

ПРИМЕНЕНИЕ АЛГОРИТМА ПОИСКА С ЗАПРЕТАМИ В ЗАДАЧЕ АВТОМАТИЗИРОВАННОГО СОСТАВЛЕНИЯ ОПТИМАЛЬНОГО ШТАТНОГО РАСПИСАНИЯ

Н.И. Боровых, О.В. Красоткина

Выполнена математическая постановка задачи как задачи бинарного программирования. Рассмотрены существующие методы решения задачи составления расписания: точные и приближенные. Выбран приближенный метаэвристический метод – поиск с запретами. Разработано приложение на ЯП Java, реализующее выбранный метод с применением библиотеки OptaPlanner, и произведено его экспериментальное исследование: сравнение близости полученного результата к решению на основе точного метода ветвей и границ, составление и анализ недельного расписания.

Ключевые слова: составление расписания, бинарное программирование, метаэвристики, поиск с запретами, метод ветвей и границ, OptaPlanner.

В крупных организациях одной из наиболее важных проблем является проблема управления персоналом. В частности встаёт задача о составлении расписания рабочих смен сотрудников. Оптимально составленное расписание позволяет организации уменьшить затраты по привлечению сотрудников, и тем самым увеличить свою экономическую эффективность.

В данной статье рассмотрена задача составления оптимального штатного расписания работы кассиров. Эта задача актуальная в современном мире, т.к. всё большую популярность приобретают крупные сетевые магазины самообслуживания, где все выбранные товары оплачиваются покупателем на кассе. В разные дни недели и в разное время суток в магазине может находиться разное количество покупателей. При недостаточном количестве кассиров или в случае недостаточной скорости работы возникают очереди, что сказывается на качестве обслуживания и приводит к потере покупателей. С другой стороны, если кассиров больше, чем необходимо, возникают неоправданные расходы на оплату времени работы лишних кассиров.

Для повышения лояльности покупателей и минимизации расходов на оплату работы кассиров следует оптимизировать расписание с учётом возможного трафика покупателей и особенностей работы каждого кассира.

На практике подобная задача часто решается человеком вручную, что занимает много времени и не гарантирует оптимальности полученного расписания. Целью данной работы является автоматизация процесса составления оптимального штатного расписания, выбор подходящий метода решения задачи и разработка приложения, реализующего выбранный метод.

Исходными данными в рассматриваемой задаче является информация о кассирах, количество доступных рабочих мест (касс) и прогноз потока покупателей. Длительные наблюдения за потоком покупателей на различных торговых предприятиях и анализ соответствующих статистических данных свидетельствуют о том, что характер распределения числа покупателей в зависимости от времени суток и времени года является весьма устойчивым [7], поэтому данная информация может быть положена в основу составления оптимального штатного расписания. Для каждого работника указывается его производительность и стоимость часа труда.

Итак, необходимо составить оптимальное расписание смен кассиров в магазине самообслуживания. Под оптимальным, в данном случае, будем понимать расписание, минимизирующее затраты магазина по привлечению работников к выполнению работ при условии выполнения сотрудниками всего объема работ и соблюдении остальных ограничений, присутствующих в задаче. Весь интервал составления расписания разбит на элементарные промежутки времени – дискретность, с которой составляется расписание. Примем дискретность составления расписания равной 15 минутам. А количество элементарных промежутков времени в интервале планирования равным 672, т.е. будем решать задачу составления недельного расписания.

В задаче составления расписания, могут присутствовать следующие ограничения:

1. ограничение на число одновременно работающих сотрудников;
2. ограничение на максимальную и минимальную продолжительность смены;
3. ограничение на общую продолжительность работы для каждого сотрудника в сутки и в течение всего периода составления расписания;
4. ограничение на продолжительность отдыха между двумя сменами;
5. ограничение, учитывающее невозможность назначения сотрудника в определенные интервалы (сотрудник в отпуске, на больничном и т.п.);
6. ограничение, учитывающее предпочтения работников относительно времени работы, количества и чередования рабочих и выходных дней и т.п.

