Оглавление

[Генетический алгоритм "Умное расписание" 2](#_Toc207618392)

[1. Общее описание 2](#_Toc207618393)

[2. Учитываемые факторы (Иерархия правил) 2](#_Toc207618394)

[Уровень 0 и 1: Жесткие ограничения (Фильтры валидации) 2](#_Toc207618395)

[Уровень 2, 3, 4: Цели и Оптимизация (Оцениваются Функцией приспособленности) 2](#_Toc207618396)

[3. Ключевые формулы и фазы 2](#_Toc207618397)

[3.1. Формулы и этапы подготовительной фазы «Фаза 1» (Подготовка данных и Прогнозирование. 2](#_Toc207618398)

[Этап 1. Процесс подготовки информации: 3](#_Toc207618399)

[Этап 2. Прогнозирование спроса: 3](#_Toc207618400)

[Этап 3. Прогнозирование Финансовых показателей: 4](#_Toc207618401)

[Этап 4. Модификация показателей: 5](#_Toc207618402)

[3.2. Формулы и этапы фазы генерации случайных расписаний «Фаза 2» (Генерация первого поколения расписаний). 5](#_Toc207618403)

[Процесс генерации одного случайного, но валидного расписания ("Хромосомы") 5](#_Toc207618404)

[Почему этот подход эффективен? 7](#_Toc207618405)

[3.3. Формулы и этапы фазы оценки поколения «Фаза 3» (Оценка первого поколения расписаний). 7](#_Toc207618406)

[Шаг 0: Инициализация полного списка счетчиков 7](#_Toc207618407)

[Шаг 1: Итеративный проход по каждому "Гену" 8](#_Toc207618408)

[Шаг 2: Финальные расчеты по Хромосоме в целом 10](#_Toc207618409)

[Шаг 3: Вычисление итоговой оценки расписания на месяц 13](#_Toc207618410)

[Шаг 4: Получение оптимального расписания (Эволюция) 14](#_Toc207618411)

[4. Пример вычислений 17](#_Toc207618412)

[Шаг 1: Предварительные расчеты (Фаза 1) 17](#_Toc207618413)

[Шаг 2: Генерация и оценка двух конкурирующих "Хромосом" 18](#_Toc207618414)

[Шаг 3: Расчет оценки для каждой Хромосомы 18](#_Toc207618415)

[Шаг 4: Эволюция — от Поколения 1 к Поколению 2 21](#_Toc207618416)

[Шаг 5: Выход на Плато (Поколения 3, 4, 5...) 22](#_Toc207618417)

[5. Что требуется для реализации: 23](#_Toc207618418)

# **Генетический алгоритм "Умное расписание"**

# **1. Общее описание**

Алгоритм работает по принципу **эволюционного поиска**: он создает "популяцию" из тысяч различных вариантов расписания ("Хромосом"), оценивает их эффективность с помощью комплексной **Функции приспособленности**, отбирает лучшие решения ("родителей") и комбинирует их для создания еще более совершенного "потомства". Этот процесс повторяется сотни раз, пока не будет найдено близкое к идеальному решение.

# **2. Учитываемые факторы (Иерархия правил)**

Алгоритм оперирует многоуровневой системой правил, от незыблемых до гибких.

## **Уровень 0 и 1: Жесткие ограничения (Фильтры валидации)**

Любой вариант расписания, нарушающий эти правила, немедленно отбраковывается.

* **Графики:** соблюдение рабочих смен, отпусков и часов работы кабинетов.
* **Структура смен:** врачи работают в фиксированных сменах (утро/вечер).
* **Расписание "звёзд":** их график заморожен и не подлежит оптимизации.
* **Соответствие "Требуемая-Имеющаяся специализация":** врач может работать только в кабинете, специализация которого явно разрешена для его текущей специальности (напр., Педиатр не может работать в кабинете Гинекологии).
* **"Жесткие" привязки к кабинетам:** являются правилом-исключением, которое переопределяет ограничения по специализации.

## **Уровень 2, 3, 4: Цели и Оптимизация (Оцениваются Функцией приспособленности)**

Эти факторы определяют "качество" валидного расписания.

* **Спрос (детализированный):** на общие слоты, ДМС, конкретные услуги (номенклатура).
* **Финансы:** Прогнозируемая совокупная выручка от услуг и косвенный доход.
* **Надежность:** Приоритет для врачей с исторически высоким коэффициентом заполняемости.
* **Стратегия:** Бонусы за врачей с признаками ДМС, Выезд, Выписка БЛ.
* **Персонал:** "Фора" для новых врачей, баланс нагрузки для опытных, учет "мягких" привязок к кабинетам.

# **3. Ключевые формулы и фазы**

## **3.1. Формулы и этапы подготовительной фазы «Фаза 1» (Подготовка данных и Прогнозирование.**

**Примечание:** *Этот этап выполняется один раз в месяц перед запуском основного цикла эволюционного поиска расписания и состоит из трех ключевых шагов: Сбор и структурирование, Прогнозирование и Модификация.*

**Этап 1. Процесс подготовки информации:**

1. **Извлечение:** настраиваются скрипты для автоматической выгрузки данных из 1С УМЦ
2. **Преобразование:** Самая важная часть.
   * **Очистка:** удаляются дубликаты, тестовые записи, исправляются очевидные ошибки (например, прием с нулевой стоимостью).
   * **Унификация:** Все идентификаторы приводятся к единому формату (или используются из 1С) (Doctor\_ID, Cabinet\_ID).
3. **Загрузка:** Все обработанные данные загружаются в четыре эталонных справочника, которые мы определили:
   * Справочник Врачи
   * Справочник Кабинеты
   * История Записей
   * Отчет по доходам врачей (помесячно)

**Этап 2. Прогнозирование спроса:**

* **Прогнозирование Спроса:**
  + **Входные данные:** История Записей за 1-2 года.
  + **Методология:** используются модели анализа временных рядов (напримерProphet). Эти модели способны улавливать сложные зависимости:
    - **Тренд:** Общий рост или падение спроса на услугу со временем.
    - **Сезонность:** Годовая (пики ОРВИ осенью), недельная (пик записей в понедельник, спад в субботу), дневная (пик вечером).
    - **Влияние праздников:** Учет государственных праздников.
  + **Что прогнозируется:** Система строит отдельные прогнозы для:
    - FSS\_x (Спроса на каждую услугу X).
    - FSS\_dms (Спроса на слоты по ДМС по каждой услуге).
  + **Корректировка:** Полученные прогнозы затем корректируются с учетом **внешних факторов**: Сезонный\_Коэф, Акционный\_Коэф и умножаются на Коэф\_Буфера (например, 1.20 (20%)).
  + Формула Финального Прогноза Спроса:
    - Финальный\_Прогноз\_Спрос = ( (Базовый\_Прогноз \* Коэф\_Сезонности) \* Коэф\_Акции ) \* Коэф\_Буфера
    - Расшифровка компонентов и переменных:

|  |  |  |  |
| --- | --- | --- | --- |
| **Компонент** | **Переменная** | **Расшифровка и пояснение** | **Пример** |
| **Базовый Прогноз** | Базовый\_Прогноз | Это "чистый" прогноз, сгенерированный статистической моделью на основе анализа исторических данных. Он уже учитывает общие тренды и регулярные колебания (например, что по понедельникам спрос всегда выше). | Модель прогнозирует, что в ноябре на услугу "Первичный прием терапевта" будет **150** обращений. |
| **Коэффициент Сезонности** | Коэф\_Сезонности | Это вручную заданный бизнесом мультипликатор, который корректирует прогноз с учетом известных, ярко выраженных сезонных всплесков, которые могут быть не до конца учтены автоматической моделью. | Для ноября и специальности "Терапия" установлен коэффициент **1.5** из-за пика ОРВИ. |
| **Коэффициент Акции** | Коэф\_Акции | Это вручную заданный бизнесом мультипликатор, который учитывает влияние запланированных маркетинговых кампаний. | На ноябрь запланирована акция "Check-up для офисных работников", которая, по оценке маркетологов, увеличит спрос на терапевтов на 20%. Коэффициент = **1.2**. |
| **Коэффициент Буфера** | Коэф\_Буфера | Стратегический запас мощности для обработки экстренных пациентов и непредвиденного спроса. | Клиника решила всегда иметь резерв в 20%. Коэффициент = **1.20**. |

* + **Результат:** **"Профиль Целевой Мощности"** — детализированная таблица с финальным прогнозом спроса на планируемый месяц.

