

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики, математики и электроники
Факультет информатики
Кафедра технической кибернетики

Отчет

о научно-исследовательской работе бакалавра

**Проектирование системы для оптимального
распределения работ по исполнителям**

Направление подготовки: 01.03.02 Прикладная математика и информатика:
Профиль "Компьютерные науки"

Студент: Анурин Андрей Сергеевич

Семестр: 7

Тема научно-исследовательской работы: Проектирование системы для
оптимального распределения работ по исполнителям

Научный руководитель

Белоусов А.А.

“ ____ ” _____ 20 ____ г.

Студент

Анурин А.С.

“ ____ ” _____ 20 ____ г.

РЕФЕРАТ

Отчет по научно-исследовательской работе бакалавра: 14 с., 8 рисунков, 0 таблиц, 9 источников, 0 приложений.

ТЕОРИЯ ОПТИМИЗАЦИИ, ТЕОРИЯ РАСПИСАНИЙ, МЕТОД СИМУЛЯЦИИ ОТЖИГА, ГЕНЕТИЧЕСКИЙ АЛГОРИТМ, ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ, ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

Объектом исследования являются различные алгоритмы дискретной оптимизации.

Цель работы — изучение методов дискретной оптимизации, научной литературы на тему задачи составления расписаний и распределения работы по исполнителям и сравнение различных методов.

Рассмотрены несколько семейств методов оптимизации, применимых к исследуемой проблеме, и необходимость модификации различных параметров. Были проанализированы сильные и слабые стороны каждого метода, подготовлен теоретический фундамент для дальнейшей реализации программного комплекса.

СОДЕРЖАНИЕ

Введение	4
1 Обзор метода симуляции отжига	5
2 Обзор генетических и эволюционных методов	8
3 Обзор методов линейного программирования и модификаций	11
Заключение.....	13
Список использованных источников.....	14

ВВЕДЕНИЕ

Некоторые исследователи называют происходящее в последние десятилетия бурное развитие и интеграцию в повседневную жизнь информационных технологий новой вехой в истории человечества: «информационной эрой». Однако, несмотря на подобного рода утверждения, во многих областях человеческой деятельности оптимизационный потенциал информационных технологий раскрыт не полностью, а основным инструментом остается ручное применение человеческого ноу-хау.

Многие дисциплины математики и компьютерных наук способны упростить, улучшить или иным способом оптимизировать производственный процесс. Теория вероятностей позволяет оценивать риски и вероятности, статистика позволяет делать качественные выводы из накопленных данных, машинное обучение позволяет строить модели, которые способны обучаться на этих данных и решать очень широкий круг задач.

В данной работе мы остановимся на теории оптимизации как на дисциплине, наиболее подходящей для решения конкретной задачи оптимального распределения задач по исполнителям. Эта задача родственна задаче, которой занимается теория расписаний, но мы будем рассматривать несколько более общий случай.

Практически любой процесс, в котором некоторому множеству работников необходимо выполнить некоторый объем работы, потенциально может быть улучшен составлением оптимального распределения. Ввиду острой актуальности этой проблемы за последние несколько десятилетий был совершен значимый прогресс в ее научном исследовании. В данной работе будут рассмотрены некоторые из основных семейств методов решения: симуляция отжига, генетические алгоритмы и линейное программирование.

1 Обзор метода симуляции отжига

В общем виде, задача дискретной или комбинаторной оптимизации звучит так: рассматривая конечное, но, возможно, комбинаторно большое множество дискретных объектов, найти в нем оптимальный [1]. Часто полный перебор всех объектов нереалистичен из-за большого объема множества возможных решений. Например, в практической задаче, которая побудила автора выбрать именно эту тему работы, типичный размер множества решений составляет около 10^{130} .

Математически в общем виде задачу оптимизации можно сформулировать так:

$$\min_{x \in X} \{f(x)\}, \quad f: X \rightarrow \mathbb{R}_+ \quad (1.1)$$

где f — целевая функция, X — дискретное множество решений.

Метод симуляции отжига принадлежит к классу так называемых метаэвристик: очень общих алгоритмов, которые решают задачу оптимизации в виде (1.1). Он часто применяется в случаях, когда пространство решений дискретное, например, как в задаче коммивояжера [2].

