. Ограничения (3) и (4) гарантируют, что общий размер и вес всех упаковок, включенные в каждую партии не превышают ограничения по вместимости транспортного средства. Уравнение (5) указывает время обработки каждой партии. Уравнение (6) выражает общее время расписания, которое должно быть сведено к минимуму (также известное как составление графика).

Если цель состоит только в том, чтобы минимизировать количество партий, а не T, общее время обработки, то это классическая задача минимального пакета. В задаче о минимальном пакете учитывается только один параметр - размер упаковки. Это NP-сложная задача, но ее можно аппроксимировать в пределах 1,5, то есть существует схема аппроксимации, которая гарантирует, что

numBatchessol/numBatchesвopt ≤ 1.5,

где numBatchessol и numBatchesвopt - это количество пакетов, полученное с использованием схемы аппроксимации, и оптимальное количество пакетов, необходимое для обработки всех заданий, соответственно [13]. В задачах с низкой корреляцией между параметрами заданий, независимостью между размером, весом и временем обработки алгоритмы для решения задачи минимальной упаковки потребовали бы множество модификаций для решения рассматриваемой здесь задачи.

**4 Решение с использованием эвристического поиска**

Обсуждаемые здесь задания — это пакеты, которые необходимо доставить. Предполагается, что все параметры, описывающие пакеты и транспортное средство, известны в начале процесса решения. Таким образом, оптимальное решение может быть найдено путем изучения всех возможных графиков. Однако данные пакеты имеют равный приоритет, и число потенциальных решений теоретически очень велико и растет в геометрической прогрессии с увеличением количества пакетов. Эвристические алгоритмы поиска должны использоваться для задач практически определенного размера. Качество решения, достигаемого с помощью алгоритмов эвристического поиска, зависит от предложенного алгоритма и продолжительности времени, в течение которого работает эвристический алгоритм.

Многие методы решения задачи минимального пакета основаны на том принципе, что пакеты сортируются в соответствии с их размером, а затем пакеты вставляются один за другим в каждую партию, не нарушая ограничений по емкости. Этот процесс продолжается до тех пор, пока все пакеты не будут назначены. Чтобы свести к минимуму количество пакетов, методы обычно пытаются вставить как можно больше пакетов в каждый пакет. Однако решение с минимальным количеством партий не гарантирует автоматического минимального общего времени обработки, что является основной целью данной статьи.

Один из способов сократить общее время обработки - объединить пакеты с близким временем доставки в одну и ту же партию. Поскольку вместимость транспортного средства, пакеты с близким временем доставки не обязательно добавляются к той же партии. Обозначим количество созданных пакетов numBatch. Чтобы уменьшить пустое пространство в каждом пакете, перед созданием нового пакета полезно определить, можно ли добавить нераспределенный пакет в ранее созданный пакет. Это основа для предложенного нами алгоритма.

**4.1 Алгоритм 1 – Заполнение пустых мест пакетами (FES)**

**Этап 1**

Сортировать пакеты по времени доставки в {Pi | i = 1,…,n} так, что   
ti ≥ ti+1, для i = 1, …, n.

**Этап 2**

Пусть numBatch = 1;

For i = 1,…, n1

1. Поиск пакета. в котором достаточно места для пакета Pi;
2. Если такого пакета не существует, создать новый пакет и добавить текущий пакет в новую партию, и увеличить numBatch на 1. Если партия с достаточным количеством места для Pi существует, тогда переход к шагу 3.
3. Добавить пакет Pi в первую найденную пустую партию;

End For

**Этап 3**

Рассчитать общее время созданного графика.

**Утверждение 1.** Алгоритм FES имеет сложность O(n2).

**Доказательство.** Этап 1 имеет сложность O(n \* log(n)) с использованием алгоритма быстрой сортировки. На этапе 2, поскольку количество пакетов меньше или максимально соответствует i на этапе 1 требуется максимально i ( ≤ n) проверочных операций. Этапы 2 и 3 требуют только одной операции. В результате цикл этапа 2 требует максимально n2 операций. На этапе 3 время обработки каждой партии рассчитывается, чтобы получить общее время графика. Поскольку максимальное количество партий n, то на этом этапе необходимо максимум O(n) операций. Следовательно, предложенный алгоритм FES имеет сложность O(n2).