**Примечание:** для показателей «**Коэффициент Сезонности»** и **«Коэффициент Акции»** существуют справочники в таблице справочников.

Когда система будет рассчитывать Финальный Прогноз Спроса для конкретной специальности (например, "Терапия") на конкретную дату (например, 10 ноября 2024 г.):

* Она возьмет Базовый\_Прогноз из модели.
* Заглянет в Справочник Сезонных Коэффициентов, найдет правило для 11-го месяца и специальности "Терапия" и умножит прогноз на 1.5.
* Заглянет в Календарь Маркетинговых Акций, увидит, что дата "10.11.2024" попадает в диапазон акции PROMO\_001 для "Терапии", и умножит результат на 1.2.
* Наконец, умножит все на Коэф\_Буфера.

**Этап 3. Прогнозирование Финансовых показателей:**

1. **Прогнозирование Финансовых Показателей и Заполняемости:**
   * **Входные данные:** История Записей и Отчет по доходам врачей.
   * **Методология:** аналогично используются модели временных рядов для каждого врача (или специальности).
   * **Что прогнозируется:**
     + Прогнозируемый\_Прямой\_Доход\_на\_слот (или Средний чек).
     + Прогнозируемый\_Совокупный\_Косвенный\_Доход\_за\_Месяц. (сумма по всем ОУ, в которых врач установлен как направивший)
     + Прогнозируемый\_Коэффициент\_Заполняемости (= (Количество занятых слотов / Общее количество доступных слотов) \* 100%).
   * **Результат:** Набор прогнозных метрик для каждого врача на планируемый месяц.

**Этап 4. Модификация показателей:**

**1. Модификация показателей для новичков**

* Идентификация новичков: Система проверяет каждого врача по условию  Стаж\_Врача < Период\_Адаптации (напр. 6 мес.)
* Применение формулы модификации: Для всех, кто проходит проверку, их прогнозные показатели (доход, заполняемость) временно "улучшаются" с помощью формулы:
  + Коэф\_Коррекции = (Период\_Адаптации - Стаж\_Врача) / Период\_Адаптации
  + Модиф\_Показатель = Реальный\_Показатель + (Средний\_по\_Специальности - Реальный\_Показатель) \* Коэф\_Коррекции  
    *(Применяется к Среднему Чеку, Коэффициенту Заполняемости. Косвенный доход)*

## **3.2. Формулы и этапы фазы генерации случайных расписаний «Фаза 2» (Генерация первого поколения расписаний).**

### Процесс генерации одного случайного, но валидного расписания ("Хромосомы")

**Задача:** заполнить месячную сетку расписания [Дата][Смена][Кабинет] валидными назначениями врачей [Врач, Роль], соблюдая все правила Уровней 0 и 1.

**Входные данные для генератора:**

* **Пустая сетка расписания:** Матрица всех доступных слотов [Дата][Смена][Кабинет] на планируемый месяц.
* **"Матрица Доступности":** предварительно рассчитанный календарь, где отмечены все нерабочие часы для каждого врача и каждого кабинета.
* **"Свод Правил":** Все наши справочники (Врачи, Кабинеты, Жесткие привязки).
* **Фиксированное расписание "звёзд".**

**Алгоритм генерации:**

**Шаг 1: "Замораживание" расписания "звёзд"**

* **Действие:** Генератор в первую очередь проходит по расписанию врачей-"звёзд" и проставляет их смены в пустую сетку. Эти ячейки помечаются как "занятые" и больше не участвуют в случайном распределении.
* **Результат:** Часть сетки уже заполнена детерминировано, создавая "каркас", вокруг которого будет строиться остальное расписание.

**Шаг 2: Создание "Пула доступных назначений"**

* **Действие:** чтобы не перебирать всех врачей для каждой ячейки, генератор создает "пул" всех возможных валидных назначений на каждую смену каждого дня.
* **Как это работает (пример для смены "Понедельник, Утро"):**
  1. Берется список **всех врачей (не-"звёзд")**.
  2. Проверяется **"Матрица Доступности"**: отсеиваются все, кто в этот день/смену не работает (в отпуске, выходной и т.д.).
  3. Оставшиеся врачи "раскладываются" на **роли**.
  4. Берется список **всех кабинетов**, доступных в эту смену (не на ТО, не заняты "звездой").
  5. **Создается список валидных пар**[Кабинет, Врач, Роль]**:** для каждого врача в каждой его роли система проверяет, какие из свободных кабинетов ему подходят согласно **жестким правилам**:
     + Соблюдена ли **"жесткая привязка"**?
     + Соответствует ли Требуемая специализация роли Специализации кабинета?
* **Результат:** на выходе мы получаем для каждой смены список всех теоретически возможных, но еще не сделанных назначений.
  1. *Пример пула для "Пн, Утро":*
     + *[[Каб\_ПЕД, Петров, Педиатрия],*
     + *[Каб\_ЛОР, Сидоров, ЛОР],*
     + *[Каб\_Общ\_1, Сидоров, Терапия],*
     + *[Каб\_Общ\_1, Егорова, Терапия], ...]*

**Шаг 3: Итеративное случайное заполнение сетки**  
Это ядро процесса генерации. Генератор проходит по каждой пустой ячейке [Дата][Смена][Кабинет] и случайным образом назначает туда врача.

* **Действие (для ячейки [Пн, Утро, Каб\_ПЕД]):**
  1. Генератор смотрит в "Пул доступных назначений" для "Пн, Утро" и находит всех кандидатов, которые могут работать в Каб\_ПЕД. Допустим, это только [Петров, Педиатрия]. Выбор очевиден, но он все еще считается "случайным" из одного элемента.
  2. Назначение [Петров, Педиатрия] ставится в ячейку.
  3. **Важнейший шаг:** сразу после этого назначения Доктор Петров помечается как **"занятый"** на смену "Пн, Утро". Он больше не может быть назначен ни в какой другой кабинет в эту же смену.
* **Действие (для ячейки [Пн, Утро, Каб\_Общ\_1]):**
  1. Генератор смотрит в "Пул..." и находит всех **еще не занятых** кандидатов для этого кабинета. Допустим, это [Сидоров, Терапия] и [Егорова, Терапия].
  2. **Происходит случайный выбор:** С вероятностью 50/50 выбирается либо Сидоров, либо Егорова.
  3. Допустим, был выбран [Егорова, Терапия]. Она ставится в ячейку.
  4. Егорова помечается как **"занятая"** на эту смену.
* **Процесс повторяется** для всех пустых ячеек, пока не будут заполнены все возможные слоты или пока не закончатся доступные врачи.

