# IOP Conference Series: Materials Science and Engineering

# Математическая модель для планирования машины пакетной обработки с несколькими несовместимыми семействами заданий, неидентичными размерами заданий, неидентичными параметрами заданий, неприемлемыми сроками выпуска и датами исполнения

**Аннотация**. В документе рассматривается проблема планирования машины пакетной обработки с несколькими несовместимыми семействами заданий, неодинаковыми размерами заданий, неидентичными параметрами заданий, и несовместимыми датами выпуска для минимизации продолжительности рабочего процесса. Задача исследования решается путем предложения модели смешанного целочисленного программирования, которая надлежащим образом учитывает параметры, рассматриваемые в задаче. Предложенное обосновано на численном примере. Проведенный эксперимент показывает, что модель может представлять значительные трудности при решении крупномасштабных случаев. В заключительной части документа представлены рамки для будущей работы и некоторые альтернативные подходы, которые можно использовать для решения этого класса проблем.

## 1. Введение

Планирование машин параллельной обработки в последнее время вызвало большой интерес у исследователей. Машины параллельной пакетной обработки по определению могут одновременно обрабатывать несколько заданий в виде пакета, в котором все задания имеют одинаковое время начала и окончания. Можно было бы найти эти проблемы, классифицированные как проблема планирования машин пакетной обработки (BPM), потому что размер партии ограничен в зависимости от производительности BPM.

В этой статье предпринята попытка решить одну из таких проблем, а именно составление расписания печи термообработки (HTF), которая находит применение в сталелитейной промышленности. Общей характеристикой сталелитейной промышленности является ассортимент продукции. Отливки (или работы) любого размера, формы, количества могут быть изготовлены. Кроме того, материалы, используемые в этом процессе, широко различаются и выбираются в зависимости от типа среды и применения работ, что делает их несовместимыми с точки зрения обработки. Таким образом, эти семьи рабочих мест называются множественными несовместимыми семьями рабочих мест. Сталелитейная промышленность является капиталоемкой отраслью, и конкуренция в этой отрасли высока. Таким образом, на сталелитейную промышленность оказывается давление, чтобы сократить сроки выполнения заказов, чтобы удовлетворить клиентов в срок. Одним из способов сокращения времени выполнения заказа является эффективное планирование машин с узкими местами.

С точки зрения пропускной способности и использования важных и дорогостоящих ресурсов в литейном производстве было высказано мнение о том, что управляемые процессом операции печи для операций плавки и разливки, а также операции печи термической обработки имеют решающее значение для удовлетворения общего объема производства. Эти две операции печи являются периодическими процессами, которые имеют отличительные ограничения на рабочие смеси в дополнение к обычной производительности и техническим ограничениям, связанным с любыми промышленными процессами. Преимущества эффективного планирования этих пакетных процессов включают более высокую загрузку оборудования, меньшую инвентаризацию незавершенного производства (WIP), более короткое время цикла и большую удовлетворенность потребителя. Хотя операции плавки и разливки имеют первостепенное значение, составление расписания печей для термообработки также имеет большое значение. Это связано, прежде всего, с тем, что время термообработки в печах часто больше по сравнению с любыми другими операциями в процессе. В этой статье предпринята попытка предложить методы, позволяющие сократить время выполнения этого процесса за счет использования этого узкого места.

Соответственно, в этой статье мы рассматриваем новую исследовательскую проблему планирования BP с несовместимыми семействами заданий (MIJF), NIJS, ​​NIJD и NARD, чтобы минимизировать время выполнения, которое является суррогатной мерой для увеличения пропускной способности и сокращения времени выполнения заказа.