Хотя FES является полиномиальным алгоритмом, в некоторых случаях он может быть улучшен некоторыми простыми модификациями. Например, рассмотрим ситуацию, в которой транспортное средство имеет ограниченный размер S = 5 и ограниченную грузоподъемность C = 10. Пакеты выдаются со следующими параметрами P1 = {2, 3, 10}; P2 = {2, 2, 8}; P3 = {3, 2, 6}; P4 = {3, 2, 6}; и P5 = {4, 2, 2}. Эти параметры выражают размер, вес и время доставки каждой упаковки соответственно. При применении FES будет достигнут следующий план. План 1: партия 1: {P1, P2}, партия 2: {P3}, партия 3: {P4} и партия 4: {P5}. Общее время этого плана составляет 24 (= 10 + 6 + 6 + 2). Однако легко понять, что План 2: {партия 1: {P1, P3}, партия 2: {P2, P4} и партия 3: {P5}} имеют лучшее время обработки (20 = 10 + 8 + 2).

FES пропускает лучшее решение, так как после Этапа 1 порядок каждого пакета фиксирован. Пакеты P1 и P2 запланированы на одну партию, что заставляет остальные пакеты присваиваться каждой отдельной партии в зависимости от их размера. План 2 может быть идентифицирован, если пакеты {P1, P2} запланированы после пакетов {P3, P4}. Эта идея может быть обобщена следующим правилом. Перед применением FES некоторые пакеты небольшого размера (или небольшого веса) временно удаляются из процесса расчета, а после достижения временного плана оставшиеся пакеты последовательно добавляются в один созданный пакет, если на нем достаточно свободного места. Исходя из этого, предлагается следующая модифицированная версия алгоритма FES.

**4.1 Алгоритм 2 – Частичное заполнение пустых мест партиями (PFES)**

Пусть k = 0.

**Этап 1**

1. Сортировать пакеты по размеру (или весу);
2. Добавить (n-k) пакетов с большим размером (или весом) в один набор, называемый набором 1. Добавить оставшиеся k пакетов во второй набор, называемый набором 2;
3. Сортировать пакеты в каждом наборе в соответствии со временем их доставки.

**Этап 2**

1. Применить алгоритм FES к пакетам в наборе 1;
2. Применить алгоритм FES к пакетам в наборе 2.

**Этап 3**

1. Рассчитать общее время созданного графика
2. Если k < n, то пусть k = k + 1 и вернуться к Этапу 1. Если k = n, остановиться.

**Утверждение 2.** PFES имеет сложность ≅ O(n3).

**Доказательство.** Используя алгоритм быстрой сортировки, можно увидеть, что Этап 1 PFES имеет сложность ≅ O(n \* log(n)). Теорема 1 показывает, что FES имеет сложность ≅ O(n2). В результате Этап 2 требует максимально n2 операций. Поскольку, количество пакетов максимально n, для Этапа 3 необходимо максимально n операций. Наконец, каждый цикл PFES имеет сложность ≅ O(n2), и таких циклов n. Отсюда следует, что PFES имеет сложность ≅ O(n3).

Алгоритм FES является частным случаем PFES, где k = 0. Алгоритм PFES всегда будет по крайней мере таким же хорошим решением, как алгоритм FES. В следующем разделе описываются результаты моделирования предложенных алгоритмов и сравнивается производительность алгоритмов с другими опубликованными алгоритмами.

**5 Результаты моделирования и сравнение**

Все используемые здесь алгоритмы реализованы на Java и были проверены в ряде экспериментов. Алгоритм SA, представленный в [9], выбран для сравнения. Другой метод сравнения - алгоритм BDP, представленный в [12]. В алгоритме BDP исходный набор пакетов делится на два непересекающихся набора, а DP используется для нахождения оптимального расписания в каждом наборе. Окончательное решение достигается путем объединения частичных графиков. Если количество пакетов велико, BDP также может быть применен рекурсивно.

Численные результаты моделирования приведены в таблице 1. В первом столбце показано количество пакетов. В следующих столбцах показано время завершения расписаний, достигнутых четырьмя реализованными алгоритмами, и время выполнения каждого алгоритма, измеренное в миллисекундах. Время завершения каждого графика измеряется в тех же единицах, что и время доставки каждого пакета.