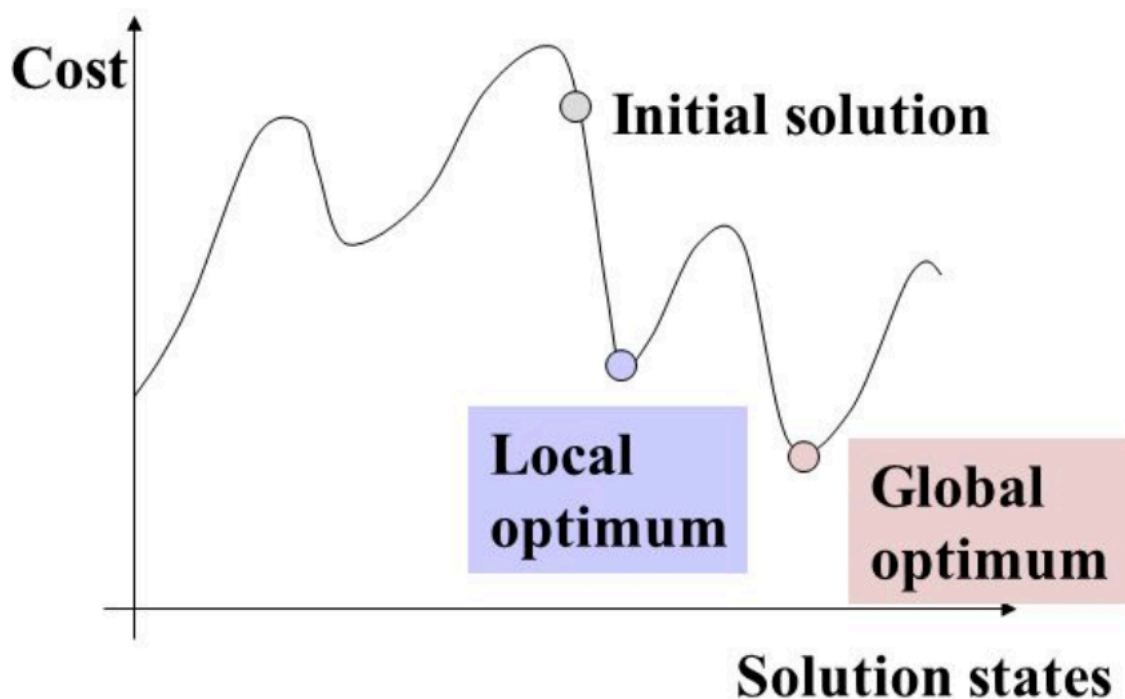


Рисунок 1 — Поиск глобального минимума

Основной мотивацией для метода симуляции отжига является поиск глобального минимума, а не локального. Рисунок 1 иллюстрирует такой поиск. Вдохновением для самого метода служил процесс отжига в металлургии, при котором материал последовательно нагревается и остужается для того, чтобы увеличить размер образуемых кристаллов в металлической решетке и уменьшить количество дефектов. Применение этого метода позволяет найти глобальный минимум функции с очень большим количеством аргументов.

Идея понижения температуры в металлургическом отжиге соответствует понижению вероятности, с которой мы перейдем в худшее решение из лучшего, в методе симуляции отжига. Самой распространенной функцией этой вероятности является $P(f_1, f_2, T) = e^{\frac{-(f_2 - f_1)}{T}}$ [3].

Важным аспектом алгоритма является выбор расписания уменьшения температуры, так как от него очень сильно зависит структура семейства получаемых решений. Рисунок 2 иллюстрирует быстрое остывание, когда у алгоритма недостаточно времени для того, чтобы полностью исследовать тонкую структуру целевой функции.

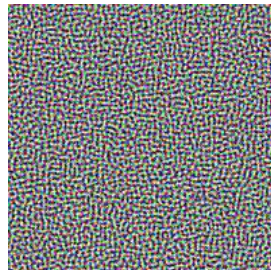


Рисунок 2 — Быстрое остывание

Рисунок 3, напротив, иллюстрирует медленное остывание, которое позволяет итоговому семейству решений захватить больше информации о структуре целевой функции.

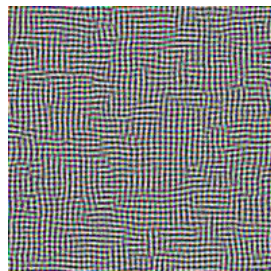


Рисунок 3 — Медленное остывание

Ввиду своей общности, алгоритм симуляции отжига хорошо применим к задаче распределения работ по исполнителям, однако, он может не приводить к удовлетворительным результатам за практические промежутки времени.