В рамках данной работы не будем учитывать всё многообразие ограничений. Выделим следующие основные ограничения:

1. ограничение на выполнения плана, т.е. количество работающих сотрудников в каждый элементарный промежуток времени, должно быть достаточным для обслуживания прогнозируемого числа клиентов;
2. ограничение на количество назначений для каждого работника в сутки (не более одного раза);
3. ограничение на максимальное и минимальное количество одно-

временно работающих сотрудников;

4. ограничение на смены, т.е. работник может быть назначен только на смену определенной длины;

Данная задача может быть сформулирована в виде задачи бинарного программирования (задачи целочисленного линейного программирования при условии бинарности целевых переменных).

Для математической формализации задачи введем следующие обозначения.

Пусть D – количество суток в рассматриваемом периоде, тогда конкретные сутки будут обозначаться $d = 1..D$. Сутки состоят из T элементарных интервалов $t = 1..T$.

Т.к. требуется, чтобы работник мог быть назначен только на смену определенной длины, заранее нужно определить S_d – количество возможных смен в сутках d (например, если будем рассматривать смены по 4 часа, т.е. по 16 элементарных интервалов, и смена может начаться в 00.00, 00.15, 00.30 и т.д., получаем 81 смену $S_d = 81$), конкретную смену из всех возможных будем определять как $s_d = 1..S_d$, продолжительность (количество элементарных интервалов) смены s_d обозначим P_{s_d} , при этом значения P_{s_d} могут выбираться только из фиксированного множества, например $\{16, 24, 32\}$. Введем понятие матрицы смен для суток $d - A^d$, в которой каждый элемент $a_{s_d,t}^d \in \{0,1\}$ – бинарная переменная, равная 1, если t -ый интервал принадлежит смене s_d , 0 – если не принадлежит. Вид матрицы A^d задаёт режим работы магазина и возможные режимы работы кассиров, т.е., например, при переходе с круглосуточного режима работы магазина на сокращенный изменяется только вид матрицы A , дополнительных ограничений при этом вводить не нужно. Также добавление новых продолжительностей смен работы кассиров изменяет только размерность A^d без изменения структуры задачи. Для упрощения будем считать, что матрицы A^d для всех d одинаковы, тогда будем обозначать их просто A .

Обозначим $p_{t,d}$ – объем работ, который должен быть выполнен за каждый элементарный промежуток времени в сутках d ;

Введем обозначение для работника $n = 1..N$, где N – общее число работников. Определим $N_{t,d}^{\max}, N_{t,d}^{\min}, t = 1..T, d = 1..D$ – максимальное и минимальное число работников, которые могут быть привлечены в момент времени t в сутки d . Для каждого работника определены c_n – затраты по его привлечению на один элементарный интервал и λ_n – производительность за один элементарный интервал.

Тогда $x_{n,s_d,d} \in \{0,1\}, n = 1..N, s_d = 1..S_d, d = 1..D$ – целевая бинарная

переменная, показывающая назначение или неназначение работника n на смену s_d в сутки d .

С учетом сделанных обозначений целевая функция имеет вид:

$$\sum_{d=1}^D \sum_{n=1}^N \sum_{s_d=1}^{S_d} c_n P_{s_d} x_{n,s_d,d} \rightarrow \min, \quad (1)$$

Ограничение на выполнение плана:

$$\sum_{n=1}^N \sum_{s_d=1}^{S_d} a_{s_d,t} \lambda_n x_{n,s_d,d} \geq p_{t,d}, t=1..T, d=1..D, \quad (2)$$

Ограничение на количество смен в сутки для каждого кассира (не более одной смены):

$$\sum_{s_d=1}^{S_d} x_{n,s_d,d} \leq 1, n=1..N, d=1..D, \quad (3)$$

Ограничение на количество одновременно работающих кассиров:

$$N_{t,d}^{\min} \leq \sum_{n=1}^N \sum_{s_d=1}^{S_d} a_{s_d,t} x_{n,s_d,d} \leq N_{t,d}^{\max}, t=1..T, d=1..D, \quad (4)$$

Итак, задача сформулирована как задача бинарного программирования, т.е. необходимо решать задачу с огромным количеством переменных, которая в общем виде является NP-полной.