**Шаг 4: Повторение для создания популяции**  
Описанный выше процесс создает **одну** случайную, но валидную "Хромосому". Чтобы создать популяцию из 1000 особей, этот процесс просто **запускается 1000 раз**. Поскольку на каждом шаге, где есть выбор из нескольких кандидатов, этот выбор случаен, каждая из 1000 сгенерированных "Хромосом" будет уникальной.

### Почему этот подход эффективен?

1. **Гарантия валидности:** Мы не генерируем "мусорные" расписания, которые потом придется отбраковывать. Каждая созданная особь с самого рождения соответствует всем жестким правилам и готова к оценке.
2. **Скорость:** Предварительное создание "Пула доступных назначений" значительно ускоряет процесс, так как на каждом шаге мы выбираем не из всех врачей, а только из заранее отфильтрованного, подходящего списка.
3. **Разнообразие:** Случайный выбор на каждом шаге обеспечивает высокое генетическое разнообразие в начальной популяции. Это дает алгоритму широкий "охват" пространства решений и снижает риск застрять в локальном оптимуме на ранних этапах.

## **3.3. Формулы и этапы фазы оценки поколения «Фаза 3» (Оценка первого поколения расписаний).**

**Задача:** вычислить единую числовую оценку (Итоговая\_Оценка) для одного полного варианта месячного расписания, которое мы называем "Хромосомой". Хромосома представляет собой совокупность всех рабочих смен. Каждая отдельная смена (уникальная связка [Дата, Смена, Кабинет] + [Врач, Роль]) является "строительным блоком" или "Геном" этого расписания.

**Входные данные для оценщика:**

1. **Одна "Хромосома":** полностью заполненная и валидная сетка расписания на месяц.
2. **"Профиль Целевой Мощности":** Детализированный прогноз спроса.
3. **"Свод Правил":** Все справочники с данными врачей, кабинетов, услуг, цен и коэффициентов.
4. **Весовые коэффициенты:** Все W\_..., настроенные бизнесом для определения приоритетов.

### Шаг 0: Инициализация полного списка счетчиков

Перед началом оценки система создает набор переменных ("счетчиков") для сбора статистики по анализируемой Хромосоме. Изначально все они равны нулю или являются пустыми словарями/списками.

**Полный список счетчиков:**

* temp\_Direct\_Revenue: Временная переменная для суммирования прямого дохода.
* temp\_Reliability\_Bonus: Временная переменная для суммирования бонуса надежности.
* Total\_DMS\_Slots, Total\_Mobile\_Slots, Total\_Sick\_Leave\_Slots: Счетчики слотов для стратегических бонусов.
* Provided\_Slots\_by\_Specialty = {}: Словарь вида {'Терапия': 120, 'ЛОР': 40, ...} для подсчета созданных слотов по каждой специальности (счетчик созданной мощности по специальности).
* Provided\_DMS\_Slots\_by\_Specialty = {}: Словарь вида {'Терапия': 30, ...} для подсчета ДМС-слотов по специальностям (счетчик созданной мощности для оказания услуг по ДМС в разрезе специальности).
* Provided\_Power\_by\_Service = {}: Словарь вида {'Кольпоскопия': 80, ...} для подсчета мощностей по каждой услуге.
* Binding\_Penalty\_Counter = 0: Счетчик нарушений "мягких" привязок.
* Scheduled\_Doctors\_List = []: Список уникальных врачей, задействованных в расписании (для расчета косвенного дохода).
* Shifts\_per\_Doctor\_by\_Specialty = {}: Словарь вида {'Терапия': {'Егорова': 12, 'Орлов': 10}, ...} для расчета корректировки за дисбаланс (словарь записывает по каждой специальности количество смен по каждому врачу) .

### Шаг 1: Итеративный проход по каждому "Гену"

Система начинает перебирать каждую запланированную смену (каждый "Ген") в Хромосоме. Для каждого гена [Дата, Смена, Кабинет, Врач, Роль] она выполняет следующие микро-расчеты и обновляет счетчики:

1. **Экономический вклад:**
   * temp\_Direct\_Revenue = N\_слотов\_в\_смене \* Модиф\_ПСЧ\_врача
2. **Вклад в надежность:**
   * temp\_Reliability\_Bonus = N\_слотов\_в\_смене \* Модиф\_Коэф\_Заполняемости\_врача

**Примечание:**

* Модиф\_ПСЧ\_врача (Модифицированный Прогнозируемый Средний Чек): это **не реальный**, а **скорректированный** с учетом стажа врача прогноз его среднего чека на планируемый месяц.
* Модиф\_Коэф\_Заполняемости\_врача (Модифицированный Коэффициент Заполняемости): это **не реальный**, а **скорректированный** прогноз вероятности того, что созданные для этого врача слоты будут заняты.

**На каком этапе они вычисляются и где хранятся?**  
Они вычисляются на **Фазе 1: "Подготовка данных и Прогнозирование"**, то есть **ДО** начала работы генетического алгоритма. Результаты этих вычислений хранятся во временной, оперативной структуре данных (например, в словаре Python или временной таблице в памяти), которая существует на время работы алгоритма.Когда на Шаге 1 оценки "Хромосомы" алгоритм доходит до конкретного врача, он просто берет уже готовое, заранее рассчитанное значение из этой временной структуры.

1. **Стратегические бонусы:**
   * Если Врач.Is\_DMS = Да, то Total\_DMS\_Slots = N\_слотов\_в\_смене. (Аналогично для Is\_Mobile и Issues\_Sick\_Leave).

**Примечание:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Полное название** | **Расшифровка** |
| **Is\_DMS** | Is DMS Doctor | Признак врача. Логическое поле (Да/Нет) в Справочнике Врачей. Указывает, работает ли данный врач с пациентами по ДМС. |
| **Total\_DMS\_Slots** | Total DMS Slots Counter | Счетчик. Переменная, в которой алгоритм накапливает суммарное количество слотов (по всем сменам), которые в данном варианте расписания ведут врачи с признаком Is\_DMS = Да. |
| **Is\_Mobile** | Is Mobile Doctor | Признак врача. (Да/Нет) в Справочнике Врачей. Указывает, может ли врач оказывать услуги на выезде. |
| **Issues\_Sick\_Leave** | Issues Sick Leave | Признак врача. (Да/Нет) в Справочнике Врачей. Указывает, имеет ли врач право выписывать больничные листы. |

1. **Подсчет мощностей:**
   * Для каждой Специальности из профиля Врача: Provided\_Slots\_by\_Specialty[Специальность] = сумм. N\_слотов\_в\_смене.
   * Если Врач.Is\_DMS = Да (если врач имеет в справочнике отметку о приеме по ДМС) то: Provided\_DMS\_Slots\_by\_Specialty[Специальность] = сумм. N\_слотов\_в\_смене.
   * Для каждой Услуги из Available\_Nomenclature\_IDs роли: Provided\_Power\_by\_Service[Услуга] = сумм. N\_слотов\_в\_смене.

