

# **ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ПРОБЛЕМЫ ПЛАНИРОВАНИЯ РАБОТЫ ВРАЧЕЙ**

## **Заявление о публикации**

Это совместная работа Гаутама П. Раджаппы, доктора Джозефа Х. Уилка и доктора Чарльза Нуна. Мы работаем над этой работой для публикации.

## **Аннотация**

Отделения неотложной помощи имеют повторяющиеся 24-часовые циклы нестационарных пуассоновских поступлений и высокий уровень вариации времени обслуживания. Задача состоит в том, чтобы найти график смен, учитывающий эффекты очередей, минимизирующий среднее время ожидания пациента и максимизирующий предпочтения врачей в отношении смен при ограничениях на время начала смены, продолжительность смены и общее количество доступных часов работы врача в день. Предлагается подход, использующий генетический алгоритм и дискретно-событийное моделирование для решения задачи составления расписания работы врачей в больнице. Подход протестирован на реальных данных о расписании работы врачей.

## **1. Введение**

В последние два десятилетия генетические алгоритмы применяются для решения сложных комбинаторных оптимизационных задач реального мира, таких как маршрутизация транспортных средств, последовательность и планирование заданий на одном и нескольких станках, задачи упаковки ранцев и контейнеров, планирование ресурсов и проблемы инвентаризации. Согласно Fukunaga et al. (2002a), проблема планирования персонала является NP-полной. Поэтому метаэвристики, такие как генетические алгоритмы и поиск Табу, являются широко используемыми методологиями для решения подобных задач.

Каждая больница сталкивается с проблемой составления расписания работы персонала с учетом его наличия и предпочтений. Хорошее расписание должно не только снижать затраты на рабочую силу, но и предоставлять больше возможностей и обеспечивать высокую степень удовлетворенности персонала. Кроме того, график работы персонала должен быть составлен таким образом, чтобы время ожидания пациентов было минимальным или значительным. Поэтому целью исследования в данной главе является использование генетического алгоритма для построения расписания смен врачей на основе таких ограничений, как предпочтения врачей, их рабочее время и среднее время ожидания пациентов. Подход тестируется на реальных наборах данных для составления расписания работы врачей.

Остальная часть главы организована следующим образом: Раздел 2 посвящен литературе, связанной с планированием работы персонала, в разделе 3 подробно описана проблема и подход генетического алгоритма, а результаты, выводы и будущие исследования описаны в разделе 4.

## **2. Обзор литературы**

Согласно Фукунаге и др. (2002a), проблема планирования персонала является NP-полной задачей. Следовательно, одним из способов получения выполнимого набора решений за разумный промежуток времени является применение эвристических и метаэвристических методов. Дин (2008a) предложил два генетических алгоритма

(эвристики), которые используют битовую строку и двумерную хромосомную структуру для составления расписания работы персонала. В частности, Дин (2008а) смоделировал штатное расписание в виде двумерной хромосомной структуры, в которой строки и столбцы представляют сотрудников и дни соответственно. Он сравнил эти результаты с результатами, полученными при представлении штатного расписания в виде битовой структуры (хромосомы). Даунсленд (1998а) предложил подход, основанный на поиске Табу и стратегических колебаниях, для составления расписания медсестер в одной из крупных британских больниц. Истон и Мансур (1999) предложили распределенный генетический алгоритм для решения задач, связанных с обобщенным покрытием множества (GSC), детерминированными целевыми программами (DGP) и стохастическими целевыми программами (SGP). Распределенный генетический алгоритм использовал штрафные функции для невыполнимых потомков, а также применял алгоритм локального поиска для повышения производительности. DGP был протестирован на трех различных наборах данных и обеспечил лучшие решения, но за счет экономии вычислительного времени.

Айкелин и Даунсленд (2004) разработали непрямой подход, в котором сначала эвристический декодер строит штатное расписание из различных комбинаций доступных ресурсов. Затем применяется генетический алгоритм для оптимизации выходного расписания, полученного с помощью эвристического декодера. Генетический алгоритм решает только проблему без ограничений, оставляя обработку ограничений эвристическому декодеру, который использует их для непосредственного поиска, а не только в штрафных функциях. Кроме того, все знания о конкретной задаче хранились в эвристическом декодере, что позволяло алгоритму быстро адаптироваться к изменениям в спецификации задачи. Результаты, полученные с помощью этого косвенного подхода, оказались более благоприятными и устойчивыми, чем результаты, полученные с помощью поиска Табу. Таномару (1995b) использовал генетический алгоритм для решения задачи планирования персонала без заранее определенных интервалов смен. Таким образом, вместо предопределенных интервалов смен горизонт планирования был разбит на равномерные временные интервалы, и персонал назначался соответствующим образом. Кроме того, после каждой итерации применялся ряд эвристик для улучшения решения. Результаты были признаны оптимальными для небольших экземпляров и хорошими для больших экземпляров задачи.

Ян и др. (2000b) использовали генетические алгоритмы для составления расписания работы медсестер в больнице, используя концепцию жестких и мягких ограничений. Задача заключалась в минимизации штрафной функции за нарушение мягких ограничений и уменьшении дисперсии в индивидуальном расписании медсестер для обеспечения справедливости расписания. Ян и др. (2000b) также предложили метод, позволяющий лицу, принимающему решения, корректировать расписание и направлять поиск во время его выполнения.

Цай и Ли (2000а) представили генетический алгоритм для решения задачи составления расписания работы медсестер со следующими тремя целями в порядке убывания важности: (1) минимизация общих затрат, (2) минимизация излишков персонала и (3) минимизация дисперсии излишков персонала. Предопределенные недельные графики были назначены, когда было найдено оптимальное количество работников для каждого графика. Затем применялась эвристика для решения нарушенных ограничений. Результаты были хорошего качества и были включены в существующую систему составления расписаний.

Пуэнте и др. (2009b) предложили комбинацию эвристического декодера и генетического алгоритма для составления расписания работы врачей в отделении неотложной помощи. Они использовали концепцию жестких и мягких ограничений, при этом мягким ограничениям присваивались веса в зависимости от их важности. Фактические результаты, полученные с помощью этого эвристического метода, позволили добиться более сбалансированного распределения смен между врачами с высокой степенью удовлетворенности. Ohki et al. (2008b) разработали кооперативный генетический алгоритм (CGA), который использует оператор кроссовера и периодически оператор мутации для решения задачи составления расписания работы медсестер. Они использовали штрафные функции для оценки разницы в части графика смен между исходным графиком, составленным в начале текущего месяца, и графиком, подлежащим новой оптимизации.

Для решения проблемы составления расписания в бельгийской больнице Берк и др. (1998b) разработали коммерческое эвристическое решение под названием Plane, в котором эвристика представляла собой комбинацию поиска Табу и алгоритмов, основанных на ручных методах составления расписания. Plane может решать (для каждой медсестры), какие обязанности можно или нельзя выполнять (в соответствии с квалификационной категорией медсестры) при нехватке персонала, а также обеспечивает объективное расписание, в котором все медсестры получают равные условия, а количество нарушенных ограничений относительно невелико.

Inoue et al. (2003c) предложили интерактивный подход к составлению расписания, в котором функция пригодности основывалась на показателе нарушения мягких ограничений. Однако на каждой итерации генерации решения пользователям предоставлялась возможность изменить расписание с учетом их мнения. Генетический алгоритм использовал комбинации кроссинговера, мутации и эвристики для восстановления кроссинговера (нового поколения). Бруско и Джейкобс (1993) предложили метод имитационного отжига для решения циклической задачи составления расписания работы персонала. Их эвристика обеспечила высокое качество решений за короткое время вычислений на тестовом наборе данных. Они также предположили, что целочисленное программирование с ветвями и границами нецелесообразно для решения циклических задач планирования персонала.