Размеры и веса упаковки генерируются случайным образом из равномерного распределения на интервале [1, 30]. Время доставки каждой посылки генерируется случайным образом из равномерного распределения на интервале [1, 50]. Рассматриваемое транспортное средство ограничено 50 единицами размера и 80 единицами веса. Количество пакетов находится в диапазоне от 50 до 104. Все эксперименты тестируются на одном и том же ПК с процессором Pentium 3 733 МГц.

Результаты моделирования показывают, что, когда число пакетов n мало (до 100), BDP реализуется для двух непересекающихся множеств. Однако, когда число увеличивается до 103 или более, рекурсивное применение BDP необходимо для сокращения времени выполнения. Алгоритм DP может исследовать до numPackage различных конфигураций, где (numPackage)! количество пакетов, включенных в каждую непересекающуюся часть. Таким образом, каждая непересекающаяся часть ограничена включением максимально 50 пакетов, чтобы сделать время решения практичным.

Алгоритм SA теоретически может работать в течение очень долгого времени, прежде чем будет выполнено условие остановки. Чтобы сравнить качество решений по времени выполнения, результаты SA и BDP записываются после завершения FES и PFES. Эти результаты показаны в таблице 2.

Чтобы получить полное расписание в любое время, BDP реализован следующим образом. Вначале исходный набор пакетов делится на несколько небольших подмножеств. Каждое подмножество включает в среднем от 5 до 10 пакетов, в зависимости от размера исходного набора. DP применяется для поиска оптимального расписания для каждого отдельного подмножества. Достигнутые графики объединяются для составления общего графика. Если время, доступное для выполнения, не истекает, эти подмножества последовательно объединяются, и весь процесс применяется снова.

Таблица 1 – Результаты экспериментов всех выбранных алгоритмов

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **FES** | | **PFES** | | **SA** | | **BDP** | |
| n | Время вып.  графика | Продолжительность | Время вып.  графика | Продолжительность | Время вып. графика | Продолжительность | Время вып. графика | Продолжительность |
| 50 | 285 | 1 | 285 | 39 | 313 | 136 | 460 | 200 |
| 100 | 583 | 3 | 581 | 112 | 696 | 133 | 1022 | 283 |
| 200 | 973 | 9 | 973 | 168 | 1154 | 216 | 1885 | 466 |
| 300 | 1544 | 18 | 1540 | 181 | 1755 | 548 | 2876 | 978 |
| 500 | 2634 | 25 | 2626 | 184 | 3017 | 1519 | 5048 | 3096 |
| 1000 | 5259 | 86 | 5229 | 872 | 5944 | 3929 | 9970 | 11346 |
| 2000 | 9954 | 96 | 9803 | 4654 | 11240 | 9720 | 19290 | 68529 |
| 5000 | 25626 | 237 | 25623 | 7299 | 29168 | 12584 | 47900 | 399820 |
| 104 | 52941 | 691 | 52741 | 80980 | 59120 | 116050 | 99135 | 2919800 |

Численное тестирование практических наборов данных показывает, что оба предложенных алгоритма достигают значительных улучшений по сравнению с SA и BDP, особенно когда число пакетов большое. Оба алгоритма SA и BDP достигают решений, аналогичных значениям FES и PFES, но после запуска в течение гораздо более длительного времени. Это приводит к выводу, что FES и PFES являются подходящими алгоритмами для решения крупномасштабных случаев, когда количество пакетов велико, а время выполнения ограничено.

Причина, по которой BDP не достигает решений такого же качества, как FES и PFES, заключается в том, что исходная проблема разделена на большое количество подзадач. Решение каждой подзадачи может быть оптимальным, но объединение полученных частичных решений может не дать хорошего общего решения. Это связано с тем, что пустой фрагмент, оставленный в созданных пакетах, слишком велик. Общий пустой фрагмент может быть уменьшен, когда количество пакетов, включенных в каждую подзадачу, увеличивается, но это приводит к экспоненциальному увеличению времени выполнения.

BDP может обрабатываться параллельно, поскольку включает в себя решение независимых подзадач. Однако параллельная обработка BDP не уменьшает сложность этого подхода; это только сокращает время работы путем решения многих независимых задач одновременно.