**Примечание:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Полное название** | **Расшифровка** |
| **Provided\_Slots\_by\_Specialty** | Provided Slots by Specialty Counter | **Счетчик.** Это словарь (ассоциативный массив), который накапливает информацию о том, сколько слотов для пациентов было создано в разрезе каждой медицинской специальности (кроме слотов по ДМС). Он выглядит так: {'Терапия': 30, 'Неврология': 10, ...}. Он используется для расчета корректировки **за неудовлетворенный спрос по специальности**. |
| **Provided\_DMS\_Slots\_by\_Specialty** | Provided DMS Slots by Specialty Counter | **Счетчик.** Это словарь (ассоциативный массив), который накапливает информацию о том, сколько слотов для пациентов по ДМС было создано в разрезе каждой медицинской специальности. Он выглядит так: {'Терапия': 30, 'Неврология': 10, ...}. Он используется для расчета **коррекции за неудовлетворенный спрос именно по ДМС**. |
| **Available\_Nomenclature\_IDs** |  | Это **поле-справочник** в таблице «Справочник Врачи». Оно представляет собой персональный список оказываемой врачом в ОП номенклатуры |
| **Provided\_Power\_by\_Service** |  | Счетчик мощности расписания по каждой услуге. Условно он хранит следующую информацию: {'кукушка': 30 теоретических слотов, 'ФГДС': 10, ...} |

1. **Подсчет для корректировок:**
   * Собираются данные для Binding\_Penalty\_Counter,
   * Врач добавляется в Scheduled\_Doctors\_List.
   * Обновляется Shifts\_per\_Doctor\_by\_Specialty.

### Шаг 2: Финальные расчеты по Хромосоме в целом

После завершения цикла по генам, система использует накопленные в счетчиках данные для вычисления итоговых компонентов главной формулы.

Подробная расшифровка формул и переменных

**1. Выручка\_от\_Услуг**

* Формула: ( Σ (min(FSS\_x, PSS\_x) \* C\_x) ) + ( Σ RI\_i )
* Назначение: оценивает прогнозируемую валовую выручку.

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Расшифровка** | **Откуда берется / Как рассчитывается** |
| FSS\_x | Forecasted Service Demand | **Прогноз спроса на Услугу X.** Берется из **"Профиля Целевой Мощности"**, рассчитанного на Фазе 1. |
| PSS\_x | Provided Service Supply | **Созданная Мощность по Услуге X**. Берется из итогового значения в счетчике Provided\_Power\_by\_Service[Услуга\_X]. |
| C\_x | Cost | **Стоимость Услуги X.** Берется из справочника номенклатуры. |
| RI\_i | Referral Income | **Прогноз косвенного доход**а врача i. Рассчитывается на Фазе 1, берется для каждого уникального врача из Scheduled\_Doctors\_List. |

**2. Бонус\_за\_Надежность**

* **Формула:** Σ (SCS\_s \* AWFR\_s)
* **Назначение:** поощряет использование востребованных врачей.

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Расшифровка** | **Откуда берется / Как рассчитывается** |
| **SCS\_s** | Scheduled Slots for Specialty | **Количество созданных слотов по специальности**s**.** Берется из итогового значения в счетчике Provided\_Slots\_by\_Specialty[Специальность\_s]. |
| **AWFR\_s** | Average Weighted Fill Rate | **Средневзвешенный модиф. коэф. заполняемости по специальности** s. Рассчитывается после цикла по формуле: (Σ (N\_слотов\_i \* Модиф\_Коэф\_i)) / SCS\_s для всех врачей i данной специальности. |
| **Модиф\_Коэф\_i** | Modified Fill Rate | **Модифицированный коэф. заполняемости врача i.** Рассчитывается на Фазе 1. **Базовый (реальный) коэффициент берется из анализа Истории Записей**, где для каждого врача за определенный период считается (Кол-во состоявшихся приемов / Кол-во созданных слотов). |

**3. Сумма\_Стратегических\_Бонусов**

* **Формула:** (W\_dms \* S\_dms) + (W\_mobile \* S\_mobile) + (W\_sickleave \* S\_sl)
* **Назначение:** Поощряет наличие врачей с важными для бизнеса признаками.

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Расшифровка** | **Откуда берется / Как рассчитывается** |
| **S\_dms** | Total DMS Slots | **Суммарное кол-во ДМС-слотов.** Берется из итогового значения счетчика Total\_DMS\_Slots. |
| **S\_mobile** | Total Mobile Slots | **Суммарное кол-во слотов выездных врачей.** Берется из счетчика Total\_Mobile\_Slots. |
| **S\_sl** | Total Sick Leave Slots | **Суммарное кол-во слотов врачей с правом выписки БЛ.** Берется из счетчика Total\_Sick\_Leave\_Slots. |
| **W\_\*\*\*** | Weight | **Вес (значимость) переменной, э**то настраиваемые бизнесом коэффициенты, которые определяют, насколько сильно бонус влияет на конечный результат алгоритма |

**4. Сумма\_Корректировок**

* **Формула:**  (W\_demand\_pen \* (P\_total + P\_dms + P\_service)) + (W\_binding\_pen \* P\_binding) + (W\_balance\_pen \* P\_balance)
* **Назначение:** наказывает за отклонения от идеала.

|  |  |  |  |
| --- | --- | --- | --- |
| **Переменная** | **Расшифровка** | **Формула / Расчет** | **Подробное объяснение** |
| W\_...\_pen | Weight for Penalty | **Вес (важность) для каждого типа корректировок.** Это настраиваемые бизнесом коэффициенты, которые определяют, насколько "больно" алгоритму будет за то или иное нарушение. Например, коррекция за неудовлетворенный спрос (W\_demand\_pen) обычно имеет самый высокий вес. |  |
| P\_total | Penalty for Total Demand | **Корректировка за дефицит общих слотов.**  Σ max(0, Прогноз\_Спроса\_s -Provided\_Slots\_by\_Specialty[s]) | Алгоритм суммирует дефицит слотов по каждой медицинской специальности s. Формула max(0, ...) означает, что коррекция начисляется только при нехватке. Если создано больше слотов, чем нужно, коррекция равна нулю. |
| P\_dms | Penalty for DMS Demand | **Корректировка за дефицит ДМС-слотов.**  Σ max(0, Прогноз\_Спроса\_ДМС\_s - Provided\_DMS\_Slots\_by\_Specialty[s]) | Аналогично P\_total, но сравнивает прогнозируемый спрос на ДМС-приемы с количеством слотов, созданных врачами, которые работают по ДМС, в разрезе каждой специальности. |
| P\_service | Penalty for Service Demand | **Корректировка за дефицит по конкретным услугам.**  Σ max(0, FSS\_x - PSS\_x) | Суммирует дефицит мощностей по **каждой** отдельной номенклатурной услуге x. Гарантирует, что на смене будут врачи с нужными узкими компетенциями. |
| P\_binding | Penalty for Soft Binding Violation | **Корректировка за нарушение "мягкой" привязки**.  Σ (1) за каждый случай. | Это просто счетчик. Алгоритм проходит по всем "генам" и, если находит случай, когда врач работает не в своем предпочитаемом кабинете (согласно справочнику по врачам), добавляет +1 к коррекции. |
| P\_balance | Penalty for Imbalance | **Корректировка за дисбаланс (ротацию).**  Σstdev(Список\_%\_загрузки\_врачей\_по\_специальности\_s) | Для каждой специальности s собирается список из % загрузки, по каждому врачу (напр., [12, 10, 8]). К этому списку применяется математическая функция **stdev (стандартное отклонение)**. stdev вычисляет меру разброса данных относительно их среднего значения. Чем больше разброс, тем выше коррекция. Коррекции по всем специальностям суммируются. |