Берк и др. (2009a) предложили алгоритм рассеянного поиска для составления расписания работы медсестер в больнице. В отличие от эвристики, которая работает с одним набором решений, алгоритм рассеянного поиска работает с популяцией решений. Алгоритм рассеянного поиска похож на меметические алгоритмы, за исключением того, что случайные решения заменяются интеллектуально разработанными правилами и решениями, созданными на основе более чем одного родителя. Результаты работы алгоритма рассеянного поиска с методом улучшения hill climbing были признаны более оптимальными при тестировании на эталонных задачах. Burke et al. (2001) использовали меметические алгоритмы для составления расписания работы медсестер и пришли к выводу, что, хотя меметический алгоритм выдает решения высокого качества, он требует большего времени вычислений, чем поиск табу. Özcan (2005) разработал меметический подход для решения проблемы ротации медсестер, где горизонт планирования составлял две недели сменного графика. Özcan (2005) использовал метод подъема на холм для оценки и устранения каждого ограничения, нарушающего график сменности. Чтобы минимизировать общее количество персонала с разным уровнем опыта в соответствии с несколькими трудовыми соглашениями, Бруннер и Эденхартер (2011) сформулировали задачу планирования персонала как смешанную целочисленную линейную программу и

решили ее с помощью эвристики на основе генерации столбцов в отделении анестезии одной больницы.

Dias et al. (2003b) разработали поиск табу и генетический алгоритм для решения проблемы ростеризации в бразильских больницах, при этом мягкие ограничения были взвешены на основе их приоритета и использованы в функции цели. Результаты на тестовом наборе данных показали, что генетический алгоритм немного превосходит поиск Табу, но на практике оба подхода были хорошо приняты персоналом больницы. Для решения задачи составления расписания работы персонала применялось множество других методов исследования операций, таких как генерация графов, программирование с ограничениями, оптимизация по Парето, смешанное целочисленное программирование, гиперэвристика и т. д., обзор которых можно найти в обзорной статье Ernst et al. (2004a).

Paul et al. (2010) представили систематический обзор литературы по моделированию работы отделений неотложной помощи за период с 1970 по 2006 год. Якобсон и др. (2006a) провели обзор различных моделей дискретно-событийного моделирования, актуальных для больниц. Также Jun et al. (1999a) провели обширный обзор по применению дискретно-событийного моделирования в здравоохранении. Кумар и Капур (Kumar and Kapur, 1989a) использовали имитационное моделирование для анализа альтернативных вариантов составления графика работы медсестер в отделении неотложной помощи в больнице Джорджтаунского университета. Розетти и др. (1999c) применили моделирование для тестирования различных вариантов штатного расписания врачей отделения неотложной помощи и анализа влияния расписания на пропускную способность пациентов и использование ресурсов. Венг и др. (2012) предложили двухуровневую схему, названную многоинструментальной интегрированной методологией (MTIM), для составления расписания персонала для каждого отделения неотложной помощи в разных больницах (распределенное решение о распределении ресурсов) в рамках бюджетных ограничений. Gendreau et al. (2007a) предложили четыре различных метода составления расписания, а именно: поиск табу, программирование с ограничениями, математическое программирование и генерацию столбцов для составления расписания работы врачей в отделении неотложной помощи в пяти различных больницах Канады. Yeh и Lin (2007b) предложили сочетание имитационного моделирования и генетических алгоритмов для улучшения качества обслуживания в отделении неотложной помощи. Имитационная модель была использована для анализа потока пациентов в отделении неотложной помощи, а генетический алгоритм - для разработки расписания работы медсестер с целью минимизации времени ожидания пациента. Laskowski et al. (2009c) применили агентные модели и модели очередей для оценки доступа пациентов и потока пациентов в отделении неотложной помощи. Xiao et al. (2010a) предложили методику инкрементного планирования ресурсов на основе временных окон (динамическое планирование), которая использует генетический алгоритм для планирования и изменения расписания ресурсов в выбранные моменты времени (временные окна). Для изучения эффективности методики их подход был интегрирован с существующей системой дискретно-событийного моделирования.

Хотя и не в сфере здравоохранения, Пантел и др. (1998c) применили двухэтапный подход, сочетающий генетический алгоритм и дискретное событийное моделирование, для решения проблем планирования работы цеха в полупроводниковой промышленности. На первом этапе они использовали дискретно-событийную имитационную модель для моделирования поведения динамической системы, а на втором этапе применили генетический алгоритм для минимизации среднего времени пребывания для производства набора партий в зависимости от заказа партии на многоцелевой-многообъектной установке

с неограниченным объемом памяти. Дискретно-событийная имитационная модель была встроена в цикл оптимизации для оценки целевой функции. В нашем подходе к решению проблемы планирования работы врачей в здравоохранении мы также встраиваем наш модуль дискретно-событийного решения в генетический алгоритм, подробности которого описаны в разделе 3.

**3. Постановка задачи и подход генетического алгоритма**

**3.1 Определение проблемы**

В обычном отделении неотложной помощи больницы пациенты поступают в случайное время, и частота поступления зависит от времени суток. Кроме того, услуги врачей носят стохастический характер. Ограничения, такие как предпочтения врачей по времени начала и продолжительности смены, среднее время ожидания пациента и ограничение на общее количество рабочих часов для всех врачей в день, делают задачу очень сложной для решения. Следовательно, эффективный алгоритм составления расписания должен учитывать все эти ограничения реального мира и выдавать результат, удовлетворяющий как врачей, так и пациентов. В данной статье предлагается подход генетического алгоритма для решения задачи составления расписания персонала, который тестируется на двух наборах данных.

**3.1.1 Наборы данных**

Данные для двух наборов данных приведены в таблице 4.1. Для двух наборов данных среднее количество пациентов, поступающих в час, считается пуассоновским, а время обслуживания - экспоненциально распределенным.

**Table 4.1: Given Data**

<b>Given Data</b>	<b>Dataset 1</b>	<b>Dataset2</b>
<b>Average service time</b>	15 minutes (exponential distribution)	33 minutes (exponential distribution)
<b>Average number of patients arriving per hour</b>	Poisson Arrivals (Table 4.2)	Poisson Arrivals (Table 4.3)
<b>Maximum physician hours per day</b>	48	68
<b>Feasible shifts with preference</b>	Table 4.4	Table 4.5

**Table 4.2: Average number of patients arriving per hour (Dataset 1)**

Hour of the day	Average number of patients arriving	Hour of the day	Average number of patients arriving
12:00 AM	3.690616	12:00 PM	8.178273
1:00 AM	2.911858	1:00 PM	7.79489
2:00 AM	2.293054	2:00 PM	7.792522
3:00 AM	2.017725	3:00 PM	8.053659
4:00 AM	1.831175	4:00 PM	7.983501
5:00 AM	1.856022	5:00 PM	7.969416
6:00 AM	2.251625	6:00 PM	8.282366
7:00 AM	3.803911	7:00 PM	7.664413
8:00 AM	5.446445	8:00 PM	7.238266
9:00 AM	7.066014	9:00 PM	6.578026
10:00 AM	7.939452	10:00 PM	5.526836
11:00 AM	8.49382	11:00 PM	4.336112

**Table 4.3: Average number of patients arriving per hour (Dataset 2)**

Hour of the day	Average number of patients arriving	Hour of the day	Average number of patients arriving
12:00 AM	2.621795	12:00 PM	7.083333
1:00 AM	1.916667	1:00 PM	6.826923
2:00 AM	1.448718	2:00 PM	6.557692
3:00 AM	1.294872	3:00 PM	6.570513
4:00 AM	1.403846	4:00 PM	6.076923
5:00 AM	1.378205	5:00 PM	6.512821
6:00 AM	1.839744	6:00 PM	6.730769
7:00 AM	2.858974	7:00 PM	6.750000
8:00 AM	4.288462	8:00 PM	6.064103
9:00 AM	5.769231	9:00 PM	5.384615
10:00 AM	6.769231	10:00 PM	4.339744
11:00 AM	7.038462	11:00 PM	3.147436

**Table 4.4: Feasible shifts with preference (Dataset 1)**

Hour of the day	Shift duration (hours)		
	8	10	12
7:00 AM	6	5	3
11:00 AM	6	3	4
3:00 PM	6	3	1
7:00 PM	4	4	2
11:00 PM	2	3	3

**Table 4.5: Feasible shifts with preference (Dataset 2)**

Hour of the day	Shift durations (hours)				
	8	9	10	11	12
7:00 AM	4	6	6	3	2
8:00 AM	4	6	6	3	2
9:00 AM	4	6	6	4	2
10:00 AM	4	6	6	4	2
11:00 AM	4	6	6	4	2
12:00 PM	5	5	5	3	2
1:00 PM	5	6	5	4	2
2:00 PM	5	6	5	4	2
3:00 PM	5	5	5	4	2
4:00 PM	6	5	5	3	2
5:00 PM	6	5	3	3	2
6:00 PM	3				
9:00 PM	2	3	3	3	
10:00 PM	3	6	6	4	2
11:00 PM	3	5	5	3	2

Из таблицы 4.4 для набора данных 1 время начала смены - 7 утра, 11 утра, 3 часа дня, 7 вечера и 11 вечера. Все смены должны начинаться только в это время. Продолжительность смены для каждого времени начала смены составляет 8, 10 или 12 часов. Предпочтения для каждой комбинации времени начала смены и ее продолжительности приведены в таблице 4.4. Предпочтения пронумерованы от 1 до 6, причем 6 - наиболее предпочтительное время начала и продолжительность смены, а 1 - наименее предпочтительное. Интерпретация возможных смен с предпочтениями для набора данных 2 (табл. 4.5) аналогична набору данных 1.