Таблица 2 – Результаты экспериментов всех выбранных алгоритмов после одинакового времени запуска.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **FES** | **PFES** | **SA** | **BDP** |  |
| n | Время вып.  графика | Время вып.  графика | Время вып.  графика | Время вып.  графика | Время вып.  графика |
| 50 | 285 | 285 | 345 | 489 | 39 |
| 100 | 583 | 581 | 756 | 1422 | 117 |
| 200 | 973 | 973 | 1357 | 1785 | 168 |
| 300 | 1544 | 1540 | 1965 | 3176 | 181 |
| 500 | 2634 | 2626 | 3214 | 5348 | 184 |
| 1000 | 5259 | 5229 | 6440 | 10950 | 872 |
| 2000 | 9954 | 9803 | 12238 | 22290 | 4654 |
| 5000 | 25626 | 25623 | 31062 | 50404 | 7299 |
| 104 | 52941 | 52741 | 63282 | 11239 | 80980 |

Качество решения, достигнутого SA, зависит от первого сгенерированного графика. По мере увеличения числа циклов среднее улучшение в решении за цикл уменьшается. При моделировании SA требовалось много времени для поиска, прежде чем она достигла таких же хороших решений, как те, которые были найдены FES или PFES. С другой стороны, SA — это алгоритм в любое время, способный обеспечить решение при остановке программы. По этой причине SA подходит для случаев, когда время, отведенное на поиск расписания, не фиксируется заранее.

Когда количество пакетов велико, более 1000, результаты моделирования (см. Таблицы 3, 4 и 5) показывают, что PFES не достигает значительно лучших результатов, чем FES. PFES достиг решения с качеством примерно на 0,1%. Это улучшение не так много, учитывая количество дополнительного времени, необходимого для запуска PFES. В общем, PFES занимает гораздо больше времени, чем FES, примерно в 10–20 раз больше, чтобы завершить его поиск. Для иллюстрации оба алгоритма были протестированы с 5000 пакетами, и качество достигнутых решений сравнивается и показано на рисунке 1. На рисунке 2 показано сравнение времени работы этих алгоритмов. Более численные результаты экспериментов с большим количеством пакетов показаны в таблицах 3, 4 и 5. Применение FES представляется наиболее подходящим подходом для использования при решении дел с большим количеством пакетов.

Таблица 3 – Результаты моделирования с 1000 пакетами.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **FES** | | **PFES** | | **SA** | | **BDP** | |
| n | Время вып.  графика | Продолжительность | Время вып.  графика | Продолжительность | Время вып. графика | Продолжительность | Время вып. графика | Продолжительность |
| 1000 | 5259 | 86 | 5229 | 872 | 5944 | 3929 | 9970 | 11346 |
| 1000 | 5296 | 35 | 5289 | 970 | 6084 | 4206 | 10265 | 11427 |
| 1000 | 5100 | 28 | 5078 | 917 | 5709 | 4134 | 10097 | 10838 |
| 1000 | 5010 | 94 | 4990 | 807 | 5692 | 3819 | 9648 | 10638 |