**Переменные, используемые в формулах корректировок:**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Расшифровка** | **Откуда берется / Как рассчитывается** |
| **Прогноз\_Спроса\_s** | Forecasted Demand for Specialty | **Прогноз спроса на слоты по специальности s.** Берется из **"Профиля Целевой Мощности"**. |
| **Provided\_Slots\_by\_Specialty[s]** | Provided Slots for Specialty | **Создано слотов по специальности s.** Берется из итогового значения в счетчике, который заполняется при проходе по всем генам. |
| **Прогноз\_Спроса\_ДМС\_s** | Forecasted DMS Demand for Specialty | **Прогноз спроса на ДМС-слоты по специальности s.** Берется из **"Профиля Целевой Мощности"**. |
| **Provided\_DMS\_Slots\_by\_Specialty[s]** | Provided DMS Slots for Specialty | **Создано ДМС-слотов по специальности s.** Берется из итогового значения в соответствующем счетчике. |
| **FSS\_x** | Forecasted Service Demand | **Прогноз спроса на Услугу X.** Берется из **"Профиля Целевой Мощности"**. |
| **PSS\_x** | Provided Service Supply | **Созданная Мощность по Услуге X.** Берется из итогового значения в счетчике Provided\_Power\_by\_Service[Услуга\_X]. |
| **Максимально\_возможное\_кол-во\_смен** | Maximum Possible Shifts | **Вычисляемый параметр.** Рассчитывается один раз на Фазе 1 для каждого врача на основе его индивидуального графика работы и табеля отпусков на планируемый месяц. |
| **Список\_%\_загрузки\_врачей\_по\_специальности\_s** | List of Workload Percentages per Doctor | **Список из процентных значений**. Формируется для каждой специальности. Для каждого врача в списке рассчитывается % загрузки по формуле: (Количество\_назначенных\_смен / Максимально\_возможное\_кол-во\_смен\_в\_месяце) \* 100 |

### Шаг 3: Вычисление итоговой оценки расписания на месяц

На этом шаге система просто подставляет все рассчитанные итоговые значения в главную формулу, получая единую, финальную, комплексную оценку для данной "Хромосомы". Эта оценка и определяет ее "приспособленность" и шансы на выживание и размножение в процессе эволюции.

**1. Главная формула: Функция приспособленности**

Итоговая\_Оценка = (W\_econ \* Выручка\_от\_Услуг) + (W\_demand \* Бонус\_Надежности) + (W\_strat \* Сумма\_Стратегических\_Бонусов) - (W\_pen \* Сумма\_Коррекций)

**Пример настройки весовых коэффициентов:**

* W\_econ (Вес экономики) = **1**.
  + **Пояснение:** Мы хотим, чтобы каждый рубль прогнозируемой выручки напрямую учитывался в оценке. Это делает оценку интуитивно понятной в денежном выражении.
* W\_demand (Вес надежности) = **5000**.
  + **Пояснение:** Этот вес значительно выше, чем у экономики. Он означает, что один пункт "надежности" (который примерно соответствует одному гарантированно заполненному слоту) для нас так же важен, как 5000 рублей потенциальной выручки. Это заставляет алгоритм предпочитать более "надежных" врачей, даже если их доход чуть ниже.
* W\_strat (Вес стратегии) = **100**.
  + **Пояснение:** Каждый слот, который ведут врачи с важными бизнес-признаками (ДМС, Выезд), добавляет 100 "бонусных очков". Это относительно небольшой вес, который будет работать как "разрушитель ничьих".
* W\_pen (Вес коррекций) = **1**.
  + **Пояснение:** Этот общий вес для блока коррекций. Важность каждой конкретной коррекции (за спрос, за дисбаланс и т.д.) регулируется **внутри** формулы расчета Суммы\_Коррекций своими собственными весами (W\_demand\_pen, W\_balance\_pen и т.д.), которые, как правило, очень высоки (например, 10,000).

### Шаг 4: Получение оптимального расписания (Эволюция)

**Исходная точка:**  есть "популяция" из 1000 вариантов расписания ("Хромосом"), и для каждого из них на предыдущем шаге была рассчитана **Итоговая Оценка (приспособленность)**.

**Цель:** создать совершенно новую популяцию из 1000 "потомков", которые, в среднем будут "сильнее" чем их родители.

Этот процесс состоит из трех последовательных стадий: **Селекция**, **Скрещивание** и **Мутация**.

**Селекция (Отбор родителей)**

**Задача:** выбрать самые лучшие и перспективные расписания из текущего поколения, чтобы именно они передали свои "гены" дальше.

**Механизм (Турнирный отбор):**

1. **Формирование "Турнира":** из всей популяции в 1000 расписаний алгоритм случайным образом выбирает небольшую группу, например, 5 расписаний.
2. **Определение Победителя:** он сравнивает **Итоговые Оценки** этих 5 расписаний. То расписание, у которого оценка самая высокая, объявляется победителем "турнира".
3. **Попадание в "Пул Родителей":** Победитель копируется в специальный временный список — "пул родителей".
4. **Повторение:** Шаги 1-3 повторяются много раз (например, 500 раз), пока "пул родителей" не заполнится.

**Что это дает?**

* **Выживание сильнейших:** расписания с высокой оценкой будут часто побеждать в турнирах и многократно попадут в "пул родителей". Их "гены" (удачные комбинации смен) получат больше шансов на распространение.
* **Элемент случайности:** даже у расписания со средней оценкой есть небольшой шанс попасть в турнир со слабыми соперниками и победить. Это поддерживает генетическое разнообразие.
* **Вымирание слабых:** расписания с очень низкой оценкой почти никогда не побеждают в турнирах и, скорее всего, не оставят "потомства". Их неудачные "гены" будут утеряны.

**Результат этого шага:**  "пул родителей" — концентрированный набор самых успешных решений из текущего поколения.

**Скрещивание (Создание потомков)**

**Задача:** скомбинировать "гены" двух сильных родителей, чтобы создать нового "потомка", который может унаследовать лучшие черты от обоих.

**Механизм (Одноточечный кроссовер):**

1. **Выбор Пары:** алгоритм случайным образом выбирает из "пула родителей" два расписания: **Родитель А** и **Родитель Б**.
2. **Выбор Точки Разрыва:** алгоритм случайным образом выбирает "точку разреза" в месячном расписании. Например, если месяц длится 30 дней, точка может выпасть после 12-го дня.
3. **"Разрез" и "Склейка":** происходит обмен генетическим материалом.

**Визуализация процесса:**

* + **Родитель А:** [Дни 1-12 от А | \*\*Дни 13-30 от А\*\*]
  + **Родитель Б:** [Дни 1-12 от Б | \*\*Дни 13-30 от Б\*\*]

Создаются два новых расписания-"потомка":

* + **Потомок 1:** [Дни 1-12 от А | \*\*Дни 13-30 от Б\*\*] *(Начало от первого, конец от второго)*
  + **Потомок 2:** [Дни 1-12 от Б | \*\*Дни 13-30 от А\*\*] *(Начало от второго, конец от первого)*

1. **Добавление в Новое Поколение:** Эти два совершенно новых "потомка" добавляются в популяцию следующего поколения.
2. **Повторение:** Шаги скрещивания 1-4 повторяются, пока новое поколение не будет полностью заполнено (1000 особей).

**Что это дает?**

* **Комбинация удачных решений:** если у Родителя А было очень удачное начало месяца, а у Родителя Б — очень удачный конец, есть шанс, что Потомок 1 станет "супер-чемпионом", унаследовав обе сильные стороны.
* **Сохранение "блоков":** Этот метод хорошо сохраняет большие, взаимосвязанные блоки решений (например, удачно составленное расписание на целую неделю), не разрушая их.

**Мутация (Внесение случайных инноваций)**

**Задача:** предотвратить застой, когда все расписания становятся слишком похожими друг на друга, и дать алгоритму шанс найти совершенно новые, не очевидные решения.