### 3.1.2 Целевые функции и ограничения

Исходя из полученных данных, ниже описаны цели и ограничения для двух наборов данных.

#### Цели.

Исходя из заданных данных, задача состоит в том, чтобы построить график смен, который:

- 1) Максимизировать предпочтения врачей.
- 2) Минимизировать среднее время ожидания для пациентов.

### Ограничения

- 1) Не допускается сверхурочная работа, т.е. график смен не должен превышать максимальное количество часов работы врача в день.
- 2) Каждый час должен быть доступен хотя бы один врач.
- 3) Смены могут начинаться только в то время, которое указано в матрице предпочтений (табл. 4.4 и табл. 4.5).

Поскольку это многоцелевая оптимизационная задача, каждой цели присваиваются веса (штрафы), а для расчета значения объективной функции используется взвешенная сумма. Noon et al. (2007) предложили математическую формулировку для данной задачи, которая была модифицирована в соответствии с нашей постановкой задачи. Математическая формулировка задачи описана в разделе 3.1.3

### **3.1.3 Математическая формулировка**

#### Индексированные множества:

*$i$  = time period from 1 ...  $T$ , where  $T$  is 24 hours*  
 *$j$  = shift index from 1 ...  $J$ , where  $J$  is the total number of potential shifts*  
 *$T_t$  = Total simulation run time*  
 *$i, j$  are integers*

$i$  - период времени от 1... $T$ , где  $T$  - 24 часа

$j$  - индекс смены от 1... $J$ , где  $J$  - общее количество потенциальных смен

$T_t$  - общее время работы симулятора

$i, j$  - целые числа

#### Параметры:

*$\mu$  = service rate (constant, exponential distribution)*  
 *$\lambda_i$  = arrival rate for time period  $i$*   
 *$d_j$  = duration of each shift  $j$*   
 *$P_j$  = Preference of shift  $j$*

*$H$  = maximum available server(physician) hours*  
 *$\omega_1$  = weight associated with average patient wait time*  
 *$\omega_2$  = weight associated with shift preference penalty*  
 *$b_{ij} = \begin{cases} 1, & \text{if shift } j \text{ is assigned to time } i \\ 0, & \text{otherwise} \end{cases}$*

$\mu$  - скорость обслуживания (постоянная, экспоненциальное распределение)

$\lambda_i$  – коэффициент прибытия для периода времени  $i$

$d_j$  – продолжительность каждой смены  $j$

$P_j$  – предпочтение смены  $j$

$H$  - максимальное количество часов работы сервера (врача)



$w_1$  – вес, связанный со средним временем ожидания пациента

$w_2$  – вес, связанный со штрафом за предпочтение сдвига ??? WHAT?

$b_{ij}$  – 1 - если смена  $j$  назначена на время  $i$ , 0 – иначе

Переменные решения:

$s_i$  = number of servers (physicians) available at time period  $i$

$x_j$  = number of servers (physicians) in shift  $j$

$s_i$  - количество серверов (врачей), доступных в период времени  $i$

$x_j$  - количество серверов (врачей) в смене  $j$

**Учетные переменные (рассчитываются на основе переменных решений и дискретно-событийного моделирования):**

$w_i$  = total average patient wait time for time period  $i$

$w_i$  - общее среднее время ожидания пациента за период времени  $i$

**Цель:**

$$Z = \text{Min}(\omega_1 \sum_i^T w_i + \omega_2 \sum_j^J (P_j * x_j)) \quad (4.1)$$

**Ограничения:**

1) Общее количество часов работы врача  $\leq H$  (максимальное количество часов работы врача в день)

$$\sum_j^J (d_j * x_j) \leq H \quad (4.2)$$

2) Количество врачей в каждой смене должно быть равно количеству врачей в каждый час

$$\sum_j^J (b_{ij}) * (x_j) = s_i \quad \forall i \quad (4.3)$$

3) Не менее одного врача каждый час

$$s_i \geq 1, \text{ integer } \forall i \quad (4.4)$$

4) Количество врачей в данной смене

$$x_j \geq 0, \text{ integer } \forall j \quad (4.5)$$

**Решите:**

$w_i = f([\lambda_i, \mu, s_i])$ : (Функция среднего времени ожидания пациента из дискретно-событийного моделирования). Модуль дискретного моделирования событий является неотъемлемой частью предложенного генетического алгоритма для оценки среднего времени ожидания пациента для каждого выполнимого графика смен.

### 3.2 Генетический алгоритм

Генетические алгоритмы - это основанные на популяции алгоритмы поиска для решения комбинаторных задач оптимизации. Впервые они были предложены Джоном Холландом (1989). В этих алгоритмах пространство поиска (популяция) задачи представляется как набор особей (хромосом), и эти особи оцениваются на основе функции приспособленности. Генетические алгоритмы генерируют решения для задач оптимизации на основе теории эволюции, используя такие понятия, как размножение, кроссинговер и мутация. Фундаментальная концепция генетического алгоритма определяет набор условий для достижения глобального оптимума. Эти условия описывают процесс воспроизводства и гарантируют, что лучшие решения останутся в будущих поколениях, а слабые решения будут исключены из будущих поколений. Это похоже на дарвиновскую концепцию выживания сильнейших в теории эволюции. Типичный генетический алгоритм состоит из следующих шагов (1989):

Шаг 1: Создайте начальную популяцию из N решений.

Шаг 2: оцените каждое решение начальной популяции с помощью фитнес-функции/целевой функции.

Шаг 3: Выберите решения в качестве родителей для нового поколения на основе вероятности или случайности. Лучшие решения (с точки зрения пригодности или цели) имеют большую вероятность быть выбранными, чем плохие решения.

Шаг 4: Используйте родительские решения из шага 3 для создания следующего поколения (так называемого потомства). Этот процесс называется кроссинговером. Потомство помещается в начальный набор решений, заменяя собой более слабые решения.

Шаг 5: случайное изменение нового поколения путем мутации. Обычно это делается с помощью вероятности мутации.

Шаг 6: Повторяйте шаги со 2 по 5 до тех пор, пока не будет достигнут критерий остановки.

Из-за ограничений этой задачи невозможно напрямую использовать генетический алгоритм в том виде, в котором он описан выше. В частности, после кроссинговера могут остаться решения, не удовлетворяющие ограничениям. Следовательно, чтобы получить выполнимый набор потомков, нам может потребоваться изменить способ скрещивания, либо удалить невыполнимые решения после мутации и заменить их решениями с более высоким значением пригодности в старой популяции (2002b) или заполнить новую популяцию с помощью эвристики расписания. В нашем подходе, если существует невыполнимое решение для будущего поколения, мы случайным образом выбираем новые графики смен из начальной популяции. Подход генетического алгоритма для набора данных 1 объясняется ниже.

#### Кодирование решения

В наборе данных 1 система очередей стабильна (рассчитана на основе приведенных данных), и в день доступно максимум 48 часов работы врача. Таким образом, мы просто принимаем решения о сменах, генерируя случайное расписание смен и оценивая его фитнес-функцию. Случайно сгенерированные графики смен определяют, сколько серверов работает в каждый час. Функция пригодности определит, насколько хорошо мощности справляются со спросом и не образуются ли большие очереди. У нас есть три

продолжительности смены - 8, 10 или 12 часов. Следовательно, максимальное количество необходимых смен будет равно простому количеству доступных часов работы врача (48 часов), деленному на наименьшую продолжительность смены (т. е. 8 часов). Таким образом, нам требуется максимум 6 смен.

Поскольку у нас 15 предпочтений, каждый индекс предпочтения в матрице предпочтений (табл. 4.6) пронумерован от 0 до 14 в строке. Например, индекс 0 - это смена в 7 утра с продолжительностью смены 8 часов, а индекс 14 - смена в 11 вечера с продолжительностью 12 часов. Графику «нет» присваивается номер 15.