Существующие точные методы решения задач целочисленного программирования, являются в той или иной степени реализацией метода ветвей и границ и фактически представляют собой направленный перебор возможных решений. Классический метод ветвей и границ потенциально может пройти все 2^n бинарных векторов, где n – число переменных, поэтому для реальной задачи, где количество переменных исчисляется тысячами, поиск оказывается неосуществимым из-за слишком большого объема вычислительной работы.

Применение точных методом также может быть неоправданным ввиду возможных неточностей построенной модели. Поэтому для решения задач, представляющих практический интерес, целесообразнее использовать приближенные методы. Эти методы позволяют найти приемлемое решение (может и не быть оптимальным, поэтому методы и называются приближенными) NP-полной задачи большой размерности за приемлемое время. Решение задачи производится обычно в два этапа: построение начального решения и улучшение начального решения. На первом этапе широко используются эвристические алгоритмы – алгоритмы, основанные на правдоподобных, но не обоснованных строго предположениях о свойствах оптимально решения задачи (например, жадные алгоритмы). На втором этапе используются алгоритмы локальной оптимизации (локального поиска). У алгоритмов локального поиска есть существенный недостаток – они

останавливаются в локальном оптимуме, поэтому глобальный оптимум может быть не достигнут.

В последние десятилетия появился новый подход к применению приближенных методов, который пытается комбинировать базовые эвристические методы на более высоком уровне для эффективного исследования пространства поиска. Эти методы называются метаэвристиками.

Многие метаэвристические методы основаны на локальном поиске, но построены так, чтобы избежать остановки в локальном оптимуме, и таким образом обследовать больше решений из пространства поиска в надежде найти хорошее решение.

Для решения поставленной задачи составления оптимального штатного расписания, выберем метаэвристику поиск с запретами (Tabu Search), разработанную Фредом Гловером[3], опираясь на результаты исследования[1]. В данном исследовании поиск с запретами сравнивался с четырьмя метаэвристиками (оптимизаций муравьиной колонии, эволюционные вычисления, итеративный локальный поиск, имитация отжига) применительно к двум задачам составления расписания (Group Shop Scheduling и Job Shop Scheduling) и показал лучшие результаты на множестве тестов. Множество других примеров[2][4][5][6] подтверждают успешность применения данной метаэвристики для составления разного рода расписаний.

Поиск с запретами принадлежит к классу стратегий локального поиска. Основная идея техники поиска с запретами – избегать циклов (посещения одних и тех же решений) в процессе поиска, при помощи списка запретов (tabu list), что позволяет не останавливаться в локальных оптимумах. В списке запретов хранится информация об истории поиска в виде фиксированного числа просмотренных решений. Принятие решения на следующей итерации зависит не только от его качества, но и от информации об истории поиска. На хранимые в списке запретов решения, накладывается табу, и они не могут быть выбраны при анализе окрестности. В этой технике на каждой итерации генерируется полная окрестность текущего решения. Лучшее решение (не табу) из окрестности выбирается для следующей генерации.

Для разработки приложения выбрана библиотека OptaPlanner версии 5.5.0, в которой реализуются популярные оптимизационные алгоритмы (эвристические и метаэвристические), в частности алгоритм поиска с запретами.

OptaPlanner – сводобно распространяемая библиотека, написанная на Java, выпущенная в соответствии с Apache Software License. Выбор обусловлен тем, что другие аналогичные по функциональности и удобству бесплатные библиотеки отсутствуют. Для написания программы будем использовать свободную интегрированную среду разработки приложений на языке программирования Java – NetBeans 7.3.

В процессе программной реализации решения задачи можно выделить несколько ключевых этапов.

Во-первых, решаемую задачу необходимо представить в виде объектной модели, т.е. описать её в объектно-ориентированном стиле в виде классов. Для представления данных задачи требуются классы: «работник» (Employee), «дата» (Date), «время смены» (ShiftTime), «смена» (определяется датой и временем смены) (Shift). Также необходим класс «назначение на смену» (ShiftAssignment), он будет являться сущностью планирования. Это означает что значение определенных полей (переменных планирования) экземпляров этого класса будут изменять в процессе оптимизации. Класс «назначение на смену» будет содержать поля: смена, номер кассы, работник. В данном случае переменной планирования будет являться поле «работник» (на работу в определенной смене на определенной кассе можно назначить одного работника из множества доступных работников для назначения или не назначать никого).