**Механизм:**

1. **Итерация:** алгоритм проходит по **каждой смене (каждому "гену")** в **каждой "Хромосоме"** нового поколения.
2. **Случайное событие:** с очень низкой вероятностью (**Mutation Rate**, например, 0.1%) он "мутирует" ген.
3. **Виды мутаций:**
   * **Замена:** В гене [..., Каб\_101, Врач\_А, ...] Врач\_А заменяется на другого **случайного, но валидного** врача, который может работать в этом кабинете в это время.
   * **Обмен (Swap):** Два случайных гена в расписании меняются местами (например, смена Врача А в понедельник меняется со сменой Врача Б в пятницу).
4. **Валидация:** Любая мутация обязательно проверяется на соответствие **жестким ограничениям**. Если мутация создает невалидное расписание, она отменяется.

**Что это дает?**

* **Выход из "локальных оптимумов":** если все лучшие решения "застряли" на хорошем, но не идеальном уровне, одна удачная мутация может подсказать новый, более перспективный путь развития.
* **Поддержание разнообразия:** мутация гарантирует, что в популяции всегда будут присутствовать новые, свежие "гены".

**Замена Поколений**

После того как новое поколение из 1000 "потомков" и "мутантов" полностью сформировано, старая популяция полностью **уничтожается**. Новое поколение занимает ее место и отправляется на **Шаг 3 (Вычисление итоговой оценки расписания на месяц)**.

Этот цикл **[Оценка -> Селекция -> Скрещивание -> Мутация]** и есть сама "Эволюция". Он повторяется сотни или тысячи раз, и с каждым витком средняя "приспособленность" популяции растет, пока алгоритм не найдет решение, которое будет максимально близко к идеалу (итоговая оценка лучшего расписания не будет увеличиваться).

**Как это работает:**

1. **Отслеживание лучшего результата:** после оценки каждого поколения алгоритм сохраняет **Итоговую Оценку** самой лучшей "Хромосомы" в этом поколении.
2. **Счетчик стагнации:** алгоритм ведет счетчик, который показывает, сколько поколений подряд лучшая оценка не улучшалась (или улучшалась незначительно, на величину меньше заданного порога epsilon).
3. **Условие остановки:** если счетчик стагнации достигает заранее определенного значения (например, **50 поколений**), алгоритм делает вывод, что он достиг "плато" — нашел вершину локального (а возможно, и глобального) оптимума, и дальнейшие вычисления, скорее всего, не принесут значимых улучшений.

**Почему это самый популярный критерий:**  
Он адаптивный. Если задача простая, алгоритм быстро найдет хорошее решение и остановится. Если задача сложная, он будет работать дольше, пока действительно не перестанет находить улучшения. Это самый эффективный способ не тратить лишнее время на вычисления.

**Пример:**

* Поколение 100: Лучшая оценка = 1,500,000
* Поколение 101: Лучшая оценка = 1,510,000 (улучшение, счетчик сброшен)
* Поколение 102: Лучшая оценка = 1,512,000 (улучшение, счетчик сброшен)
* Поколение 103: Лучшая оценка = 1,512,000 (улучшения нет, счетчик = 1)
* ...
* Поколение 152: Лучшая оценка = 1,512,000 (улучшения нет, счетчик = 50) -> **СТОП. Эволюция завершена.**

# **4. Пример вычислений**

**Задача:** сформировать оптимальное расписание на **5 дней (Пн-Пт)**, которые мы условно считаем за месяц, для клиники с 5 кабинетами и 5 врачами.

**Входные данные:**

* **Кабинеты (5 шт.):** 1 Гинекологический (Каб\_ГИН), 1 ЛОР (Каб\_ЛОР), 1 Педиатрический (Каб\_ПЕД), 2 Общих консультационных (Каб\_Общ\_1, Каб\_Общ\_2).
* **Смены:** Утро (У) - 10 слотов.
* **Врачи (5 шт.):**
  + **Иванова (Гинеколог, "Звезда"):** работает в Каб\_ГИН по Пн, Ср, Пт (У). Доход 50,000/смена (Максимально возможных смен: 3).
  + **Петров (Педиатр):** Доступен все 5 дней, мягкая привязка к Каб\_ПЕД. Доход 40,000/смена. Коэф. заполняемости 0.9 (Максимально возможных смен: 5).
  + **Сидоров (ЛОР):** Доступен все 5 дней, требует Каб\_ЛОР. Доход 35,000/смена. Коэф. заполняемости 0.8 (Максимально возможных смен: 5).
  + **Егорова (Терапевт):** Доступен все 5 дней, работает по ДМС. Делает услугу "Справка в бассейн" (СБ). Доход 28,000/смена. Коэф. заполняемости 0.85 (Максимально возможных смен: 5).
  + **Орлов (Терапевт, "Новичок"):** Работает Пн, Ср, Пт, стаж 1 мес. Реальный доход 20,000/смена. Реальный коэф. заполняемости 0.5. Работает по ДМС (Максимально возможных смен: 3).
* **Прогноз спроса на 5 дней (уже с учетом буфера +20%):**
  + Гинекология: 4 смены
  + Педиатрия: 5 смен
  + ЛОР: 3 смены
  + Терапия: 7 смен (из них 3 по ДМС)
  + Услуга "Справка в бассейн" (СБ): 2 шт. (стоимость включена в доход смены)
* **Веса для формулы итоговой оценки:**
  + W\_econ (Вес экономики) = 1
  + W\_demand (Вес надежности) = 5000
  + W\_strat (Вес стратегии) = 100
  + W\_pen (Общий вес коррекций) = 1 (внутри него W\_demand\_pen = 10000)

### Шаг 1: Предварительные расчеты (Фаза 1)

* **Модификация показателей Орлова ("Новичок"):**
  + Средний доход терапевта = 28,000. Средний коэф. = 0.85. Период адаптации = 6 мес.
  + Коэф\_Коррекции = (6 - 1) / 6 = 0.833
  + Модиф\_Доход = 20000 + (28000 - 20000) \* 0.833 = 26,664
  + Модиф\_Коэф\_Заполняемости = 0.5 + (0.85 - 0.5) \* 0.833 = 0.79

### Шаг 2: Генерация и оценка двух конкурирующих "Хромосом"

**Хромосома А (Акцент на самых "надежных" врачах)**

* *Логика генератора: чтобы покрыть спрос на Терапию, он выбрал Егорову (коэф. 0.85) и Сидорова (коэф. 0.8), проигнорировав новичка Орлова.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **День** | **Смена** | **Каб\_ГИН** | **Каб\_ПЕД** | **Каб\_ЛОР** | **Каб\_Общ\_1** | **Каб\_Общ\_2** |
| Пн | У | Иванова | Петров | Сидоров | Егорова | (пусто) |
| Вт | У | (пусто) | Петров | Сидоров | Егорова | (пусто) |
| Ср | У | Иванова | Петров | (пусто) | Егорова | (пусто) |
| Чт | У | (пусто) | Петров | Сидоров | Егорова | (пусто) |
| Пт | У | Иванова | Петров | (пусто) | Егорова | (пусто) |

**Хромосома Б (Более сбалансированная)**

* *Логика генератора: он использовал всех доступных терапевтов, включая новичка, для более равномерного распределения.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **День** | **Смена** | **Каб\_ГИН** | **Каб\_ПЕД** | **Каб\_ЛОР** | **Каб\_Общ\_1** | **Каб\_Общ\_2** |
| Пн | У | Иванова | Петров | Сидоров | Егорова | Орлов |
| Вт | У | (пусто) | Петров | Сидоров | Егорова | (пусто) |
| Ср | У | Иванова | Петров | (пусто) | Орлов | (пусто) |
| Чт | У | (пусто) | Петров | Сидоров | Егорова | (пусто) |
| Пт | У | Иванова | Петров | (пусто) | Егорова | Орлов |