**Table 4.6: Shift index (Shift preference) matrix (Dataset 1)**

Shift index (preferences)	Shift duration (hours)		
	8	10	12
Hour of the day	8	10	12
7:00 AM	0(6)	1(5)	2(3)
11:00 AM	3(6)	4(3)	5(4)
3:00 PM	6(6)	7(3)	8(1)
7:00 PM	9(4)	10(4)	11(2)
11:00 PM	12(2)	13(3)	14(3)
No schedule	15		

### Шаг 1: Начальная популяция

Для начальной популяции я сначала случайным образом генерирую 2000 графиков смен размером 6 (максимальное количество смен). Например, одно расписание смен может быть [0,3,4,5,14,15], а другая последовательность смен может быть [4,15,4,9,10,11]. Затем каждый из 2000 случайно сгенерированных графиков смен проверяется на предмет наличия хотя бы одного врача в каждый час и отсутствия сверхурочных в графике смен (максимум 48 часов работы врача в день). Если в случайно сгенерированном графике смен есть хотя бы один врач каждый час и нет сверхурочных, этот график смен добавляется в начальную популяцию. Этот процесс продолжается до тех пор, пока не будет сгенерировано заранее определенное количество начальной популяции, которое в нашем случае равно 500.

### Шаг 2: Оценка фитнес-функции

Он включает в себя два этапа, как показано ниже:

#### 1) Валидность (достоверность) последовательности смен:

Это делается для того, чтобы проверить, есть ли в наличии хотя бы один врач каждый час и нет ли сверхурочных в графике смен (максимум 48 часов работы врача в день). Если в случайно сгенерированном графике смен есть хотя бы один врач каждый час и нет сверхурочных, этот график смен добавляется в популяцию.

#### 2) Оценка фитнес-функции:

Для каждого графика смен в популяции вычисляется его фитнес-функция на основе двух целей: 1) максимизация предпочтений врача и 2) минимизация среднего времени ожидания пациента.

- Максимизация предпочтений врача: Накладывается штраф в размере (6 - предпочтение конкретной смены). Например, для 8-часовой смены в 7 утра штраф составляет  $6 - 6 = 0$ .

- Минимизация среднего времени ожидания пациента: Дискретная событийная симуляция продолжительностью 2400 часов (100 дней \* 24 часа в сутки) проводится для каждого из графиков смен на основе скорости прибытия пациентов и наличия врачей в час.

- Затем выпуклая комбинация весов (штрафов) для каждой из для каждой из двух вышеуказанных целей используется для оценки функции пригодности.

### **Шаг 3: Выбор (Селекция)**

Случайно сгенерированные графики смен сортируются в соответствии с наименьшим значением пригодности. Затем 100 лучших сменных графиков отбираются для дальнейшей генерации.

### **Шаг 4: Кроссинговер (Скрещивание)**

- Элитизм: 5 лучших графиков сдвига, полученных на этапе отбора, всегда добавляются в будущее поколение.

- Выбор родителя: Из пула графиков смен выбираются 2 родителя случайным образом выбираются 2 родителя и генерируются 2 дочерних расписания смен с помощью одноточечного кроссинговера для нового поколения.

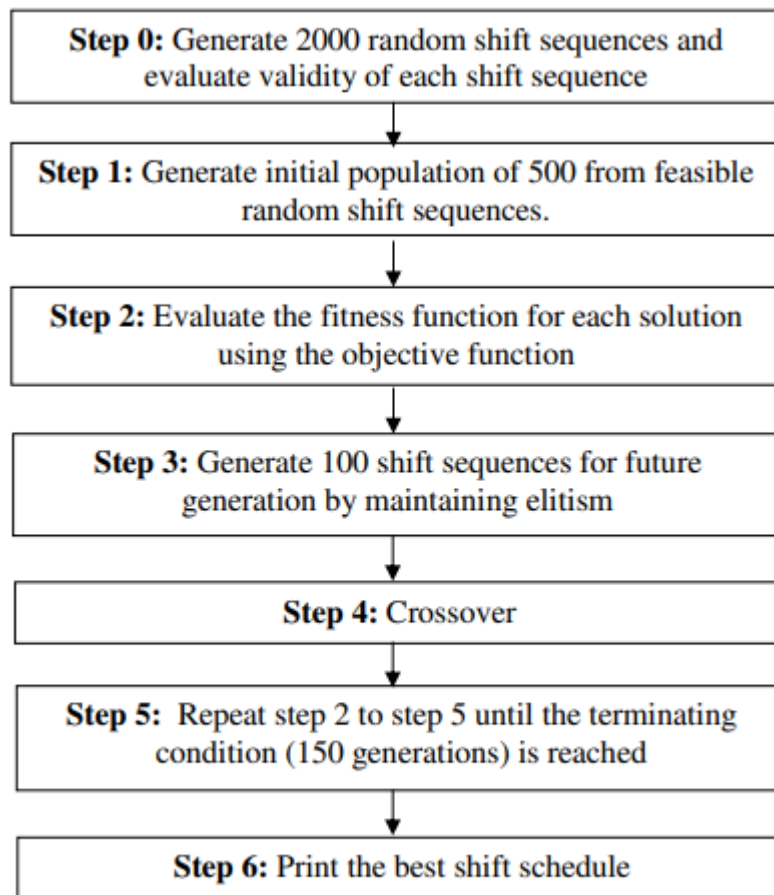
- Вероятность кроссинговера установлена на 1.

- Мутация не производится.

- Если не удастся найти выполнимые расписания, я случайным образом добавляю выполнимые расписания в новое поколение, пока не будет достигнут размер популяции 100.

### **Шаг 5: Условие прекращения**

Затем шаги 2, 3 и 4 повторяются в течение фиксированного числа поколений (условие завершения), которое в нашей задаче равно 150. Схема генетического алгоритма для набора данных 1 показана на рисунке 4.1 ниже:



**Figure 4.1: Genetic Algorithm Flowchart (Dataset 1)**

Для набора данных 2 система очередей стабильна (рассчитано по приведенным данным), и в день доступно максимум 68 часов работы врача. Следовательно, максимальное количество необходимых смен будет равно простому количеству доступных часов работы врача (68 часов), деленному на наименьшую продолжительность смены (т. е. 8 часов). Следовательно, нам требуется максимум 9 смен. Поскольку у нас 70 предпочтений, каждый индекс предпочтения в матрице предпочтений (табл. 4.7) пронумерован от 0 до 69 в строке. Ни одному графику не присвоен индекс 70. Помимо кодирования решения, подход генетического алгоритма к набору данных 2 аналогичен подходу к набору данных 1. Из-за большего размера задачи параметры генетического алгоритма, такие как размер популяции, количество поколений и т. д., были увеличены в 3 раза для набора 2 по сравнению с набором 1. Также следует отметить, что все параметры генетического алгоритма, такие как размер популяции, условие завершения и т. д., были заданы в ходе пилотного тестирования.

**Table 4.7: Shift index (Shift preference) matrix (Dataset 2)**

Shift index(preference)	Shift duration (hours)				
Hour of the day	8	9	10	11	12
7:00 AM	0(4)	1(6)	2(6)	3(3)	4(2)
8:00 AM	5(4)	6(6)	7(6)	8(3)	9(2)
9:00 AM	10(4)	11(6)	12(6)	13(4)	14(2)
10:00 AM	15(4)	16(6)	17(6)	18(4)	19(2)
11:00 AM	20(4)	21(6)	22(6)	23(4)	24(2)
12:00 PM	25(5)	26(5)	27(5)	28(3)	29(2)
1:00 PM	30(5)	31(6)	32(5)	33(4)	34(2)
2:00 PM	35(5)	36(6)	37(5)	38(4)	39(2)
3:00 PM	40(5)	41(5)	42(5)	43(4)	44(2)
4:00 PM	45(6)	46(5)	47(5)	48(3)	49(2)
5:00 PM	50(6)	51(5)	52(3)	53(3)	54(2)
6:00 PM	55(3)				
9:00 PM	56(2)	57(3)	58(3)	59(3)	
10:00 PM	60(3)	61(6)	62(6)	63(4)	64(2)
11:00 PM	65(3)	66(5)	67(5)	68(3)	69(2)
No schedule	70				

## 4. Results, Conclusions, and Future Work

### 4.1 Результаты

Генетический алгоритм для данного исследования был разработан на языке Java на компьютере с операционной системой Windows7, Intel i5 2.4 Ghz, 4 GB RAM. Модуль дискретно-событийного моделирования для оценки среднего времени ожидания пациента также был написан на Java и интегрирован с генетическим алгоритмом для составления расписания смен. Алгоритм был запущен для выпуклой комбинации весов для целевых функций. В силу своей простоты для расчета объективной функции был использован подход взвешенной суммы (Abdullah et al., 2006). Результаты для выпуклой комбинации весов в диапазоне от 0 до 1 для наборов данных 1 и 2 показаны в табл. 4.8 и табл. 4.9 соответственно.

