Математическая модель расписания.

Расписание:

1. Коллекция заказов *i*, в которых для данного расписания характерны *Weight(i)*, *V(i)*, *M(i)*,*<list>OrderParts(i)*, *V(i, m)*, *k(i,m)* - ЛА для части заказа, *PlantSGP(i, m)*, времена начала и окончания исполнения части заказа *(i,m)*, число ходок *nCasts(i, m)*, *map<nCont(i, plant, cont)*.
2. Коллекция электролизеров *<list>E(cast)*, из которых поступает сырец для каждой ходки: *<map>Electrcast<cast, <list>E(VE(cast, electr))>*. Второй список в map содержит объекты объемы, забираемые из электролизера, вместе с их идентификаторами. (опциональный элемент расписания, не являющийся директивным для расписания)
3. Расписание миксеров. См. 4.
4. Расписание ЛМ. Для каждой ЛМ вводим *<map>LM\_Availability<LM, <list>TimeframeLM(timeframe)>*. *Timeframe* - промежуток времени, у которого назначены время начала *tbegin*, время конца *tend* промежутка, тип работ в этот промежуток (ENUM: *FLUSHING, RETOOLING, ORDER\_PROCESSING, PERIODIC\_OPERATION\_ID*), ссылка на сущность, для которой выполняется работа. Все timeframe в коллекции упорядочены про времени начала *tbegin*. *tend(i)* <= *tbegin(i+1)* в рамках коллекции. Ссылка на сущность - *(i, m)*.
5. Расписание фильтра. См. 4. + оставшийся ресурс фильтра.
6. Расписание оснастки. См. 4 + оставшийся ресурс оснастки.
7. Расписание линий гомогенизации. См. 4.
8. Расписание линий резки. См. 4.
9. Запрос о транспорте. *<map>nCont(i, plant, cont, date)* - коллекция, которая определяет ежедневную потребность завода в контейнерах определенного типа.

Построение первого расписание (идея основана на информации, полученной от технологов):

1. Построение первого расписания основано на жадном алгоритме назначения заказов на ЛА с учетом совпадения химии и оснасток заказов.
2. Presolve построения первого расписания:
   1. Заказы сортируются по Form(prod).
   2. Заказы сортируются по маркам mark(Prod) внутри сортировки по продуктам.
   3. Внутри групп заказов с одинаковой маркой производится сортировка заказов по параметрам оснастки ЛА Diameter(Order) или Width(Order) и Height(Order). Заказы, попавшие в одну группу на 3-й сортировке, будут назначены на один ЛА в виду возможности их объединения в промежуточные ходки и отсутствия технологических операций для перехода между заказами.
3. Presolve: для каждого заказа рассчитываем массу заказа как масса, умещающаяся в целом числе контейнеров, но меньшая чем масса заказа + толеранс.
4. Выбирается ЛА, который имеет время доступности (освобождения) минимальное из возможных. Для такого ЛА выбираем 2-3 (результат тестирования алгоритма) группы, которые можно выполнить на ЛА, полученные в результате шага 2.c с минимальным временем max(tпереоснастка, суммапо миксерам и коптильникам tпромывки) и назначем группы на выбранный ЛА, создавая различные расписания.  
   Рассчитываем количество ходок на суммарный объем заказов группы: с объемом заказа минус толеранс, объемом заказа, объемом заказа плюс толеранс. Берем максимальое количество целого количества ходок, либо максимальное число, если целого количества ходок нет.  
   Рассчитываем время обработки заказов на ЛА по ходкам: для каждой ходки определяем время занятости миксеров, копильников, литейных машин, линий гомогенизации, линий резки.
5. Для каждого полученного расписания переходим к пункту 3, если для расписания еще присутствуют неназначенные группы заказов.
6. При невозможности найти время назначения заказа на ЛА происходит перестроение расписания с учетом назначения текущего заказа первым.
7. Postsolve построения первого поколения расписаний. Помечаем заказы, назначенные на один ЛА, относящиеся к одной группе, как заказы для которых невозможна перестановка.  
   Проводим проверку на выполнимость расписания. Если оно занимает время, выходищее за рамки планируемого срока, проводим уточнение количества ходок для каждого заказа. Начиная с самого дешевого заказа уменьшаем количество ходок до объем заказа минус толеранс, пока не получим расписание, выполнимое в расчитываемый период.  
   Коррекция количества ходок, необходимости в транспорте.

Построение перестановочного расписания:

1. Для каждой группы заказов в рамках текущего расписания считаем время перехода с предыдущей группы на текущую и с текущей на следующую как сумму штрафа за просрочку, затраты на промывку и чистку миксеров, затраты на промывку фильтров. (минимальное время между временем переоснастки и временем, получаемым суммированием времен промывки на всех агрегатах).
2. Сортируем группы заказов в рамках данного расписания по стоимости затрат.
3. Делаем перестановки групп заказов. В перестановках учувствуют группы заказов с наибольшей стоимостью перехода к производству текущего заказа (для начала можно взять 10% заказов, точное значение стоит уточнить при тестировании).
4. Смотрим заказ из очереди на перестановку. Смотрим, можем ли мы менять заказ с другими. Для этого текущий заказ пытаемся поставить на место заказа следующий за ним в нашей отсортированной очереди. Если перестановка сдвигает следующие заказы – сдвигаем их с учетом передвижения.  
   Рассчитываем ограничения получившегося расписания и применяем перестановку или переходим к следующей группе заказов, откатывая перестановку.  
   Если не получилось передвинуть заказ – переходим к следующему в очереди перестановки.

Алгоритм расчета оптимального расписания:

1. Строим расписания – аналог расписания технологов.
2. Из расписания, полученного в шаге 1, строим 50 (определить по результатам тестирования) расписаний перестановками.
3. При достижении 50 при переходе к генерации новых расписаний оставляем только 50 с наилучшим значением функции маржи.

Цель алгоритма – проверка возможности улучшения первоначального расписания с помощью перестановок заказов, требующих затратных переходов на производство текущего заказа.