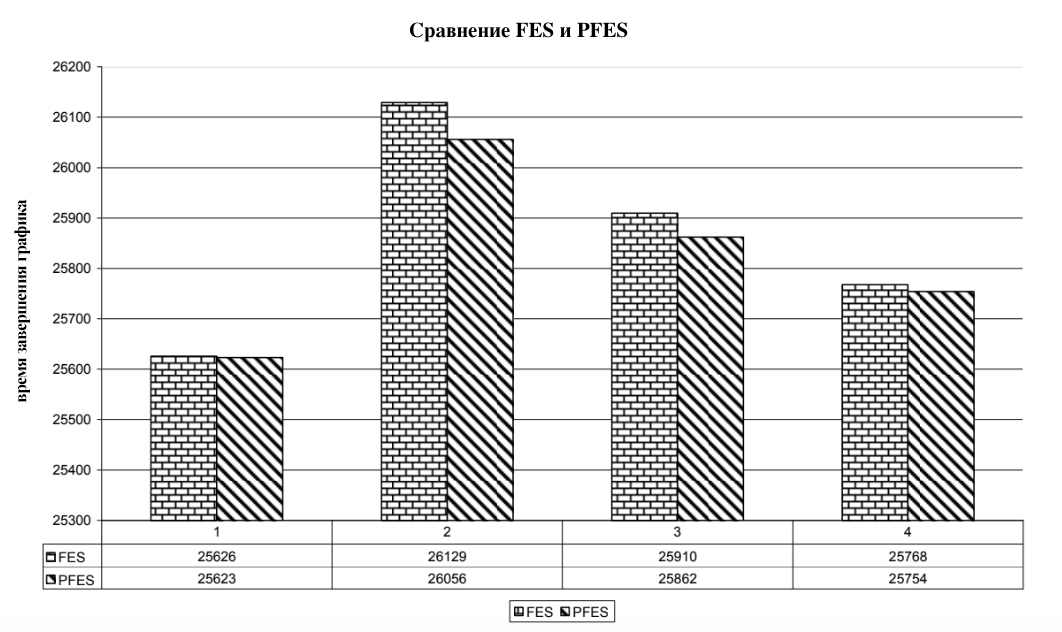


Рисунок 1 - Сравнение времени выполнения плана между FES и PFES с 5000 пакетами

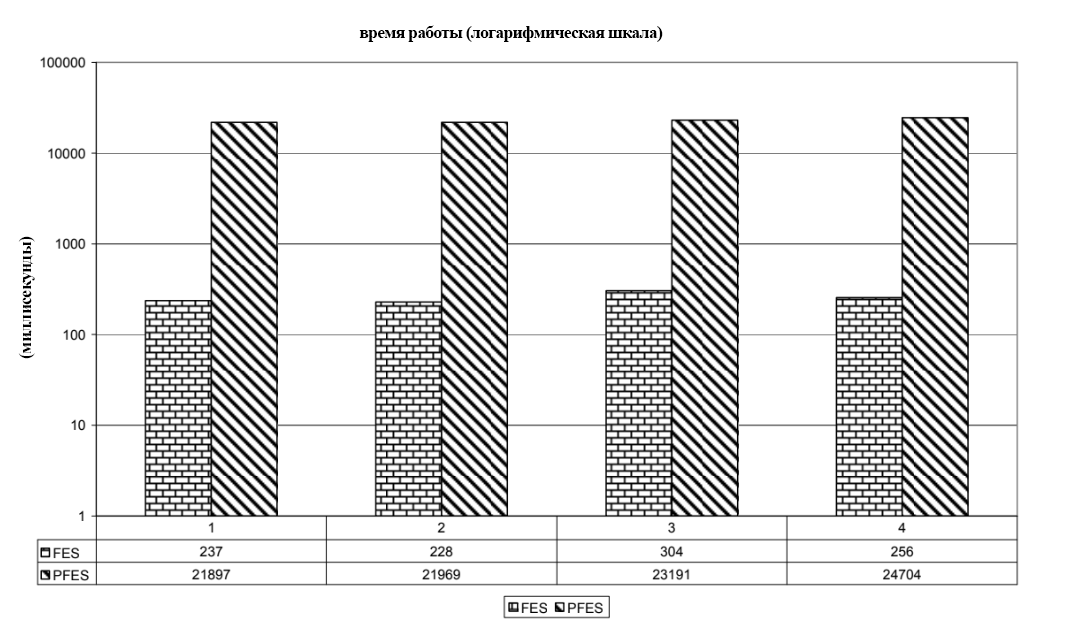
****

Рисунок 2 – Сравнение времени работы между FES и FPES с 5000 пакетами

Таблица 4 - Результаты моделирования с 2000 пакетами.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **FES** | | **PFES** | | **SA** | | **BDP** | |
| n | Время вып.  графика | Продолжительность | Время вып.  графика | Продолжительность | Время вып. графика | Продолжительность | Время вып. графика | Продолжительность |
| 2000 | 9954 | 96 | 9803 | 4654 | 19290 | 9720 | 11240 | 68529 |
| 2000 | 10412 | 77 | 10361 | 4994 | 19480 | 10144 | 11563 | 56806 |
| 2000 | 10355 | 57 | 10299 | 4985 | 19642 | 8723 | 11446 | 54158 |
| 2000 | 10329 | 83 | 19397 | 5913 | 19663 | 11076 | 11633 | 62643 |