**Table 4.8: Weighted sum approach results (Dataset 1)**

Case #	Preference Weight	Average patient wait time Weight	GA Time(sec)	Total Preference Violation	Average patient wait time(min)	Total physician hours	Shift Schedule
1	1	0	340.964	3	37.94	44	[0, 6, 14, 15, 0, 6]
2	0.9	0.1	116.315	3	20.89	44	[14, 0, 3, 6, 15, 6]
3	0.8	0.2	116.923	3	20.89	44	[6, 15, 14, 6, 3, 0]
4	0.7	0.3	118.778	4	15.31	48	[6, 0, 6, 0, 12, 3]
5	0.6	0.4	118.633	4	15.31	48	[12, 3, 6, 0, 0, 6]
6	0.5	0.5	118.827	4	15.31	48	[3, 12, 6, 0, 0, 6]
7	0.4	0.6	119.131	5	15.31	48	[14, 0, 5, 15, 3, 6]
8	0.3	0.7	118.623	4	15.31	48	[3, 6, 12, 6, 0, 0]
9	0.2	0.8	119.079	8	13.88	48	[11, 0, 1, 15, 4, 6]
10	0.1	0.9	119.995	8	13.88	48	[1, 15, 11, 0, 4, 6]
11	0	1	119.41	8	13.88	48	[6, 1, 11, 15, 4, 0]

**Table 4.9: Weighted sum approach results (Dataset 2)**

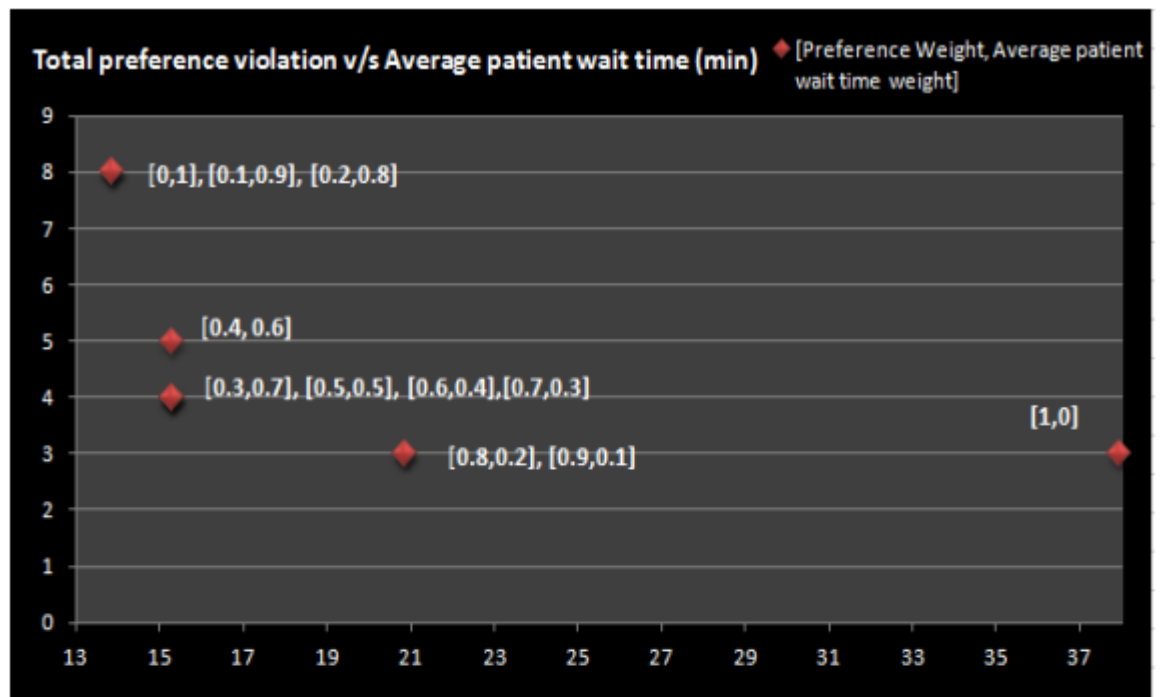
Case #	Preference Weight	Average patient wait time Weight	GA Time(sec)	Total Preference Violation	Average patient wait time (min)	Total Physician hours	Shift Schedule
1	1	0	1170.017	0	35.95	64	[21, 1, 70, 36, 45, 1, 22, 62, 70]
2	0.9	0.1	1101.017	0	35.68	65	[17, 2, 7, 45, 36, 70, 70, 61, 11]
3	0.8	0.2	1145.259	2	35.46	64	[45, 12, 70, 7, 30, 6, 22, 66, 70]
4	0.7	0.3	1092.167	2	34.39	66	[21, 2, 32, 70, 11, 61, 12, 70, 51]
5	0.6	0.4	1122.037	3	31.21	66	[1, 12, 70, 7, 30, 17, 42, 70, 66]
6	0.5	0.5	899.491	3	30.94	67	[2, 62, 1, 70, 70, 21, 32, 50, 13]
7	0.4	0.6	917.144	6	30.27	67	[21, 11, 70, 33, 36, 70, 68, 1, 26]
8	0.3	0.7	910.54	12	27.28	68	[69, 30, 2, 16, 70, 70, 24, 21, 55]
9	0.2	0.8	926.297	10	26.88	68	[68, 31, 27, 70, 46, 3, 0, 12, 70]
10	0.1	0.9	907.925	13	26.38	67	[45, 10, 19, 70, 70, 8, 27, 5, 67]
11	0	1	885.737	16	25.58	67	[5, 70, 16, 19, 68, 70, 23, 55, 5]

Как видно из таблиц 4.8 и 4.9 выше, для наборов данных 1 и 2 нулевой веса для объективной функции среднего времени ожидания пациента приводит к среднему времени ожидания пациента 37,94 минуты и 35,95 минуты соответственно, а когда предпочтениям врачей не присваивается вес, среднее время ожидания пациента составляет 13,88 минуты и 25,58 минуты соответственно. Кроме того, по мере уменьшения веса предпочтений от 1 до 0 и увеличения веса среднего времени ожидания пациента от 0 до 1, общее нарушение предпочтений увеличивается, а среднее время ожидания пациента уменьшается для двух наборов данных. Время вычислений генетического алгоритма показано в четвертом столбце (GA Time (sec)). Как видно из таблицы 4.8, для набора данных 1 в первых трех случаях, когда предпочтение врача имеет больший вес, общее количество использованных часов врача составляет всего 44 часа по сравнению с максимальными 48 часами, доступными каждый день. В то время как для набора данных 2 в таблице 4.9 есть только два случая (случай № 8 и случай № 9), когда максимальное количество доступных часов работы врача, составляющее 68 часов, было полностью использовано.