2 Обзор генетических и эволюционных методов

Так же, как и метод отжига, генетические алгоритмы относятся к классу метаэвристик, которые решают очень широкий класс задач [4]. Рисунок 4 показывает интересное частное применения генетических алгоритмов на практике.



Рисунок 4 — Антенна, спроектированная генетическим алгоритмом

Основной идеей является представление решения в виде вектора чисел, который соответствует геному в биологии, и осуществление над ним специфических операций. В генетическом алгоритме поддерживается популяция решений-кандидатов, которые каждую итерацию скрещиваются между собой. В результате скрещивания получаются решения, которые наследуют часть своей структуры от обоих родителей, при этом привнося свои собственные случайные изменения. Из нового поколения выбирается определенное количество лучших решений с точки зрения целевой функции, остальные удаляются. С новым множеством решений-кандидатов алгоритм совершает следующую итерацию. Рисунок 5 иллюстрирует этот процесс.

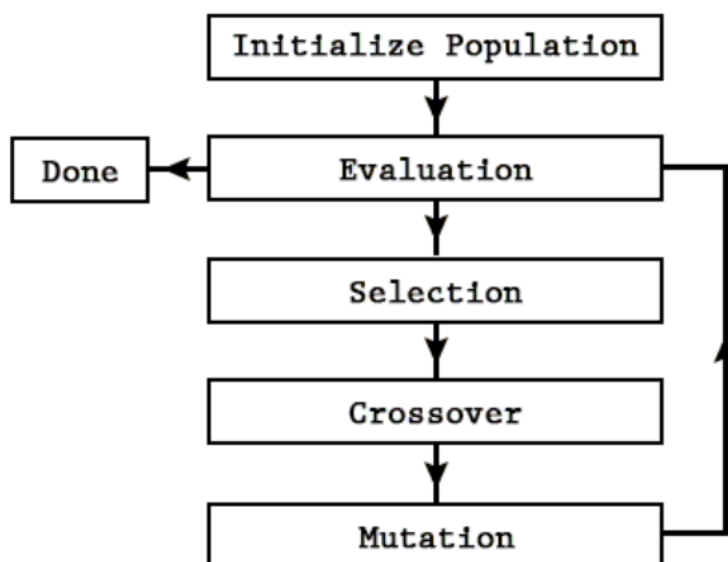


Рисунок 5 — Блок-схема работы генетического алгоритма

Огромный объем литературы [5] посвящен различным модификациям стандартного генетического алгоритма. Вводится элемент случайности, сложная структура самих популяций, новые методы селекции, кроссинговера или мутации. Существуют даже квантовые варианты генетических алгоритмов.

Генетические алгоритмы так же часто применяются на практике. Интересным примером является онлайн игра, в которой генетические алгоритмы формируют дизайн машины, которая должна проехать по полосе препятствий [6]. Рисунок 6 показывает интерфейс этой игры и несколько машин.