Во-вторых, необходимо определить правила вычисления счета. В библиотеке OptaPlanner отсутствует понятие целевой функции. Для сравнения решений по качеству используется счет двух уровней. Лучшим считается решение с самым высоким счетом: сначала максимизируется счет первого уровня (hardscore), потом второго (softscore). Счет первого уровня будем вычислять как сумму штрафов за нарушение ограничений (2,3), взяв со знаком «минус» (ограничение (4) не учитывается, т.к. при данной объектной модели задачи не может быть нарушено). Тогда допустимое решение, т.е. решение не нарушающее представленных в задаче ограничений, имеет счет 0. А правило для вычисления счета можно записать следующим образом:

$$\begin{aligned} \text{hardscore} = & \left(\sum_{d=1}^D \sum_{t=1}^T \sum_{n=1}^N \sum_{s_d=1}^{S_d} \text{if}(a_{s_d,t} \lambda_n x_{n,s_d,d} - p_{t,d} < 0) ? (a_{s_d,t} \lambda_n x_{n,s_d,d} - p_{t,d}) : 0 \right) + \\ & + \left(\sum_{d=1}^D \sum_{n=1}^N \sum_{s_d=1}^{S_d} \text{if}(1 - x_{n,s_d,d} < 0) ? (1 - x_{n,s_d,d}) : 0 \right) \end{aligned}$$

Тогда учесть необходимость минимизации целевой функции (1) можно с помощью счета второго уровня, который будем вычислять следующим образом:

$$\text{softscore} = - \sum_{d=1}^D \sum_{n=1}^N \sum_{s_d=1}^{S_d} c_n P_{s_d} x_{n,s_d,d}$$

Созданное приложение считывает данные из файлов в формате Microsoft Excel, решает задачу составления расписания на основе метода поиска с запретами и имеет графический интерфейс, основанный на библиотеке Swing, для вывода полученного результата.

Проведем эксперимент, чтобы выяснить насколько результат, полу-

ченный с помощью поиска с запретами близок к оптимальному результату, т.е. тому, который мы получим, используя точный метод – метод ветвей и границ.

Решение задачи точным методом производится в пакете MATLAB версии 7.11(R2010b), с использованием встроенного набора инструментов Optimization Toolbox, в котором содержится функция для решения задачи бинарного программирования `binprog`, реализующая метод ветвей и границ.

Эксперимент проводится на машине Intel Core i3 CPU, 2.27 ГГц, ОЗУ 3 Гб.

Учитывая, что точные методы не применимы для реальных задач из-за огромного пространства поиска, ограничим количество переменных. Проведем эксперимент 20 раз, генерируя данные случайным образом (рис.1).