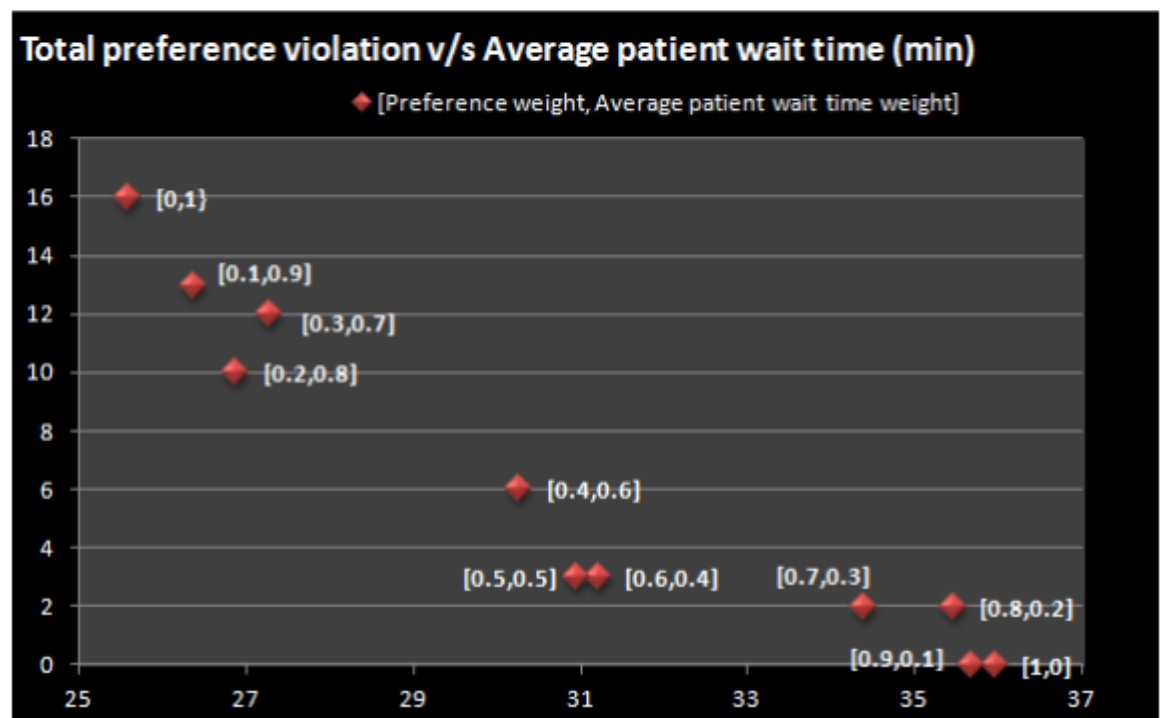
Графики сдвигов для каждой выпуклой комбинации весов показаны в последнем столбце табл. 4.8 и табл. 4.9. Например, для случая №2 в наборе данных 1 наилучшим графиком сдвига является [14,0,3,6,15,6]. Используя таблицу 4.6, график смен выглядит следующим образом:

- 14 Начало смены в 11 вечера на 12 часов
- 0 Начало смены в 7 утра на 8 часов
- 3 Начало смены в 11 утра на 8 часов
- 6 Начало смены в 15:00 на 8 часов
- 15 Нет графика
- 6 Начало смены в 15:00 на 8 часов

Аналогичная интерпретация может быть сделана для всех случаев в двух наборах данных. График зависимости общего нарушения предпочтений от среднего времени ожидания пациента для всех выпуклых комбинаций веса для набора данных 1 и набора данных 2 показан на рисунке 4.2 и рисунке 4.3 соответственно.



**Figure 4.2: Total preference violation v/s Average patient wait time (min)(Dataset 1)**



**Figure 4.3: Total preference violation v/s Average patient wait time (min)(Dataset 2)**

Для примеров № 2, № 6 и № 11 из набора данных 1 количество врачей, доступных в час, и «количество пациентов, находящихся в состоянии» показаны в таблице 4.10, а график, показывающий, как график сменности обрабатывает поступления пациентов в час, представлен на рисунке 4.4(А), рисунке 4.4(В) и рисунке 4.4(С) соответственно. Число пациентов, способных обслужить пациентов» показывает количество пациентов, которые могут быть обслужены врачами каждый час для каждого графика сменности.



Table 4.10: Number of patients of capacity (Dataset 1)

	Case # 2			Case # 6		Case # 11	
Hour of the day	Average number of patient arrivals/hr	Available physicians/hr	Number of patients of capacity	Available physicians/hr	Number of patients of capacity	Available physicians/hr	Number of patients of capacity
12:00 AM	3.690616	1	4	1	4	1	4
1:00 AM	2.911858	1	4	1	4	1	4
2:00 AM	2.293054	1	4	1	4	1	4
3:00 AM	2.017725	1	4	1	4	1	4
4:00 AM	1.831175	1	4	1	4	1	4
5:00 AM	1.856022	1	4	1	4	1	4
6:00 AM	2.251625	1	4	1	4	1	4
7:00 AM	3.803911	2	8	2	8	2	8
8:00 AM	5.446445	2	8	2	8	2	8
9:00 AM	7.066014	2	8	2	8	2	8
10:00 AM	7.939452	2	8	2	8	2	8
11:00 AM	8.493820	2	8	3	12	3	12
12:00 PM	8.178273	2	8	3	12	3	12
1:00 PM	7.794890	2	8	3	12	3	12
2:00 PM	7.792522	2	8	3	12	3	12
3:00 PM	8.053659	3	12	3	12	3	12
4:00 PM	7.983501	3	12	3	12	3	12
5:00 PM	7.969416	3	12	3	12	2	8
6:00 PM	8.282366	3	12	3	12	2	8
7:00 PM	7.664413	2	8	2	8	3	12
8:00 PM	7.238266	2	8	2	8	3	12
9:00 PM	6.578026	2	8	2	8	2	8
10:00 PM	5.526836	2	8	2	8	2	8
11:00 PM	4.336112	1	4	1	4	1	4

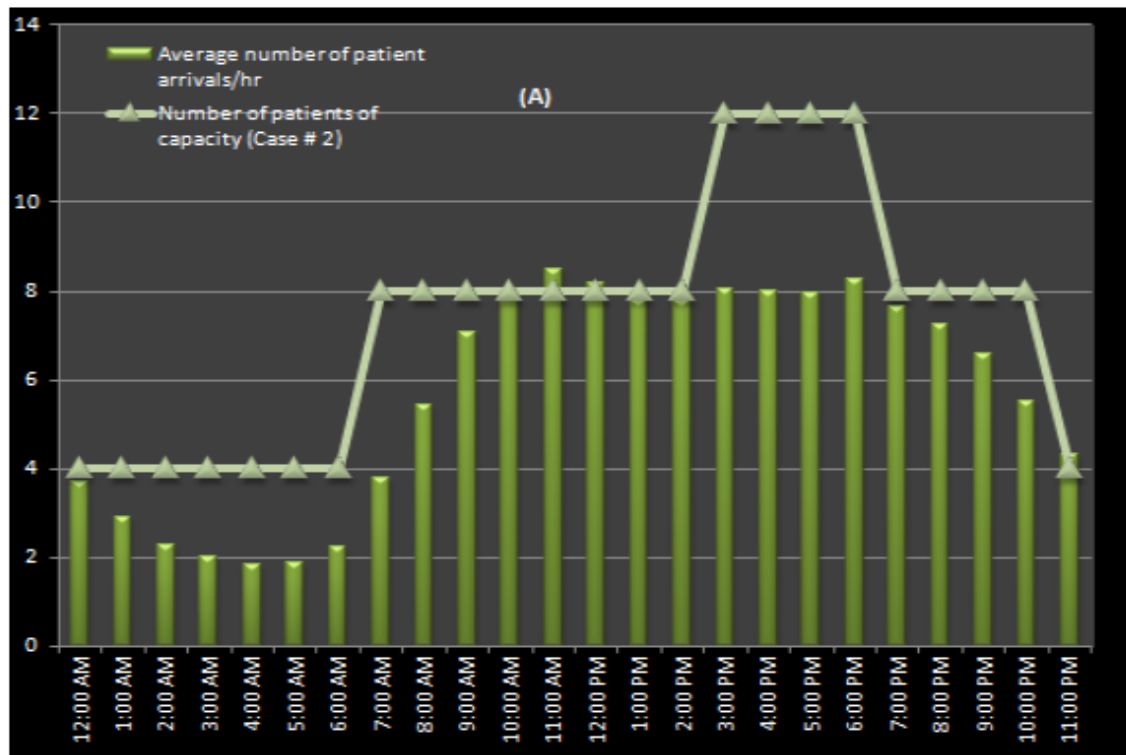


Figure 4.4(A): Number of patients of capacity plot (Case # 2, Dataset 1)