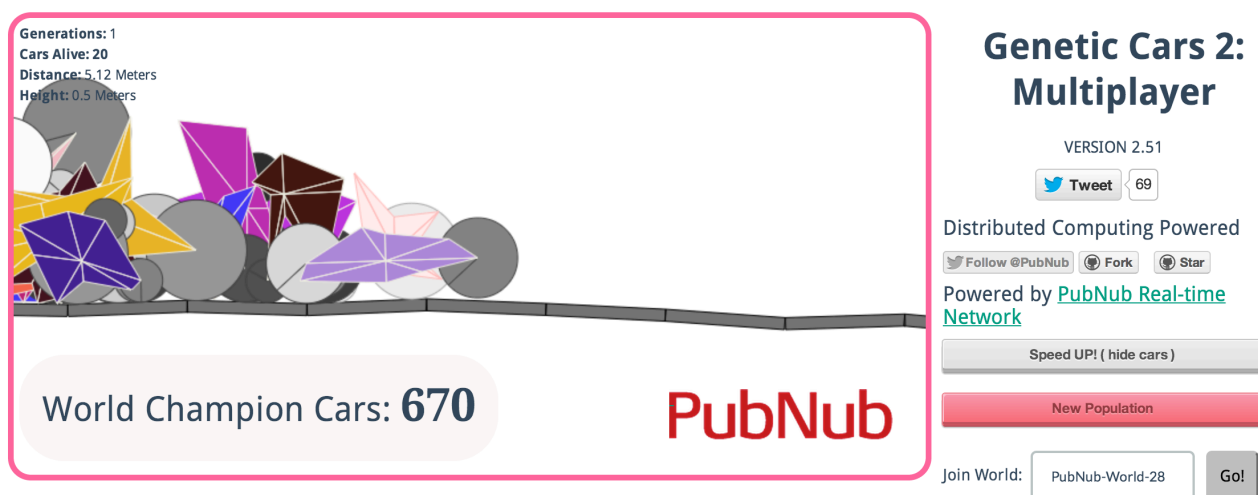


Рисунок 6 — Соревнование генетических автомобилей

Также генетические алгоритмы применяются для решения задачи распределения работы по исполнителям. Одна группа исследователей занималась оптимизацией работы медиков в госпитале, модулируя процесс

оптимизации как генетический алгоритм [7]. Классическим результатом в этой области является исследование применимости генетических алгоритмов в задаче упорядочения выполнения инструкций центрального процессора [8].

3 Обзор методов линейного программирования и модификаций

В отличие от вышеперечисленных методов, линейное программирование используется для решения задач очень специфического вида, который выражается следующим образом:

$$\begin{aligned} \min \{c^T x\}, \\ Ax \leq b, \\ x \geq 0. \end{aligned} \quad (3.1)$$

Таким образом, целевая функция зависит от аргумента линейно, а множество допустимых решений представляет из себя выпуклый многогранник. Рисунок 7 представляет из себя пример такого множества с указанием направления роста целевой функции.

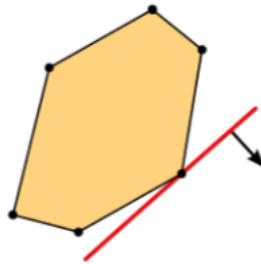


Рисунок 7 — Область решений задачи линейного программирования

Такая постановка задачи позволяет использовать множество методов, пользующихся линейностью граничных условий, например, симплекс метод или метод внутренних точек [9]. Было достигнуто множество теоретических результатов, связанных с этими методами.

Однако не каждая практическая задача может быть сведена к задаче линейного программирования. Родственной задачей является задача целочисленного программирования, в которой в формуле (3.1) все переменные являются не вещественными, а целыми числами. Такая формулировка позволяет тривиально представить задачу распределения работ по исполнителям. Пусть в матрице A на i -ой строчке и j -ом столбце стоит единица тогда и только тогда, когда в итоговом расписании i -ый исполнитель берет на себя j -ую задачу.

Следующая проблема возникает с видом целевой функции, ведь не каждая значимая на практике задача сводима к линейным отношениям искомого значения от входного.

В общем случае задача целочисленного программирования является NP-сложной. Тем не менее, существует набор алгоритмов и программных инструментов, которые занимаются решением таких задач. Одним из самых популярных таких инструментов является «COIN-OR CBC» ввиду открытости своего исходного кода и приемлемой производительности. Существует целая индустрия коммерческих систем решения задач целочисленного программирования, с десятками различных платных программ.

Как ни странно, основой большинства из них является тот же самый симплекс метод, который можно модифицировать таким образом, чтобы релаксировать решения более широкой гладкой задачи на дискретную сетку. Рисунок 8 иллюстрирует такую релаксацию.