## 2. Задача исследования

Задача планирования, определенная в этой статье, может быть обозначена в трех-полевом обозначении: α | β | γ как 1|pbatch, несколько несовместимых семейств работ, неодинаковые размеры заданий, неприемлемые сроки выпуска и сроки выполнения | Cmax проблема. Мы предлагаем математическую модель для задачи исследования со следующими допущениями:

* Должно быть обработано *n* заданий (*n* одиночных приведений), и эти задания принадлежат одной семье. Каждое задание имеет размер (с точки зрения веса) с параметрами - длина, - ширина и – высота
* Все работы должны проходить через BP
* Существует один BP, который имеет ограничение по пропускной способности с точки зрения размера (с точки зрения веса) и параметров - длина, - ширина и – высота
* BP доступен постоянно
* Ориентация рабочих мест не допускается
* Все данные , , , , , , и являются детерминированными и известны априори
* Предполагается, что ; ; ; и для всех рабочих мест *i*
* Предполагается, что рабочие места имеют несогласованные сроки выпуска и сроки исполнения. т.е. . не означает, что
* Предполагается, что время обработки семейства заданий не зависит от количества заданий в партии и является постоянным
* Время подготовки, если оно есть, включено во время обработки, и при переключении с одной партии на другую нет времени на установку
* После начала обработки пакета BP не может быть прервана, и другие задания не могут быть введены в BP, пока текущая обработка не будет завершена.
* Поломки машины не учитываются

Работа организована следующим образом: в разделе 3 мы рассмотрим соответствующую работу. В разделе 4 мы предлагаем математическую модель. В разделе 5 мы проверяем математическую модель на численном примере. Мы завершаем работу с кратким описанием и возможным будущим направлением исследований в разделе 6.

## 3. Обзор связанных работ:

Полупроводниковая отрасль наиболее заинтересовала исследователей [10], в то время как существует лишь несколько исследований, близких к рассматриваемой исследовательской проблеме. Это может быть связано прежде всего с экономической и технологической важностью, связанной с полупроводниковой промышленностью.

В [5] рассмотрена проблема планирования BPM с несовместимыми семействами заданий с целью минимизации времени выполнения, общего времени завершения и общего взвешенного времени завершения. Авторы предлагают математическую модель, набор эвристических алгоритмов и метаэвристику для различных целей.

Помимо этого исследования, другие исследования фокусируются на метаэвристических подходах для решения задач планирования BPM [4], [11], [2].

Наши исследования отличаются от других исследований тем, что мы рассматриваем важный аспект, а именно неидентичные рабочие параметры, которые могут быть не актуальны в полупроводниковой промышленности, но важны в сталелитейной промышленности. Кроме того, наше исследование рассматривает несогласованные времена выпуска, которые не рассматриваются в литературе. Наконец, в отличие от исследований, которые предлагают метаэвристику, наше исследование пытается разработать математическую модель, которая дала бы оптимальное решение для исследовательской задачи.

Помимо полупроводниковой промышленности, есть только несколько исследований, которые направлены на решение проблемы планирования печи для термообработки. Таким образом, мы концентрируемся только на этих нескольких исследованиях, которые очень близки к проблеме исследования.

[8] исследовали проблему планирования нескольких печей для термообработки (HTF) для сталелитейной промышленности с неидентичным размером работы и допущением идентичных параметров работы для максимизации использования процессоров периодического действия при динамическом прибытии работы с приемлемым временем выпуска и из-за даты. Они используют трехэтапный подход к планированию печей термообработки, а именно выбор BPM, выбор семейства работ. Они проводят серию вычислительных экспериментов, в которых качество решения эвристических алгоритмов сравнивается с оцененными оптимальными решениями для каждой проблемы. Они пришли к выводу, что эвристические алгоритмы, основанные на критериях выбора семейства работ, которые используют: (а) простой средний приоритет работы, (б) простой средний размер работы и (в) средневзвешенный размер работы, выполняются лучше при меньших вычислительных затратах.

[9] изучили проблему планирования, рассмотренную ранее [8], с целью минимизации общего взвешенного опоздания. Они предложили четыре жадных эвристических алгоритма, которые различаются в зависимости от выбора семейства работ. Проведенные ими подробные вычислительные эксперименты показывают, что в среднем эвристические алгоритмы, использующие критерии выбора семейства работ, основанные на (а) совокупной дате выполнения и (b) взвешенной совокупной дате выполнения с размером задания в качестве веса, работают лучше по сравнению со статистическими процедурами оценки.