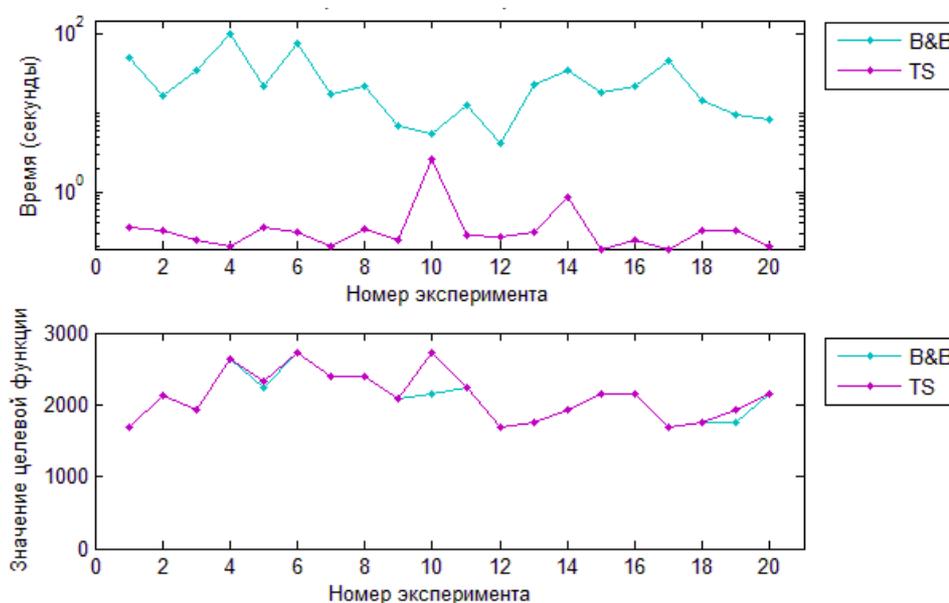


Рис. 1. Результат эксперимента

Вычислим абсолютное (по формуле (5)) и относительное (по формуле (6)) среднеквадратичное отклонение результата полученного приближенным методом от оптимального решения.

$$S_{абс} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_i)^2}, \quad (5)$$

$$S_{отн} = \frac{S_{абс}}{\sqrt{\frac{1}{n} \sum_{i=1}^n \bar{x}_i}} \cdot 100\%, \quad (6)$$

где x_i – значение целевой функции, полученное в i -ом эксперименте методом поиска с запретами; \bar{x}_i – значение целевой функции, полученное в i -ом эксперименте методом ветвей и границ.

Получены отклонения: $S_{abc} = 131.45$, $S_{отн} = 6.3\%$.

Можно сделать вывод, что качество решения, получаемого приближенным методом достаточно хорошее, и решение найдено за приемлемое время.

Теперь протестируем приложение на реальной задаче: построим недельное расписание.

На рис. 2 отражено как меняется счет *hardscore* в процессе решения задачи: начиная от огромного количества нарушенных ограничений до удовлетворения всех жестких ограничений задачи, т.е. до получения допустимого решения.