Таблица 5 - Результаты моделирования с 5000 пакетами.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **FES** | | **PFES** | | **SA** | | **BDP** | |
| n | Время вып.  графика | Продолжительность | Время вып.  графика | Продолжительность | Время вып. графика | Продолжительность | Время вып. графика | Продолжительность |
| 5000 | 25626 | 236 | 25613 | 7299 | 29168 | 12584 | 47900 | 399820 |
| 5000 | 26129 | 228 | 28956 | 7323 | 29639 | 12272 | 49227 | 399636 |
| 5000 | 25910 | 304 | 25862 | 7730 | 29641 | 11606 | 48374 | 432436 |
| 5000 | 25768 | 256 | 25754 | 8235 | 29898 | 11492 | 48509 | 417261 |

**6 Обсуждение и будущая работа**

Очень важной областью в промышленности, где используются машины для пакетной обработки, является производство керамики (http://www.ceramicx.com). Многие различные компоненты обрабатываются с использованием одной и той же машины, например, печи, как указано в разделе 1. Другой типичный пример использования машины периодической обработки - для распределения молока или свежих продуктов. Небольшие транспортные средства обычно используются для раздачи молока или свежих продуктов с целью экономии расхода топлива.

Обсуждения с практиками некоторых из этих компаний выявили важные факты, которые должны быть включены в решение. Проблемы распределения продуктов питания отличаются от случая AGV, изученного в этой статье, некоторыми важными аспектами. Первое отличие заключается в том, что время в пути между местами не является постоянным. Временная переменная ti, связанная с каждым пакетом Pi, является непрерывной переменной с некоторым, часто предсказуемым, распределением. Распределение сильно зависит как от маршрутов, выбранных между производителем и потребителями, так и от времени суток, в течение которого маршрут проходит. Второе основное различие в распределении пищевых продуктов состоит в том, что время загрузки и выгрузки является переменным, и между партиями существует время наладки, необходимое для подготовки пакетов и загрузки их в транспортные средства. Эти переменные должны быть включены в наше будущее исследование. В отличие от случая с AGV, в случае раздачи продуктов заказы обычно повторяются периодически. В результате многие задачи планирования почти идеально повторяются. Периоды заказов обычно зависят от некоторых важных факторов, таких как время, цена за единицу продукции и некоторые внешние события, например, изменения условий поставки, стоимости топлива и т. д. Идентификация этих периодов позволяет производителям прогнозировать краткосрочные будущие потребности и помогает производителям создавать надежные производственные планы. Использование успешных графиков для аналогичных предыдущих случаев, включая незначительные изменения, может значительно сэкономить время при создании расписания. Рассуждения, основанные на конкретных случаях (CBR, [1]), подходят для подобных задач.

Были даны некоторые полезные комментарии о печной машине в керамической компании, в связи с особенностями этой машины. Установка в печной машине занимает значительное время, в результате чего сушильная машина должна быть прогрета до определенной температуры. Охлаждение температуры печи также занимает определенное время. Дополнительные факторы, связанные с выполнением задания, такие как входная температура и давление, должны учитываться в решении по планированию. Партии, состоящие из заданий, которые будут обрабатываться при одной и той же температуре, и давлении, должны обрабатываться вместе, чтобы сэкономить время на настройку и разгрузку печи.

В некоторых случаях для ускорения распространения услуг могут использоваться транспортные средства с небольшой вместимостью. Тем не менее, расход топлива значительно увеличивается, потому что транспортным средствам, возможно, придется путешествовать по более длинным маршрутам. Выбор подходящей вместимости автомобиля для экономии общих расходов на топливо. Эта проблема будет включена в наше будущее исследование.

**7 Обобщение**

В этой статье рассматривается проблема планирования для одного станка пакетной обработки, где учитываются ограничения производительности машины. Два алгоритма были представлены для решения случаев, когда задания (пакеты для транспортировки) не являются приоритетными и не идентичны по размеру или весу. Оба предложенных алгоритма являются полиномиальными и применимы для решения крупномасштабных задач. Экспериментальные результаты показывают существенное улучшение предложенных алгоритмов по сравнению с некоторыми другими опубликованными. Предполагаются некоторые упрощения. Например, все задания доступны в нулевое время, а их крайний срок по-прежнему опущен. Учет этих ограничений в задаче планирования является частью цели нашего будущего исследования. Следующий шаг нашего исследования заключается в планировании партий с минимальным временем завершения, при условии, что каждая работа связана с определенным сроком.