Наше исследование отличается от этих исследований тем, что мы рассматриваем неидентичные рабочие параметры, которые приближают наше исследование к реальности.

## 4. Разработка математической модели для планирования BP с MIJF, NIJS и NIJD

Проблему планирования BPM с MIJF, NIJD, NIJS и NARD можно рассматривать как продолжение трехмерной задачи упаковки бинов.

Существуют различные подходы к решению такого типа исследовательской задачи, такие как математические модели, эвристические методы или метаэвристические методы. Так как из всех этих трех математические модели обеспечивают оптимальные решения проблемы, эта статья подходит к проблеме, предлагая математическую модель.

Математические модели, предложенные различными исследователями ([3], [12], [1]) для задачи упаковки бинов, явно рассматривают характеристики NIJD, и нет никаких ограничений на емкость бинов с точки зрения размеров работ, рассматриваемых в этих исследованиях.

Планирование BPM, однако, является более сложным. Во-первых, существует несколько несовместимых рабочих семей, что означает, что рабочие места из одной семьи не могут быть совмещены с работой из другой семьи. Кроме того, даже в пределах одной семейной работы размеры работы (отливки) варьируются в широких пределах (от 10 г до 1000 кг), что накладывает огромное ограничение на количество рабочих мест, которые могут быть размещены в партии, чтобы удовлетворить ограничения производительности BP в условиях размера, в дополнение к параметрам работы. Соответственно, среди математических моделей, доступных в литературе для решения проблемы упаковки бина, модель MILP, предложенная в [1], соответствующим образом модифицирована для решения проблемы планирования BP с ограничением несовместимости и ограничениями емкости как по размерам, так и по параметрам заданий. Предложенная модель MILP для планирования BP с MIJF, NIJD, NIJS и NARD выглядит следующим образом:

Индексы

*i, k* – 1, 2, 3, …, n Работы

*j* – 1,2,3,…,m Пакеты

*p* – 1,2,3,…,n Семьи

Параметры

M - Большое произвольное число

( , , ) - Параметры, указывающие длину, ширину и высоту задания *i*

- Размер работы *i*

(*L*, *W*, *H*) - Длина, ширина и высота BP

*S* - Размер емкости BP

- Время обработки задания *i*

- Время выпуска работы *i*

Непрерывные переменные решения

(, , ) - координаты переднего левого нижнего угла задания *i*. Предполагается, что начало координат зафиксировано в переднем левом нижнем углу BP.

(, , ) - координаты переднего левого нижнего угла задания *k*. Предполагается, что начало координат зафиксировано в переднем левом нижнем углу BP.