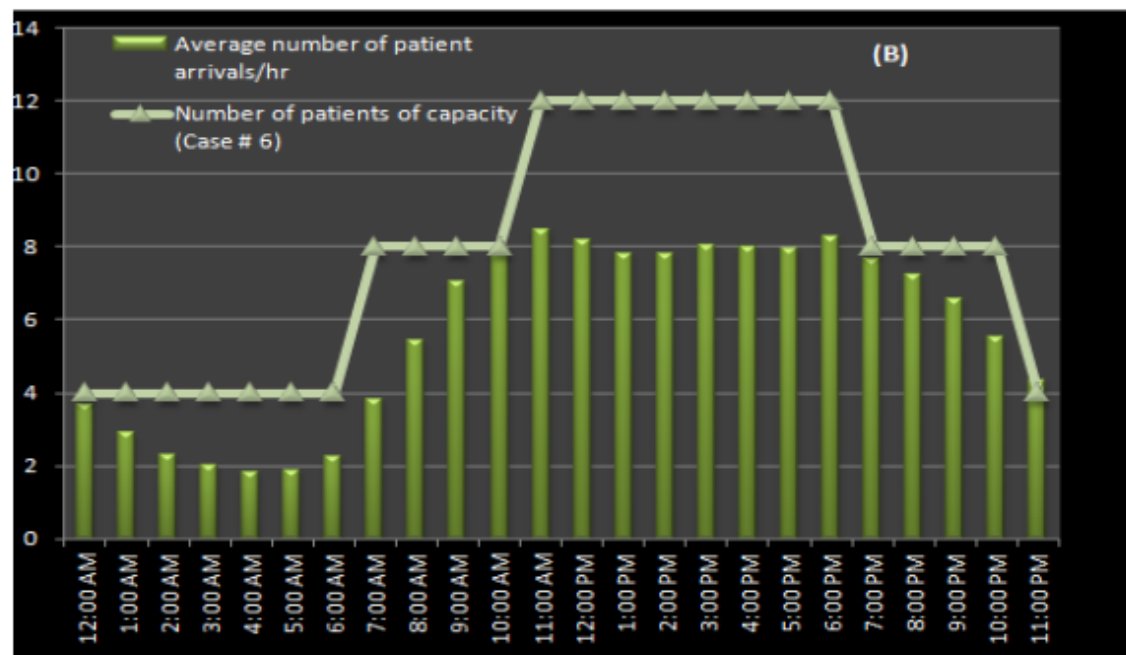
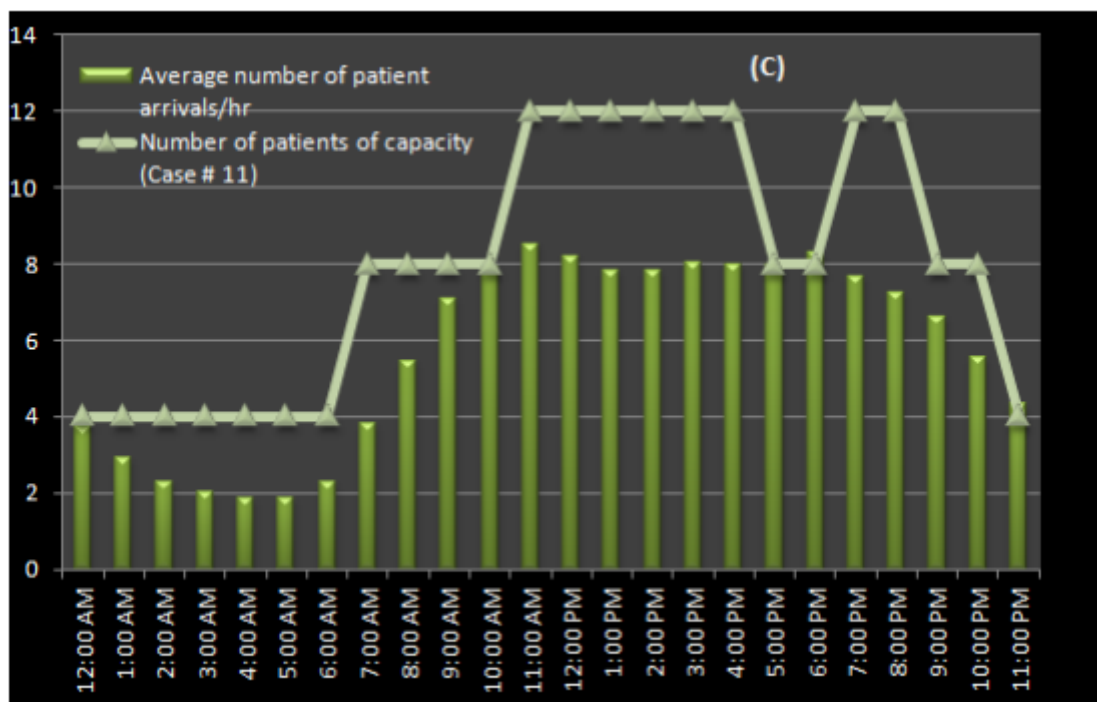


Figure 4.4(B): Number of patients of capacity plot (Case # 6, Dataset 1)



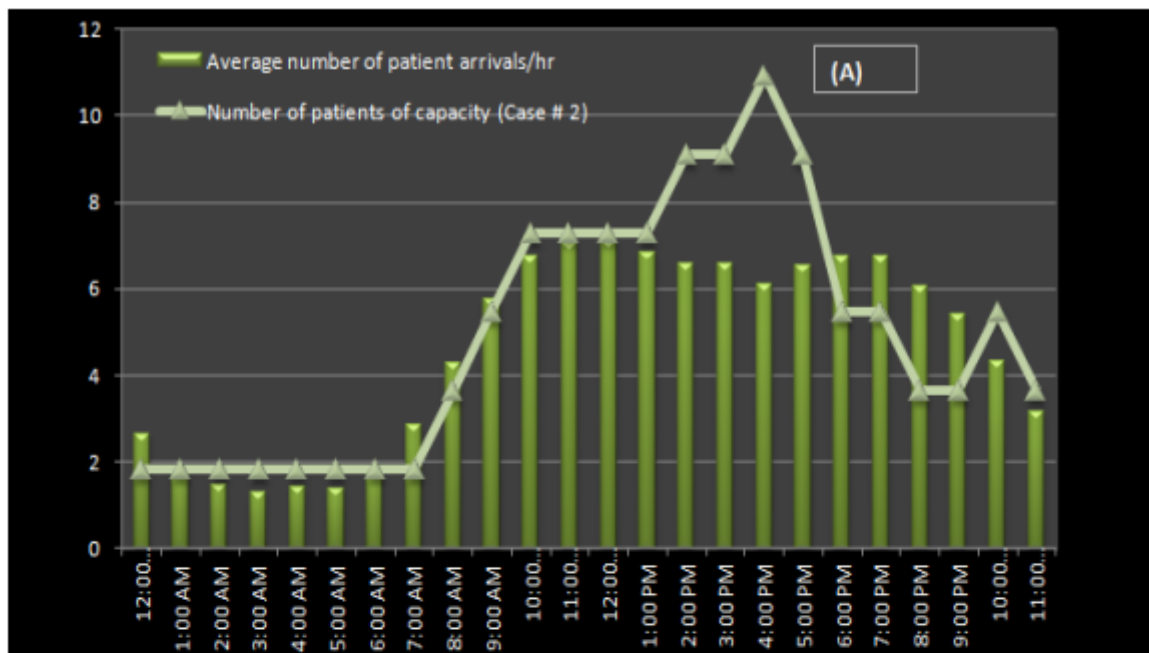
**Figure 4.4(C): Number of patients of capacity plot (Case # 11, Dataset 1)**

Столбцы на графиках выше представляют среднюю скорость прибытия пациентов в течение каждого часа, а линии - возможности врачей по обслуживанию пациентов. Как видно из трех графиков выше, когда весовые коэффициенты больше направлены на сокращение среднего времени ожидания пациента по сравнению с предпочтениями врачей (рис. 4.4(C)), генетический алгоритм генерирует графики смен, которые имеют тенденцию к увеличению пропускной способности в часы пиковых прибытий пациентов по сравнению с примером # 2, где предпочтения врачей имеют больший вес. Следовательно, добавление дополнительных мощностей приводит к уменьшению среднего времени ожидания пациента (пример № 11) по сравнению с примером № 2.