### Шаг 3: Расчет оценки для каждой Хромосомы

**РАСЧЕТ ДЛЯ ХРОМОСОМЫ А:**

* **1. Выручка\_от\_Услуг (Прогнозируемая выручка):**
  + Иванова: 3 \* 50000 = 150,000
  + Петров: 5 \* 40000 = 200,000
  + Сидоров: 3 \* 35000 = 105,000
  + Егорова: 5 \* 28000 = 140,000
  + *Итого:* **595,000** (косвенный доход для упрощения считаем включенным)
* **2. Бонус\_за\_Надежность:**
  + Гинекология: 30 слотов \* 1.0 (у "звезды") = 30
  + Педиатрия: 50 слотов \* 0.9 = 45
  + ЛОР: 30 слотов \* 0.8 = 24
  + Терапия: 50 слотов \* 0.85 = 42.5
  + *Итого:* **141.5**
* **3. Сумма\_Стратегических\_Бонусов:**
  + ДМС: Егорова (ДМС) работает 5 смен (50 слотов). *Итого:* **50**
* **4. Сумма\_Коррекций:**
  + **Спрос на слоты:**
    - Педиатрия: создано 5, нужно 5. (ОК)
    - ЛОР: создано 3, нужно 3. (ОК)
    - Терапия: создано 5, нужно 7. **Дефицит = 2**. P\_total = 2.
  + **Спрос ДМС:**
    - Терапия: создано 5 ДМС-смен (Егорова), нужно 3. (ОК)
  + **Спрос на услуги:**
    - "Справка в бассейн": Мощность 5 смен (Егорова), нужно 2. (ОК)
  + **Баланс (Терапевты):**
    - Считаем назначенные смены:
      * Егорова: 5 смен.
      * Орлов: 0 смен.
    - Считаем % загрузки от максимума:
      * % загрузки Егоровой = (5 / 5) \* 100 = 100%.
      * % загрузки Орлова = (0 / 3) \* 100 = 0%.
    - Список для расчета: [100, 0].
    - Считаем stdev: stdev(100, 0) = 70.71.
    - P\_balance = 70.71 Очень высокий дисбаланс!
  + **Итого коррекций:** (10000 \* P\_total) + (1 \* P\_balance) = (10000 \* 2) + (1\*70.71) = 20070,71
* **ИТОГОВАЯ ОЦЕНКА (Хромосома А):**  
  = (1 \* 595000) + (5000 \* 141.5) + (100 \* 50) - (1 \* 20070,71)  
  = 595000 + 707500 + 5000 – 20071,71 = \*\*1 287 428,29\*\*

**РАСЧЕТ ДЛЯ ХРОМОСОМЫ Б:**

* **1. Выручка\_от\_Услуг (используем модиф. доход Орлова):**
  + Иванова: 150,000
  + Петров: 200,000
  + Сидоров: 105,000
  + Егорова: 4 \* 28000 = 112,000
  + Орлов: 3 \* 26664 = 79,992
  + *Итого:* **646,992**
* **2. Бонус\_за\_Надежность (используем модиф. коэф. Орлова):**
  + Гинек., Педиатр., ЛОР: 30 + 45 + 24 = 99
  + Терапия (средневзвеш. (40\*0.85 + 30\*0.79)/70 = 0.82): 70 слотов \* 0.82 = 57.7
  + *Итого:* **156.7**
* **3.**Сумма\_Стратегических\_Бонусов:
  + ДМС: Егорова (4 смены) + Орлов (3 смены) = 7 смен (70 слотов). *Итого:* **70**
* **4. Сумма\_Коррекций:**
  + **Спрос на слоты:** Все спросы покрыты. P\_total = 0.
  + **Спрос ДМС:** Создано 7 ДМС-смен, нужно 3. (ОК)
  + **Спрос на услуги:** "Справка в бассейн" - Мощность 4 смены Егоровой, нужно 2. (ОК)
  + **Баланс (Терапевты):**
    - Считаем назначенные смены:
      * Егорова: 4 смены.
      * Орлов: 3 смены.
    - Считаем % загрузки от максимума:
      * % загрузки Егоровой = (4 / 5) \* 100 = 80%.
      * % загрузки Орлова = (3 / 3) \* 100 = 100%.
    - Список для расчета: [80, 100].
    - Считаем stdev: stdev(80, 100) = 14.14
    - P\_balance = 14.14 хороший баланс!
  + **Итого коррекций:** 0 + (1 \* 14.14) = 14.14 (очень низкий)
* **ИТОГОВАЯ ОЦЕНКА (Хромосома Б):**  
  = (1 \* 646992) + (5000 \* 156.7) + (100 \* 70) - (1 \* 14.14)  
  = 646992 + 783500 + 7000 – 14.14 = \*\*1 437 477,86\*\*

**Вывод по первому поколению**

**Хромосома Б** получает **значительно более высокую оценку** и будет выбрана для следующего поколения.

**Почему?**  
Несмотря на то, что в **Хромосоме А** работали самые "надежные" и опытные врачи, она **провалилась по главному критерию — не смогла удовлетворить прогнозируемый спрос на терапевтов**. Это привело к огромной коррекции, которая нивелировала все ее преимущества.

**Хромосома Б**, хоть и использовала "слабого" новичка, смогла **полностью покрыть весь спектр спроса**, имела **лучший баланс нагрузки** и **больший стратегический бонус** за ДМС. В сумме это сделало ее гораздо более "приспособленным" и выгодным для клиники решением.

Представим, что после фазы генерации и оценки у нас есть три лучших "особи" в популяции:

* **Хромосома А (Акцент на "надежных"):**
  + **Недостатки:** не покрыт спрос на терапию, сильный дисбаланс нагрузки.
  + **Итоговая Оценка:** 1,287,428
* **Хромосома Б (Сбалансированная):**
  + **Преимущества:** полностью покрыт весь спрос, хороший баланс нагрузки, задействован новичок.
  + **Итоговая Оценка:** 1,437,478
* **Хромосома В (Акцент на доходе, игнорирование новичка):**
  + *Представим еще один вариант, где система поставила Сидорова в роли терапевта, чтобы не использовать "слабого" новичка Орлова.*
  + **Недостатки:** Сильный дисбаланс (Орлов - 0 смен).
  + **Итоговая Оценка:** **1,410,000** (Высокий доход, но коррекция за дисбаланс снизила оценку).

**Вывод по Поколению 1:** **Хромосома Б** является безусловным "чемпионом" этого поколения.

### Шаг 4: Эволюция — от Поколения 1 к Поколению 2

**4.1. Селекция (Отбор родителей)**

* **Механизм:** алгоритм проводит "турниры".
* **Результат:** **хромосома Б (1.437к)** и **Хромосома В (1.410к)** будут очень часто побеждать и доминировать в "пуле родителей". Хромосома А (1.287к) будет отсеяна как бесперспективная.
* **Выбранные родители для скрещивания:** **Родитель Б** и **Родитель В**.

**4.2. Скрещивание (Создание потомков)**

* **Механизм:** одноточечный кроссовер. Алгоритм выбирает случайную точку разрыва, например, **после Среды**.