- время завершения последней партии

- время завершения работы *i*

- время выпуска партии *j*

- время обработки партии *j*

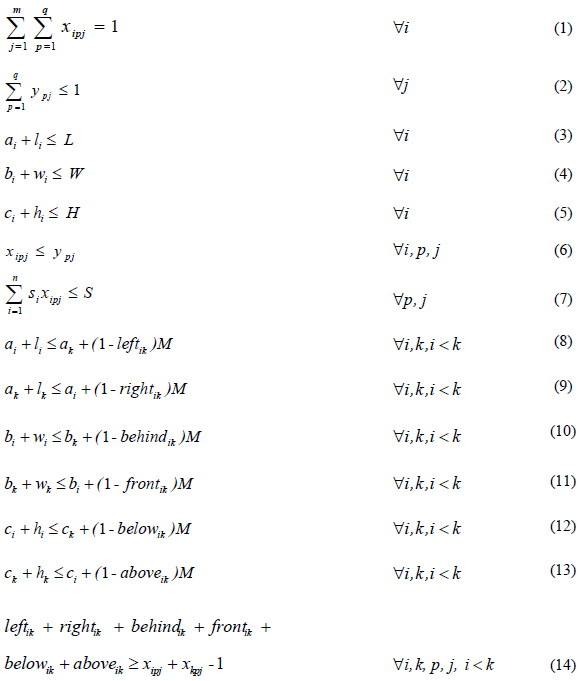
- время завершения партии *j*

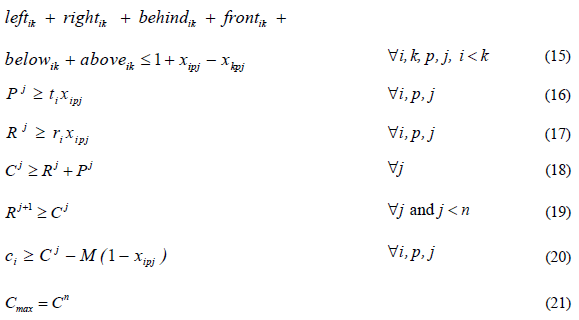
Двоичные переменные решения

Двоичные переменные решения

Min

При условии:





Двоичное ограничение:

Неотрицательные ограничения:

Целью модели является минимизация времени завершения последней партии. Набор ограничений (1) гарантирует, что каждое задание семейства заданий назначается пакету только один раз. Набор ограничений (2) предусматривает, что для партии может быть назначено не более одного семейства. Ограничения (3) - (5) гарантируют, что все задания, назначенные для пакета, соответствуют физическим измерениям BP. Ограничение (6) гарантируют, что задание из семейства заданий назначается для пакета тогда и только тогда, когда это семейство заданий присваивается этой партии. Ограничение (7) гарантирует, что грузоподъемность BP не будет превышена. Ограничения (8–13) гарантируют, что задания не пересекаются друг с другом. Эта проверка на перекрытие необходима, только если пара заданий размещена в одном и том же BP. Об этом заботятся Ограничения (14–15).

Ограничение (16) определяет время обработки пакета как большее или равное времени обработки каждого задания (= время обработки семейства заданий) в этом пакете. Ограничение (17) гарантирует, что время выпуска пакета является максимальным из времени выпуска всех заданий в пакете. Расчет времени выполнения работ выполняется (18). Обеспечение того, чтобы время выполнения этих заданий равнялось времени завершения предыдущего пакета (19). Ограничение (20) гарантирует, что время завершения задания, по крайней мере, равно времени завершения пакета, если и только это задание находится в этом пакете. Ограничение (21) - это время завершения последней партии.

Число бинарных переменных в предлагаемой модели MILP равно *nm + fm + 3n(n-1),* а модель имеет *3n(n-1) + 2n(n-1)m + 4nfm + 5n + 3m + fm + 1* количество ограничений, где *f* - количество семейств рабочих мест, *n* - количество рабочих мест, *m* - количество ограничений.

## **5. Проверка предложенной модели MILP**

Чтобы проверить предложенную модель MILP, мы разработали числовой пример, предполагая, что один BP с двумя семействами заданий имеет n = 10 заданий со временем обработки первого семейства заданий, равным 15 часам, а второго - 12 часов. Емкость BP по размерам BP = 1000 кг; и по параметрам(измерениям): L = 1500 мм; W = 950 мм; H = 900 мм. Кроме того, размер, параметры и время обработки 10 заданий приведены в таблице 1.

Оптимальное решение для предложенной модели MILP получено с помощью LINGO. Соответствующий код набора LINGO для генерации приведен в Приложении 1. Используя этот код набора, модель ILP генерируется и решается для числового примера, представленного в Таблице 1.