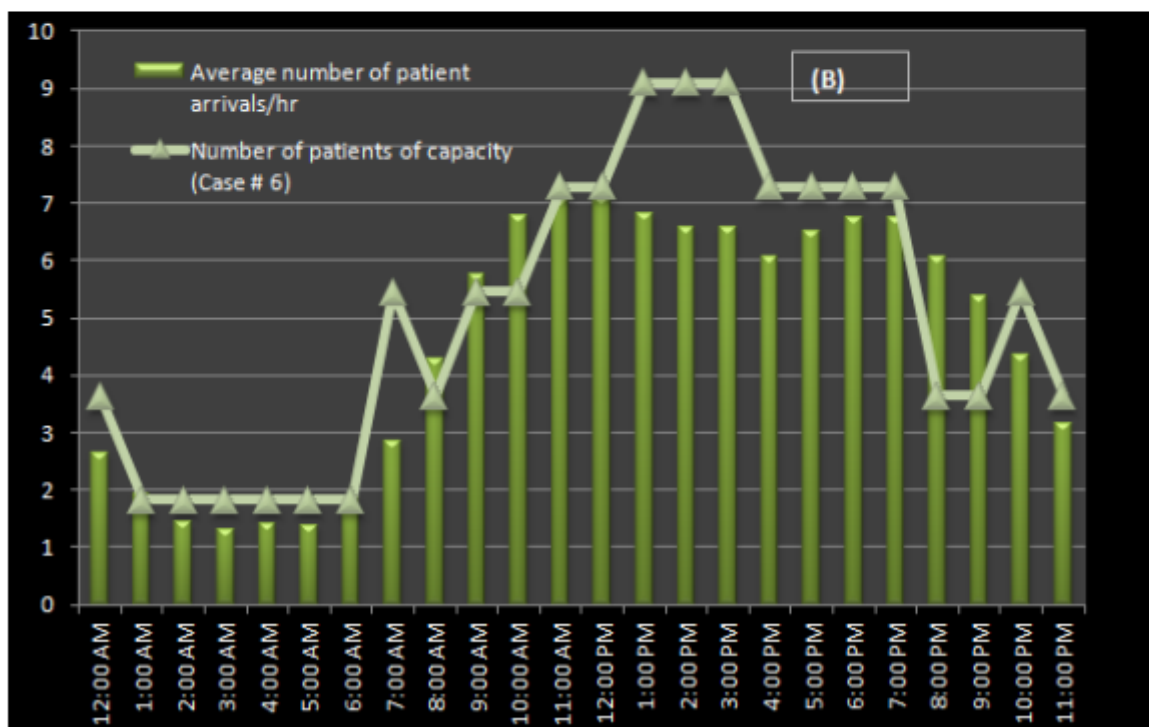
Аналогичным образом, для примеров № 2, № 6 и № 11 из набора данных 2 количество врачей, доступных в час, и «количество пациентов вместимости» показаны в таблице 4.11, а график, показывающий, как график сменности обрабатывает прибытие пациентов в каждый час, показан на рисунке 4.5(A), рисунке 4.5(B) и рисунке 4.5(C) соответственно. Графики для этих случаев можно интерпретировать так же, как они были интерпретированы для набора данных 1.

Table 4.11: Number of patients of capacity (Dataset 2)

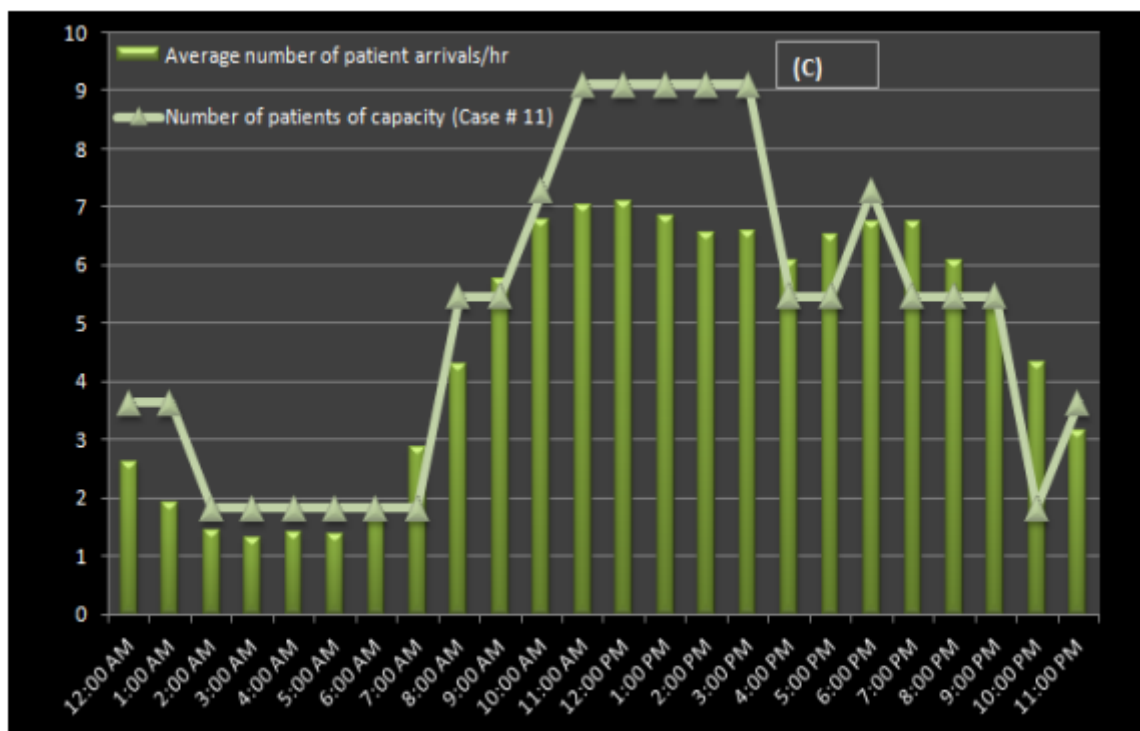
	Case # 2		Case # 6		Case # 11		
Hour of the day	Average number of patient arrivals/hr	Available physicians/hr	Number of patients of capacity	Available physicians/hr	Number of patients of capacity	Available physicians/hr	Number of patients of capacity
12:00 AM	2.621795	1	1.82	2	3.64	2	3.64
1:00 AM	1.916667	1	1.82	1	1.82	2	3.64
2:00 AM	1.448718	1	1.82	1	1.82	1	1.82
3:00 AM	1.294872	1	1.82	1	1.82	1	1.82
4:00 AM	1.403846	1	1.82	1	1.82	1	1.82
5:00 AM	1.378205	1	1.82	1	1.82	1	1.82
6:00 AM	1.839744	1	1.82	1	1.82	1	1.82
7:00 AM	2.858974	1	1.82	3	5.46	1	1.82
8:00 AM	4.288462	2	3.64	2	3.64	3	5.46
9:00 AM	5.769231	3	5.45	3	5.46	3	5.46
10:00 AM	6.769231	4	7.27	3	5.46	4	7.28
11:00 AM	7.038462	4	7.27	4	7.28	5	9.1
12:00 PM	7.083333	4	7.27	4	7.28	5	9.1
1:00 PM	6.826923	4	7.27	5	9.10	5	9.1
2:00 PM	6.557692	5	9.09	5	9.10	5	9.1
3:00 PM	6.570513	5	9.09	5	9.10	5	9.1
4:00 PM	6.076923	6	10.91	4	7.28	3	5.46
5:00 PM	6.512821	5	9.09	4	7.28	3	5.46
6:00 PM	6.730769	3	5.45	4	7.28	4	7.28
7:00 PM	6.750000	3	5.45	4	7.28	3	5.46
8:00 PM	6.064103	2	3.64	2	3.64	3	5.46
9:00 PM	5.384615	2	3.64	2	3.64	3	5.46
10:00 PM	4.339744	3	5.45	3	5.46	1	1.82
11:00 PM	3.147436	2	3.64	2	3.64	2	3.64



**Figure 4.5(A): Number of patients of capacity plot (Case # 2, Dataset 2)**



**Figure 4.5(B): Number of patients of capacity plot (Case # 6, Dataset 2)**



**Figure 4.5(C): Number of patients of capacity plot (Case # 11, Dataset 2)**

## 4.2 Выводы и дальнейшая работа

В данной работе представлен подход генетического алгоритма к решению проблемы планирования персонала. Как отмечает Michalewicz (1995a), результаты генетического алгоритма очень специфичны для конкретной задачи, и предложенный генетический алгоритм также очень специфичен для данной задачи. Кроме того, в генетический алгоритм было встроено дискретное моделирование событий для оценки среднего времени ожидания пациента. Одним из основных недостатков использования подхода взвешенной суммы является то, что объективная функция очень чувствительна к весам. Поэтому в будущем я хотел бы использовать альтернативный подход, предложенный Хаджелой и Лином (Hajela and Lin, 1992), в котором за один прогон может быть получено несколько решений. Кроме того, в данной задаче рассматривается только общее расписание работы врача. В будущем я хотел бы модифицировать свой генетический алгоритм таким образом, чтобы он мог составлять расписания для каждого отдельного врача.