**Визуализация родителей (упрощенно):**

* **Родитель Б (сбалансированный):**  
  [Пн(Б) | Вт(Б) | Ср(Б) | \*\*Чт(Б) | Пт(Б)\*\*]
* **Родитель В (доходный, но несбалансированный):**  
  [Пн(В) | Вт(В) | Ср(В) | \*\*Чт(В) | Пт(В)\*\*]

**Создаются два "потомка":**

* **Потомок Г (Новый Чемпион):** [Пн(Б) | Вт(Б) | Ср(Б) | \*\*Чт(В) | Пт(В)\*\*]
* **Потомок Д (Неудачный гибрид):** [Пн(В) | Вт(В) | Ср(В) | \*\*Чт(Б) | Пт(Б)\*\*]

**4.3. Оценка Нового Поколения (Поколение 2)**

Теперь необходимо рассчитать полную оценку для **Потомка Г**.

* **Анализ Потомка Г:**
  + **Начало (Пн-Ср) от Родителя Б:** Эта часть уже была сбалансированной и включала новичка Орлова.
  + **Конец (Чт-Пт) от Родителя В:** В этой части Родитель В, возможно, использовал более доходные комбинации врачей (например, поставил Егорову вместо Орлова), так как не "заботился" о балансе.
* **Что произошло:** Потомок Г унаследовал **сбалансированное начало**, которое уже дало смены новичку, и **высокодоходный конец**. В результате его **общий баланс нагрузки** может оказаться лучше, чем у Родителя В, а **общий доход** — выше, чем у Родителя Б. Он мог случайно взять лучшее от обоих!

**Примерный пересчет оценки для Потомка Г:**

* **1. Выручка\_от\_Услуг:** оказалась чуть выше, чем у Родителя Б, за счет более доходного конца недели. Итого: \*\*652,000\*\*
* **2. Бонус\_за\_Надежность:** практически не изменился. Итого: \*\*155.0\*\*
* **3. Сумма\_Стратегических\_Бонусов:** практически не изменилась. Итого: \*\*70\*\*
* **4. Сумма\_Коррекций:**
  + **Спрос:** по-прежнему полностью покрыт. Коррекция за спрос = 0.
  + **Баланс:** Егорова: 5 смен, Орлов: 2, Сидоров: 2.
    - Максимально возможные смены для каждого (из исходных данных):
      * Егорова: 5 смен (доступна все 5 дней).
      * Орлов: 3 смены (доступен Пн, Ср, Пт).
      * Сидоров: 5 смен (доступен все 5 дней).
* **Расчет P\_balance** (Корректировка за дисбаланс Терапевтов) для "Потомка Г":
  + Считаем % загрузки от личного максимума для каждого врача:
  + % загрузки Егоровой = (5 назначенных / 5 возможных) \* 100 = 100%.
  + % загрузки Орлова = (2 назначенных / 3 возможных) \* 100 = 66.7%.
  + % загрузки Сидорова = (2 назначенных / 5 возможных) \* 100 = 40%.
* Формируем список из процентов загрузки для расчета:
  + [100, 66.7, 40]
* Считаем stdev (стандартное отклонение) для этого списка:
* Среднее значение = (100 + 66.7 + 40) / 3 = 68.9%.
* stdev(100, 66.7, 40) ≈ 30.
* P\_balance = 30.
* **ИТОГОВАЯ ОЦЕНКА (Потомок Г):**  
  = (1 \* 652000) + (5000 \* 155.0) + (100 \* 70) - (1 \* 1.41)  
  = 652000 + 775000 + 7000 - 30 = \*\*1,433,970\*\*

**Вывод по Поколению 2:** **Потомок Г** стал новым чемпионом! Его оценка (1,433,970) выше, чем у лучшего из родителей (1,432,478). Произошла **успешная эволюция**.

**Шаг 5: Выход на Плато (Поколения 3, 4, 5...)**

Теперь представим, что этот процесс повторяется. В **Поколении 3** родителями, скорее всего, станут **Потомок Г** и **Родитель Б** (так как они теперь два лучших решения).

* **Проблема:** они уже очень похожи друг на друга. Их "генетический материал" почти идентичен.
* **Скрещивание:** их потомки будут получаться очень похожими на них самих, с минимальными отличиями.
* **Мутация:** иногда случайная мутация будет вносить небольшие изменения, но они редко будут приводить к значительному улучшению, так как расписание уже очень хорошо оптимизировано.

**Как это выглядит в цифрах (гипотетически):**

* **Лучшая оценка в Поколении 1:** 1,432,478
* **Лучшая оценка в Поколении 2:** 1,433,970 *(Улучшение есть, счетчик стагнации сброшен)*
* **Лучшая оценка в Поколении 3:** 1,434,150 *(Небольшое улучшение, счетчик стагнации сброшен)*
* **Лучшая оценка в Поколении 4:** 1,434,150 *(Улучшений нет,****счетчик стагнации = 1****)*
* **Лучшая оценка в Поколении 5:** 1,434,150 *(Улучшений нет,****счетчик стагнации = 2****)*
* ...
* **Лучшая оценка в Поколении 53:** 1,434,150 *(Улучшений нет,****счетчик стагнации = 50****)*

**В этот момент срабатывает критерий остановки.**

**Вывод:** Алгоритм "понимает", что нашел решение, максимально близкое к идеалу, потому что **процесс эволюции исчерпал себя**. Он достиг "плато", вершины, с которой дальнейшие шаги не ведут к значительным улучшениям. Система перепробовала тысячи комбинаций и пришла к выводу, что расписание с оценкой **1,434,150** является лучшим, что можно построить в рамках заданных правил и данных. Три последних отличных варианта будут предложены как финальные решения.

# **5. Что требуется для реализации:**

**Технические требования**

* Создать SQL базу в которой будет храниться информация в требуемом для алгоритма виде.
* Придётся связать SQL базу алгоритма с 1С (выгрузка исторических данных), иначе процесс не будет автоматическим.
* Минимальные требования к железу на котором вычисляется расписание: CPU: 4–8 потоков (можно дольше считать, 2–3 ночных прогона), RAM: 8–16 ГБ (популяция и кэши), SSD, Опционально:1 × потребительский GPU ускорит оценку популяции, но не обязателен. (условно можно сделать расчет за несколько дней на ПК аналитика)

**Расчет требуемой мощности**

Исходим из примерного числа кабинетов и врачей по Инсити: **39 кабинетов × 2 смены × 30 дней = 2 340 генов** (ячейки вида [дата, смена, кабинет]); **80 врачей**.

Обозначения:

* **S** — число генов в хромосоме (у нас 2 340).
* **P** — размер популяции (например, 1 200–1 800).
* **G** — число поколений (например, 300–800).
* Цена одной «оценки гена» (обновление счётчиков по врачу/услугам/специальности) — зависит от реализации.

Полная «тяжесть» оптимизации ≈ **P × G × S** оценок генов.

Пример:

* **P = 1 500**, **G = 400**, **S = 2 340** ⇒ всего ≈ **1.404 млрд** оценок генов.
* Если держать эффективность ~**200 000 генов/сек на поток CPU**, то на одном потоке это ~**1.95 ч** чистой оценки.
* С учётом накладных (генерация популяции, селекция, кроссовер/мутация, сортировки, журналы) умножаем ×2–3 ⇒ **~4–6 часов на поток**.
* На **16 потоках** это **~15–25 минут чистой оценки** или **~1–2 часа “всё включено”** за полный прогон; на **32 потоках** — ещё быстрее.
* Даже если взять запас прочности ×3 (неидеальные данные/код) — **вписываемся в «пару дней» многократно**, можно запускать несколько сценариев с разными весами.

Память: если хранить хромосому как int32/uint16 (врач, роль, кабинет) ~**16 байт/ген**, то популяция **P×S×16**: для P=1 500 — примерно **54 МБ**. С буферами/индексами должны уложится **в 2–4 ГБ**. 32–64 ГБ с огромным запасом.