Таблица 1: Численный пример для задачи исследования

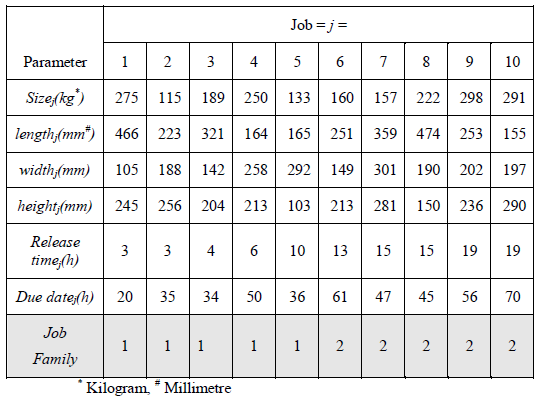
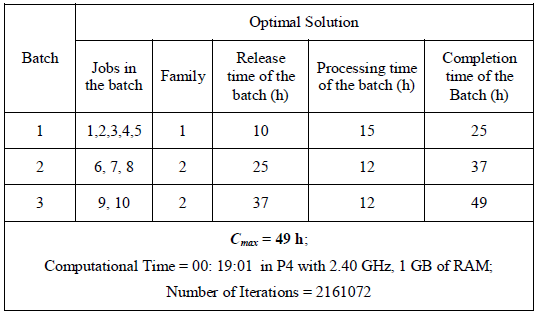


Таблица 2: Оптимальное решение, полученное с использованием предложенной модели MILP



Оптимальное решение, полученное для числового примера, представлено в таблице 2, в которой приведены подробные сведения о задании партии, сведениях о семействе заданий, времени выпуска, времени обработки и времени завершения каждой партии.

## 6. Резюме и будущая работа

Основываясь на наших наблюдениях от сталелитейной промышленности, мы обратились к важной исследовательской проблеме планирования HTF в сталелитейной промышленности. Этой проблеме не было уделено должного внимания в литературе. Это привело нас к разработке математической модели для планирования HTF с MIJF, NIJD, NIJS и NARD с целью минимизации рабочего времени. Предложенная математическая модель проверена на числовом примере путем генерации и решения модели в LINGO.

Можно отметить, что даже для мелкомасштабного экземпляра, такого как тот, который используется для решения математической модели в этой статье, время вычислений достаточно велико. В целом, общепризнанно, что целочисленные модели программирования трудно решить с оптимальностью, особенно для больших случаев. Таким образом, возможно, что более крупные экземпляры могут представлять значительные трудности при решении модели, предложенной в этой статье. Принимая во внимание эти вещи, можно сначала тщательно протестировать модель в больших масштабах, чтобы оценить время вычислений для этого класса моделей. Во-вторых, можно подумать о переформулировании этой модели, чтобы ее можно было решить в разумные сроки.

В качестве альтернативы, проблема может быть решена с помощью эвристических методов, которые могут очень быстро привести к почти оптимальным решениям. Кроме того, могут быть разработаны метаэвристические методы, такие как генетические алгоритмы, поиск по табу и т. д., для решения проблемы, которые могут обеспечить лучшие решения, чем простой эвристический метод, за счет больших вычислительных усилий.

# Приложение 1

SETS :

FAMILIES:STARTIME;

JOBS:P,Q,HT,WEIGHT,X,Y,Z,RELTIME,DUEDATE,PROCTIME,COMPTIMEJOB;

FNCS:CTIME,FNCHEIGHT,FNCCAPACITY,FNCLENGTH,FNCWIDTH,PROCTIMEBAT,RELTIMEBAT,

COMPTIMEBAT;

ORIENTATION(JOBS,JOBS)|&1 #LT# &2:LEFT,RIGHT,BEHIND,INFRONT,BELOW,ABOVE;

FAMTOFNCS(FAMILIES,FNCS):FTOB;

JOBTOFAMTOFNC(JOBS,FAMILIES,FNCS):JTOFTOB;

ENDSETS

DATA :

FAMILIES = F1,F2 ;

JOBS,FNCS,P,Q,HT,WEIGHT,FNCLENGTH,FNCWIDTH,PROCTIME,RELTIME,DUEDATE,FNCHEIGHT

,FNCCAPACITY = @OLE ('C:\\3DINCOMPCH2\\12jobs.XLS');

ENDDATA

BIGM=1500;

MAXJOBS = @SIZE (FNCS);

!OBJECTIVE TO MINIMIZE MAKESPAN;

MIN = CMAX;

!SUBJECT TO CONSTRAINTS:

!ASSIGN EACH JOB TO ONE BATCH;

@FOR (JOBS(I):

@SUM (FAMILIES(FAM):

@SUM (FNCS(B): JTOFTOB(I,FAM,B)))=1);

!AT MOST ONE JOB FAMILY CAN BE ASSINGED TO A BATCH;

@FOR (FNCS(B):

@SUM (FAMILIES(FAM): FTOB(FAM,B))<=1);

! A JOB FROM JOB FAMILY IS ASSINGED TO A BTATCH ONLY IF

THE JOB FAMILY IS ASSIGNED TO A BATCH;

@FOR (JOBS(I):

@FOR (FAMILIES(FAM):

@FOR (FNCS(B): JTOFTOB(I,FAM,B) <= FTOB(FAM,B))));

!SIZE CONSTRAINT;

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@SUM (JOBS(I): WEIGHT(I)\*JTOFTOB(I,FAM,J)) <= 1000));

!ORIENTATION OF JOBS CONSTRAINT;

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: X(I)+ P(I) <= X(K)+(1-LEFT(I,K))\*BIGM) );

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: X(K)+ P(K) <= X(I)+(1-RIGHT(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Y(I)+ Q(I)<= Y(K)+(1-BEHIND(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Y(K)+Q(K) <= Y(I)+(1-INFRONT(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Z(I)+ HT(I) <= Z(K)+(1-BELOW(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Z(K)+ HT(K) <= Z(I)+(1-ABOVE(I,K))\*BIGM ));

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K:

LEFT(I,K)+RIGHT(I,K)+BEHIND(I,K)+INFRONT(I,K)+BELOW(I,K)+ABOVE(I,K)

>= JTOFTOB(I,FAM,J)+JTOFTOB(K,FAM,J)-1))));

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K:

LEFT(I,K)+RIGHT(I,K)+BEHIND(I,K)+INFRONT(I,K)+BELOW(I,K)+ABOVE(I,K)

<= 1+JTOFTOB(I,FAM,J)-JTOFTOB(K,FAM,J)))));

!LENGTH CONSTRAINT;

@FOR (JOBS(I): X(I)+ P(I)<= 1500);

!WIDTH CONSTRAINT;

@FOR (JOBS(I): Y(I)+ Q(I)<= 950);

!HEIGHT CONSTRAINT;

@FOR (JOBS(I): Z(I)+ HT(I)<= 900);

!CALCULATE PROCESSING TIME OF A BATCH;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): PROCTIMEBAT(B) >= PROCTIME(J)\* JTOFTOB(J,FAM,B))));

!RELEASE TIME OF A BATCH IS MAXIMUM OF ALL RELEASE TIMES OF JOBS IN A BATCH;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): RELTIMEBAT(B) >= RELTIME(J)\* JTOFTOB(J,FAM,B))));

!CALCULATE COMPLETION TIME OF A BATCH;

@FOR (FNCS(B): COMPTIMEBAT(B) = RELTIMEBAT(B)+PROCTIMEBAT(B) );

!RELEASE TIME OF A BATCH SHOULD BE LESS THAN THE COMPLETION TIME OF

THE PREVIOUS BATCH;

@FOR (FNCS(B)|B #LT# @SIZE (FNCS): RELTIMEBAT(B+1) >= COMPTIMEBAT(B) );

!CALCULATE COMPLETION TIME OF A JOB;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): COMPTIMEJOB(J) >= COMPTIMEBAT(B)-1000\*(1-JTOFTOB(J,FAM,B)))));

!COMPLETION TIME OF A JOB IS LESS THAN ITS DUE DATE;

!@FOR(JOBS(J): COMPTIMEJOB(J) <= DUEDATE(J));

!CALCULATE CMAX;

@FOR (FNCS(B)|B #EQ# @SIZE (FNCS): CMAX = COMPTIMEBAT(B) );

!DECISION VARIABLES;

@FOR (JOBS(I)|I #GE# 1 #AND# I #LE# 5:

@FOR (FAMILIES(FAM)|FAM #EQ# 1:

@for (FNCS(B): @bin (JTOFTOB(I,FAM,B)))));

@FOR (JOBS(I)|I #GE# 6 :

@FOR (FAMILIES(FAM)|FAM #EQ# SETS :

FAMILIES:STARTIME;

JOBS:P,Q,HT,WEIGHT,X,Y,Z,RELTIME,DUEDATE,PROCTIME,COMPTIMEJOB;

FNCS:CTIME,FNCHEIGHT,FNCCAPACITY,FNCLENGTH,FNCWIDTH,PROCTIMEBAT,RELTIMEBAT,

COMPTIMEBAT;

ORIENTATION(JOBS,JOBS)|&1 #LT# &2:LEFT,RIGHT,BEHIND,INFRONT,BELOW,ABOVE;

FAMTOFNCS(FAMILIES,FNCS):FTOB;

JOBTOFAMTOFNC(JOBS,FAMILIES,FNCS):JTOFTOB;

ENDSETS

DATA :

FAMILIES = F1,F2 ;

JOBS,FNCS,P,Q,HT,WEIGHT,FNCLENGTH,FNCWIDTH,PROCTIME,RELTIME,DUEDATE,FNCHEIGHT

,FNCCAPACITY = @OLE ('C:\\3DINCOMPCH2\\12jobs.XLS');

ENDDATA

BIGM=1500;

MAXJOBS = @SIZE (FNCS);

!OBJECTIVE TO MINIMIZE MAKESPAN;

MIN = CMAX;

!SUBJECT TO CONSTRAINTS:

!ASSIGN EACH JOB TO ONE BATCH;

@FOR (JOBS(I):

@SUM (FAMILIES(FAM):

@SUM (FNCS(B): JTOFTOB(I,FAM,B)))=1);

!AT MOST ONE JOB FAMILY CAN BE ASSINGED TO A BATCH;

@FOR (FNCS(B):

@SUM (FAMILIES(FAM): FTOB(FAM,B))<=1);

! A JOB FROM JOB FAMILY IS ASSINGED TO A BTATCH ONLY IF

THE JOB FAMILY IS ASSIGNED TO A BATCH;

@FOR (JOBS(I):

@FOR (FAMILIES(FAM):

@FOR (FNCS(B): JTOFTOB(I,FAM,B) <= FTOB(FAM,B))));

!SIZE CONSTRAINT;

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@SUM (JOBS(I): WEIGHT(I)\*JTOFTOB(I,FAM,J)) <= 1000));

!ORIENTATION OF JOBS CONSTRAINT;

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: X(I)+ P(I) <= X(K)+(1-LEFT(I,K))\*BIGM) );

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: X(K)+ P(K) <= X(I)+(1-RIGHT(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Y(I)+ Q(I)<= Y(K)+(1-BEHIND(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Y(K)+Q(K) <= Y(I)+(1-INFRONT(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Z(I)+ HT(I) <= Z(K)+(1-BELOW(I,K))\*BIGM ));

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K: Z(K)+ HT(K) <= Z(I)+(1-ABOVE(I,K))\*BIGM ));

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K:

LEFT(I,K)+RIGHT(I,K)+BEHIND(I,K)+INFRONT(I,K)+BELOW(I,K)+ABOVE(I,K)

>= JTOFTOB(I,FAM,J)+JTOFTOB(K,FAM,J)-1))));

@FOR (FAMILIES(FAM):

@FOR (FNCS(J):

@FOR (JOBS(K):

@FOR (JOBS(I)|I #LT# K:

LEFT(I,K)+RIGHT(I,K)+BEHIND(I,K)+INFRONT(I,K)+BELOW(I,K)+ABOVE(I,K)

<= 1+JTOFTOB(I,FAM,J)-JTOFTOB(K,FAM,J)))));

!LENGTH CONSTRAINT;

@FOR (JOBS(I): X(I)+ P(I)<= 1500);

!WIDTH CONSTRAINT;

@FOR (JOBS(I): Y(I)+ Q(I)<= 950);