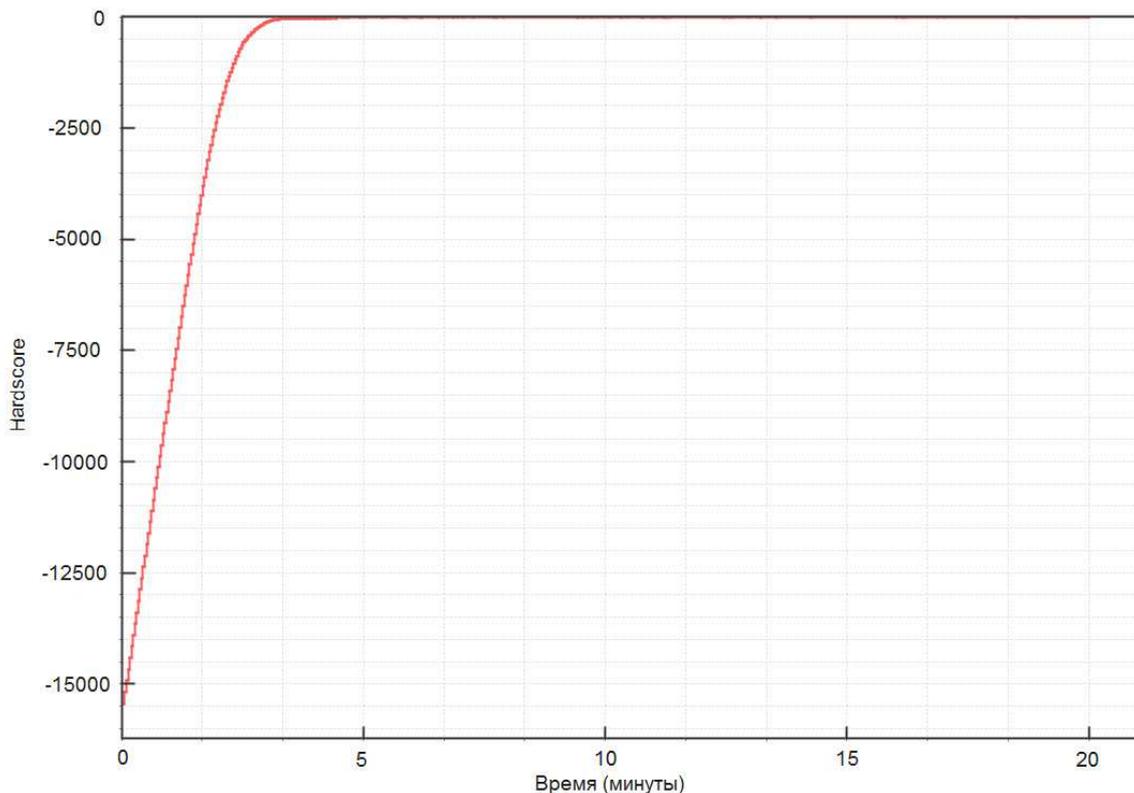


Рис. 2. Зависимость счета *hardscore* от времени

На рис. 3 показано как меняется счет *softscore* в процессе решения задачи: от нулевого (когда не назначен ни один работник и затраты равны 0) до счета, показывающего затраты за неделю на привлечение работников при выполнении поставленных условий. На рис. 4 показано, как обеспечивается потребность в производительности труда для обслуживания прогнозируемого числа клиентов.