**Благодарность**

Эта работа была частично профинансирована словацкими национальными агентствами Vega по проекту № 2/4148/25 и APVV по контрактам № 51-024604 и 51-011602. Он также был частично поддержан Ирландским исследовательским советом по науке, технике и технологиям (IRCSET).

**Библиографический список**

[1] Aamodt, A.—Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications, IOS Press, Vol. 7, 1994, No. 1, pp. 39–59.

[2] Albers, S.— Brucker, P.: The Complexity of One-Machine Batching Problems. Discrete Applied Mathematics Vol. 47, 1993, pp. 87–107.

[3] Augustine, J. E.—Seiden, S. S.: Linear Time Approximation Schemes for Vehicle. Scheduling. Theoretical Computer Science, Vol. 324, 2004, Vol. 2–3, pp. 147–160.

[4] Baptiste, P.—LePape, C.— Nuijten, W.: Constraint-Based Scheduling. Kluwer Academic Publishers, Boston Hardbound, ISBN 0-7923-7408-8, 2001.

[5] Coffman, Jr. E. G.—Yannakakis, M.— Magazine, M. J.—Santos, C.: Batch Sizing and Sequencing on a Single Machine. Annals of operation research, Vol. 26, 1990, pp. 135–147.

[6] Cheraghi, S. H.—Vishwaram, V.—Krishnan, K. K.: Scheduling a Single Batch-Processing Machine with Disagreeable Ready Times and Due Dates. International Journal of Industrial Engineering, Vol. 10, 2003, No. 2.

[7] Karuno, Y.—Nagamochi, H.: 2-Approximation Algorithms for the Multi-Vehicle Scheduling Problem on a Path with Release and Handling Times. Discrete Applied Mathematics, Vol. 129, 2003, No. 2–3, pp. 433–447.

[8] Kasin, O.—Mason, S. J.: Scheduling Batch Processing Machines in Complex Job Shops. Proceedings of the 2001 Winter Simulation Conference, WSC ’01.

[9] Melouk, S.—Damodaran, P.—Chang, P. Y.: Minimizing Makespan for Single Machine Batch Processing with Non-Identical Job Sizes Using Simulated Annealing. Int. Journal of Production Economics, Vol. 87, 2004, pp. 141–147.

[10] Soner, H. M.—Touzi, N.: Stochastic Target Problems, Dynamic Programming, and Viscosity Solutions. SIAM Journal on Control and Optimization, Vol. 41, 2002, No. 2, pp. 404–424. Industrial Engineering Applications and Practice: User’s Encyclopedia. In CD, ISBN: 0-9654599-0-X.

[11] Papadaki, K.—Powel, W. B.: Exploiting Structure in Adaptive Dynamic Programming Algorithms for a Stochastic Batch Service Problem. European Journal of Operation Research, Vol. 142, 2002, No. 1, pp. 108–127.

[12] Simchi-Levi, D.: New Worst Case Results for the Bin Packing Problem. Naval Res. Logistics 41, 1994, pp. 579–585.

[13] Sung, C. S.—Choung, Y. I.: Minimizing Makespan on a Single Burn-In Oven in Semiconductor Manufacturing. European Journal of Production Research, Vol. 120, 2000, pp. 559–574.

[14] Sutton, R.—Barto, A.: Reinforcement Learning. The MIT Press, Cambridge, Massachusetts, 1998.

[15] Wagelmans, A. P. M.—Gerodimos, A. E. (2000): Improved Dynamic Programs for Some Batching Problems Involving the Maximum Lateness Criterion. Operations Research Letters 27, pp. 109–118. Using Heuristic Search for Solving Single Machine Batch Processing Problems 419

[16] Wang, J.—Luh, P. B.—Zhao, X.—Wang, J.: An Optimization-Based Algorithm for Job-Shop Scheduling. A Journal of Indian Academy of Sciences, A Special Issue on Competitive Manufacturing Systems, Vol. 22, 1997, No. 2, pp. 241–256.