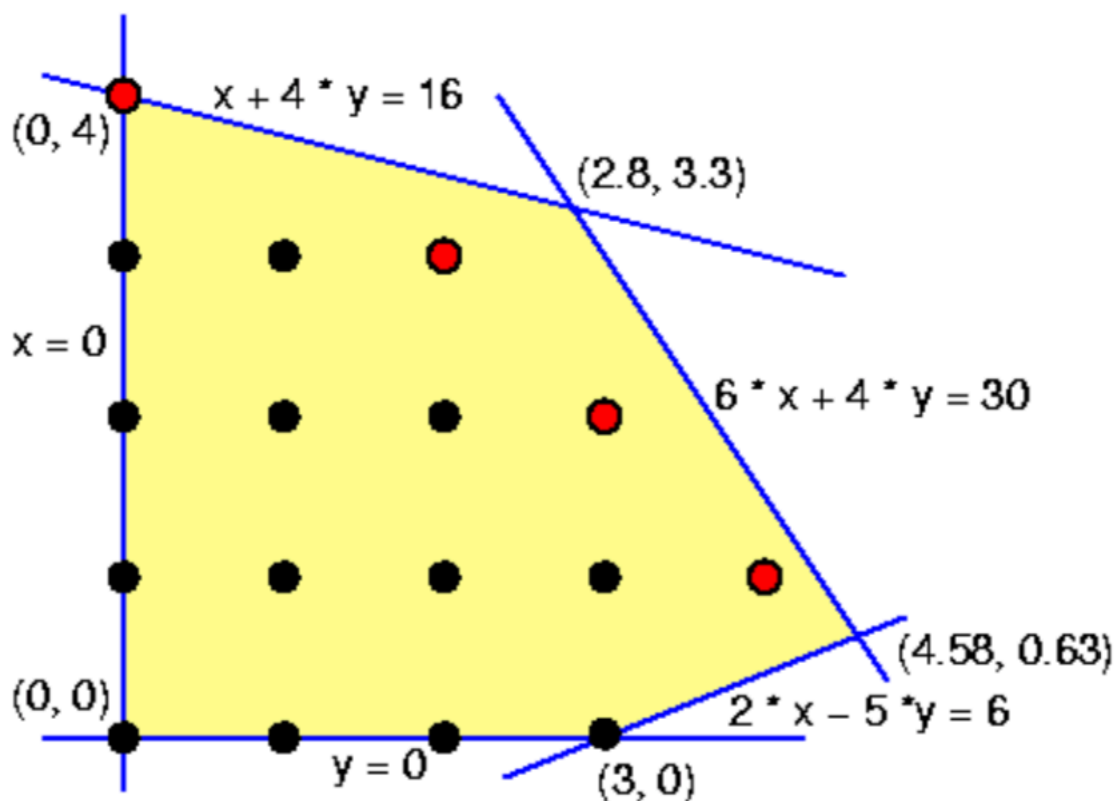


Рисунок 8 — Расширение задачи целочисленного программирования

ЗАКЛЮЧЕНИЕ

В данной работе был произведен обзор задачи распределения работ по исполнителям и основных подходов к ее решению. Для каждого метода были приведены его плюсы и минусы, была проанализирована существующая научная литература и актуальные модификации методов дискретной оптимизации.

Многие методы оказались либо слишком общими для практической применимости, либо слишком специализированными и рассчитанными на другую природу входных данных. Тем не менее, было выделено множество алгоритмов, обладающих потенциалом и достаточной гибкостью для применения для решения практической задачи.

В целом, был заложен теоретический фундамент для дальнейшей реализации алгоритмов оптимизации и программного комплекса для решения задачи оптимального распределения работ по исполнителям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Nocedal J. Numerical Optimization [Текст] / Nocedal J., Wright S. – Нью Йорк: Springer, 2006. – 651 с.
- 2 Chibante R. Simulated Annealing, Theory with Applications [Текст] / Chibante R. 2011. – 300 с.
- 3 Kirkpatrick S. Optimization by Simulated Annealing [Текст] / Kirkpatrick S., Gellat Jr C., Vecchi M.: Science, 1983. – 680 с.
- 4 Kalyanmoy D. Handbook of Evolutionary Computation [Текст] / Kalyanmoy D., Spears W.: Institute of Physics Publishing, 1997.
- 5 Mutingi M. Grouping Genetic Algorithms: Advances and Applications [Текст] / Mutingi M., Mbohwa C.: Springer International Publishing, 2017. – 243 с.
- 6 Genetic Cars 2 (Globally Distributed) [Электронный ресурс] – 2018. – URL: <https://github.com/pubnub/genetic-car-2> (дата обращения: 22.12.2018).
- 7 Йех Д. Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department [Текст] – Тайвань: Elsevier, 2006. – 1073 с.
- 8 Xoy E. A genetic algorithm for multiprocessor scheduling [Текст]: IEEE, 1994.
- 9 G. S. Linear and Integer Optimization: Theory and Practice [Текст] / Sierksma G., Zwols Y.: CRC Press, 2015.