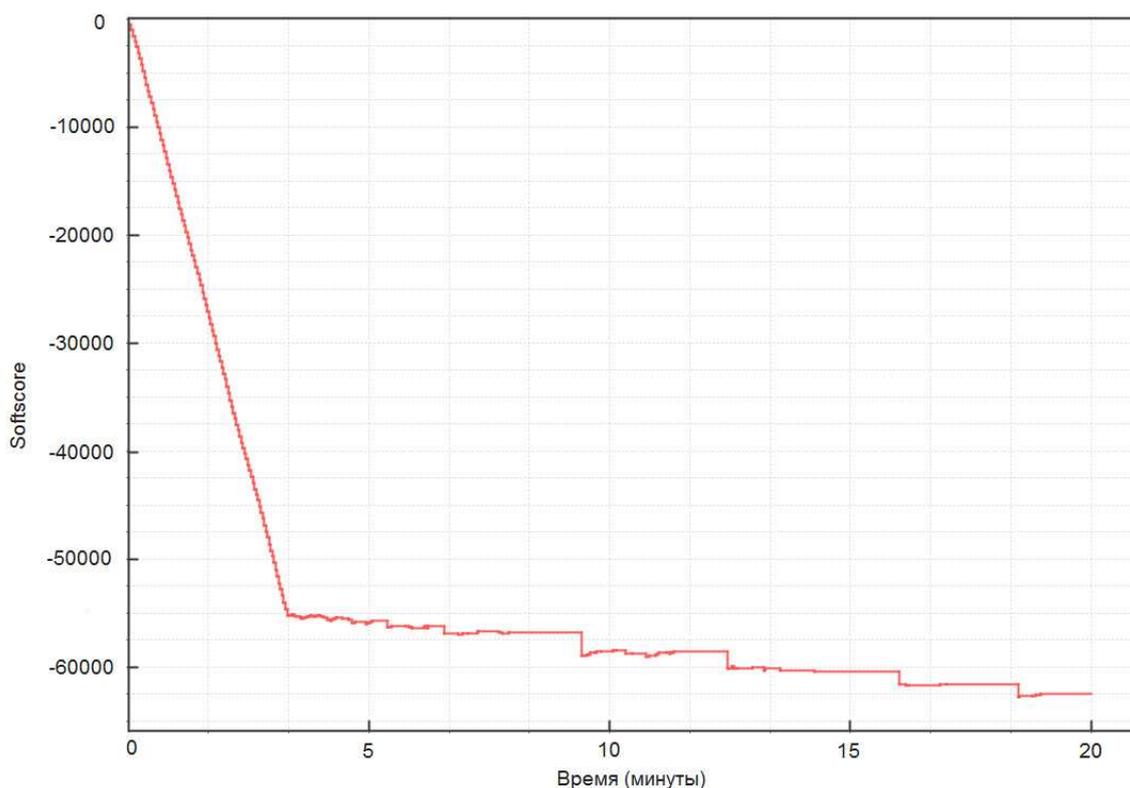


Рис. 3. Зависимость счета softscore от времени

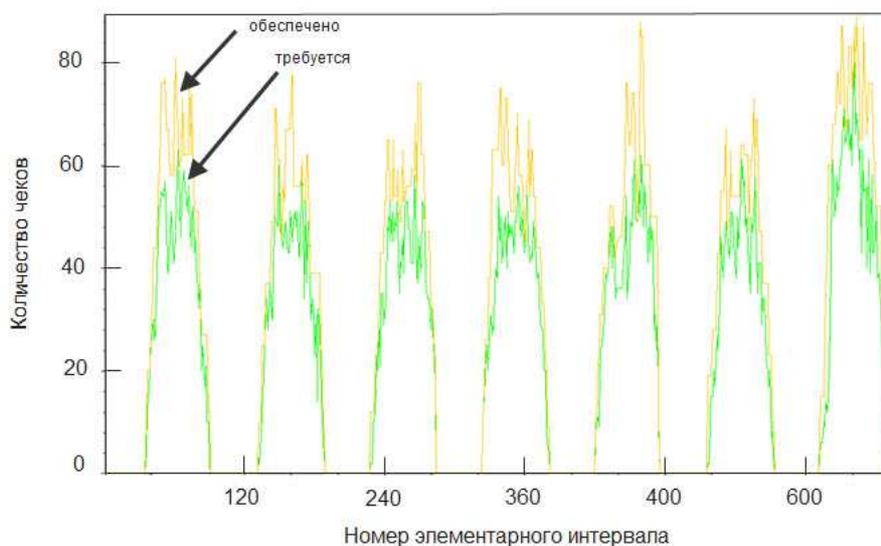


Рис. 4 – Соотношение потребности в производительности и обеспеченной производительности

Итак, с помощью разработанного приложения на основе выбранного метаэвристического метода поиска с запретами получено расписание, которое удовлетворяет описанным ограничениям задачи за приемлемое время.

Список литературы

1. Blum C. Metaheuristics for group shop scheduling : дис. – Université Libre de Bruxelles, 2002.
2. Burke E.K., Kendall G., Soubeiga E.. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9(6):451–470, 2003.
3. Glover F., Laguna M. Tabu search. Dordrecht: Kluwer Acad. Publ., 1997.
4. Luca Di Gaspero and Andrea Schaerf. Tabu search techniques for examination timetables. *Lecture Notes in Computer Science*, 2079:104–117, 2001.
5. Musliu N. Applying tabu search to the rotating workforce scheduling problem //The 5th Metaheuristics International Conference; MIC. 2003. Т. 3.
6. White G.M., Xie B.S., Zonjic S. Using tabu search with longer-term memory and relaxation to create examination schedules. *European Journal of Operational Research*, 153:80–91, 2004.
7. Пиликов Н. П. Секретные технологии организации труда у мировых гигантов розничной торговли Carrefour и Auchan [Электронный ресурс] // Планирование и организация многосменного режима работы [сайт]. [2004]. URL: http://www.mnogosmenka.ru/pilikov/indi_graphiki.htm (дата обращения 10.04.2013).

Боровых Наталья Игоревна, natali.bor1992@mail.ru, Россия, Тула, Тульский государственный университет,

Красоткина Ольга Вячеславовна, канд. физ.-мат. наук, доц., o.v.krasotkina@yandex.ru, Россия, Тула, Тульский государственный университет.

APPLYING TABU SEARCH TO THE AUTOMATED OPTIMAL STAFF SCHEDULING PROBLEM

N.I.Borovykh, O.V.Krasotkina

Mathematical model of the problem as a binary programming problem is provided. Survey of existing methods (exact and approximate) of scheduling is given. Approximate metaheuristic method – Tabu Search – is chosen. Created Java application with OptaPlanner library using applies chosen method. Several experiments were implemented: comparison of approximate solution and exact solution found with branch and bound method, weekly schedule creating and analyzing.

Key words: scheduling, binary programming, metaheuristics, tabu search, branch and bound, OptaPlanner.

Borovykh Natalia Igorevna, natali.bor1992@mail.ru, Russia, Tula, Tula State University,

Krasotkina Olga Vyacheslavovna, candidate of physical and mathematical science, docent, o.v.krasotkina@yandex.ru, Russia, Tula, Tula State University