!HEIGHT CONSTRAINT;

@FOR (JOBS(I): Z(I)+ HT(I)<= 900);

!CALCULATE PROCESSING TIME OF A BATCH;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): PROCTIMEBAT(B) >= PROCTIME(J)\* JTOFTOB(J,FAM,B))));

!RELEASE TIME OF A BATCH IS MAXIMUM OF ALL RELEASE TIMES OF JOBS IN A BATCH;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): RELTIMEBAT(B) >= RELTIME(J)\* JTOFTOB(J,FAM,B))));

!CALCULATE COMPLETION TIME OF A BATCH;

@FOR (FNCS(B): COMPTIMEBAT(B) = RELTIMEBAT(B)+PROCTIMEBAT(B) );

!RELEASE TIME OF A BATCH SHOULD BE LESS THAN THE COMPLETION TIME OF

THE PREVIOUS BATCH;

@FOR (FNCS(B)|B #LT# @SIZE (FNCS): RELTIMEBAT(B+1) >= COMPTIMEBAT(B) );

!CALCULATE COMPLETION TIME OF A JOB;

@FOR (FAMILIES(FAM):

@FOR (FNCS(B):

@FOR (JOBS(J): COMPTIMEJOB(J) >= COMPTIMEBAT(B)-1000\*(1-JTOFTOB(J,FAM,B)))));

!COMPLETION TIME OF A JOB IS LESS THAN ITS DUE DATE;

@FOR (JOBS(J): COMPTIMEJOB(J) <= DUEDATE(J));

!CALCULATE CMAX;

@FOR (FNCS(B)|B #EQ# @SIZE (FNCS): CMAX = COMPTIMEBAT(B) );

!DECISION VARIABLES;

@FOR (JOBS(I)|I #GE# 1 #AND# I #LE# 5:

@FOR (FAMILIES(FAM)|FAM #EQ# 1:

@for (FNCS(B): @bin (JTOFTOB(I,FAM,B)))));

@FOR (JOBS(I)|I #GE# 6 :

@FOR (FAMILIES(FAM)|FAM #EQ# 1:

@for (FNCS(B): JTOFTOB(I,FAM,B)=0)));

@FOR (JOBS(I)|I #GE# 6 :

@FOR (FAMILIES(FAM)|FAM #EQ# 2:

@for (FNCS(B): @bin (JTOFTOB(I,FAM,B)))));

@FOR (JOBS(I)|I #LE# 5 :

@FOR (FAMILIES(FAM)|FAM #EQ# 2:

@for (FNCS(B): JTOFTOB(I,FAM,B)=0)));

@FOR (FAMTOFNCS: @BIN (FTOB));

@FOR (ORIENTATION: @BIN (LEFT));

@FOR (ORIENTATION: @BIN (RIGHT));

@FOR (ORIENTATION: @BIN (BEHIND));

@FOR (ORIENTATION: @BIN (INFRONT));

@FOR (ORIENTATION: @BIN (BELOW));

@FOR (ORIENTATION: @BIN (ABOVE));

@for (FNCS(B): JTOFTOB(I,FAM,B)=0)));

@FOR (JOBS(I)|I #GE# 6 :

@FOR (FAMILIES(FAM)|FAM #EQ# 2:

@for (FNCS(B): @bin (JTOFTOB(I,FAM,B)))));

@FOR (JOBS(I)|I #LE# 5 :

@FOR (FAMILIES(FAM)|FAM #EQ# 2:

@for (FNCS(B): JTOFTOB(I,FAM,B)=0)));

@FOR (FAMTOFNCS: @BIN (FTOB));

@FOR (ORIENTATION: @BIN (LEFT));

@FOR (ORIENTATION: @BIN (RIGHT));

@FOR (ORIENTATION: @BIN (BEHIND));

@FOR (ORIENTATION: @BIN (INFRONT));

@FOR (ORIENTATION: @BIN (BELOW));

@FOR (ORIENTATION: @BIN (ABOVE));