

УДК 658.58

Разработка алгоритма составления расписания ремонтных работ оборудования с помощью методов программирования в ограничениях

Новикова Татьяна Алексеевна boiko_tatjana@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Россия

Аннотация. Сформулирована актуальность разработки автоматизированных решений составления расписаний ремонтных работ для промышленного оборудования, основанных на методах теории расписаний, программирования в ограничениях (Constraint Programming, CP) и задаче выполнимости булевых формул (Satisfiability Problem, SAT). Определено понятие программирования в ограничениях, подразумевающее задание переменных, их значений и ограничений. Описан переход от задачи программирования в ограничениях к задаче выполнимости булевых формул. Разработан алгоритм формирования плана-графика ремонтных работ, основанный на open source Python-библиотеке ORtools, которая совмещает в себе методы CP и SAT. Обоснована необходимость добавления в базовый функционал библиотеки рекурсивного метода, позволяющего составлять расписание без априорного задания точного числа ремонтов.

Ключевые слова: планирование ремонтных работ, программирование в ограничениях, задача выполнимости булевых формул, ORtools

Development of an algorithm for scheduling equipment repairs using constraint programming methods

Novikova Tatiana Alekseevna boiko_tatjana@mail.ru

BMSTU, Moscow, Russia

Abstract. The article states the relevance of developing automated solutions for scheduling repairs of industrial equipment based on the methods of scheduling theory, constraint programming (CP) and the satisfiability problem of Boolean formulas (SAT). The concept of constraint programming is defined, which implies the definition of variables, their values and constraints. The transition from the problem of constraint programming to the satisfiability problem of Boolean formulas is described. An algorithm for generating a repair schedule has been developed based on the open-source Python library ORtools, which combines the CP and SAT methods. The necessity of adding a recursive method to the library's basic functionality has been substantiated, allowing for the creation of a schedule without a priori specifying the exact number of repairs.

Keywords: repair scheduling, constraint programming, Boolean formula satisfiability problem, ORtools

Введение. Исправное и работоспособное оборудование способствует бесперебойности производственного процесса, сокращению внеплановых простоев, повышению качества производимой продукции. В связи с этим важной задачей предприятия является сохранение первоначальных свойств оборудования за счет эффективного планирования ремонтных мероприятий. Составление оптимального плана-графика ремонтных работ может быть реализовано с помощью теории расписаний, одним из методов которой является программирование в ограничениях (англ. Constraint Programming, CP).

Для решения задач CP часто используются отдельные программные продукты и библиотеки, написанные для наиболее популярных языков программирования. Однако стандартные инструменты позволяют составлять расписание только с первоначально фиксированным числом ремонтов, что не соответствует постановке задачи циклического планирования. Следовательно, актуальной является задача разработки алгоритма составления расписаний ремонтных работ, который будет дополнять стандартные возможности базовых инструментов.

Целью работы является разработка алгоритма планирования ремонтных работ промышленного оборудования с помощью методов программирования в ограничениях. Для достижения поставленной цели необходима реализация следующих задач: определить понятие программирования в ограничениях, задать основные атрибуты исследования (переменные, их значения и ограничения), разработать алгоритм составления расписания ремонтных работ. Научная новизна исследования заключается в расширении базового функционала возможностью составлять планы-графики без первоначального априорного указания точного числа ремонтных мероприятий.

Материалы и методы; результаты. Под программированием в ограничениях понимают метод программирования, использующий ограничения и их взаимосвязи для решения сложных задач комбинаторной оптимизации [1]. Constraint Programming относится к парадигме декларативного программирования. В отличие от императивной парадигмы, при которой описывается конкретный алгоритм или последовательность действий для решения задачи, в декларативной парадигме программист задает набор ограничений, которые должны быть выполнены, и позволяет системе найти оптимальное решение с учетом этих ограничений [2].

Программирование в ограничениях напрямую связано с теорией удовлетворения ограничений. Задача удовлетворения ограничений (англ. Constraint Satisfaction Problem, CSP) представляет собой набор: переменных $V = \{V_1, \dots, V_n\}$, значений, которые может принимать каждая переменная $D = \{D_1, \dots, D_n\}$, где $x_i \in D_i, i = (1, n)$ и ограничений $C = \{C_1, \dots, C_m\}$, которые накладываются на значения переменных [3, 4].

Процесс решения CSP-задачи заключается в поиске присваивания таких значений переменным, при которых ни одно из установленных ограничений не нарушается. В стандартных библиотеках, например в библиотеке ORtools, написанной для языка программирования Python, для решения CSP-задачи

используются алгоритмы решения задачи выполнимости булевых формул (англ. Satisfiability Problem, SAT), которая может быть определена как задача присвоения переменным логической формулы значений «истина» или «ложь» таким образом, чтобы в конечном итоге формула оказалась истинной.

Использование задачи выполнимости в программных продуктах обусловлено тем, что она работает с булевыми переменными, в то время как в CSP-задачах присутствуют переменные с областью допустимых значений (используются целочисленные переменные). Для решения задачи удовлетворения ограничений, которая с помощью замены переменных приводится к булевым формулам, используются SAT-решатели (англ. SAT-solver) — алгоритмы, решающие задачу выполнимости и включающие в себя преобразование формулы в конъюнктивную нормальную форму.

В библиотеке ORtools используется один из наиболее эффективных алгоритмов — LCG (англ. Lazy Clause Generation), который совмещает в себе концепции Constraint Programming и SAT-solver [5]. При разработке алгоритма планирования расписания ремонтных работ использовался язык программирования Python и SAT-решатель из библиотеки ORtools, основанный на LCG.

Разработанный алгоритм составления расписания ремонтов оборудования, оформленный в соответствии с принципами объектно-ориентированной парадигмы, имплементирован в виде отдельного класса, для которого определено семь методов (см. рисунок). Первый метод — `__init__` конструирует экземпляр класса, который содержит следующие переменные: горизонт планирования (задается пользователем), значение наработки на отказ оборудования (расчетная величина), момент наступления первого ремонта (задается пользователем, не может быть больше значения наработки на отказ), длительность ремонта (определяется из паспорта оборудования), количество ремонтных бригад (задается пользователем).

Второй метод — `set_variables_and_constraints` добавляет в модель целевую функцию, переменные, домены и ограничения. Целевую функцию можно определить как суммарную наработку всех единиц оборудования до начала последнего ремонта в рассматриваемом периоде. В данной задаче целевую функцию необходимо максимизировать.

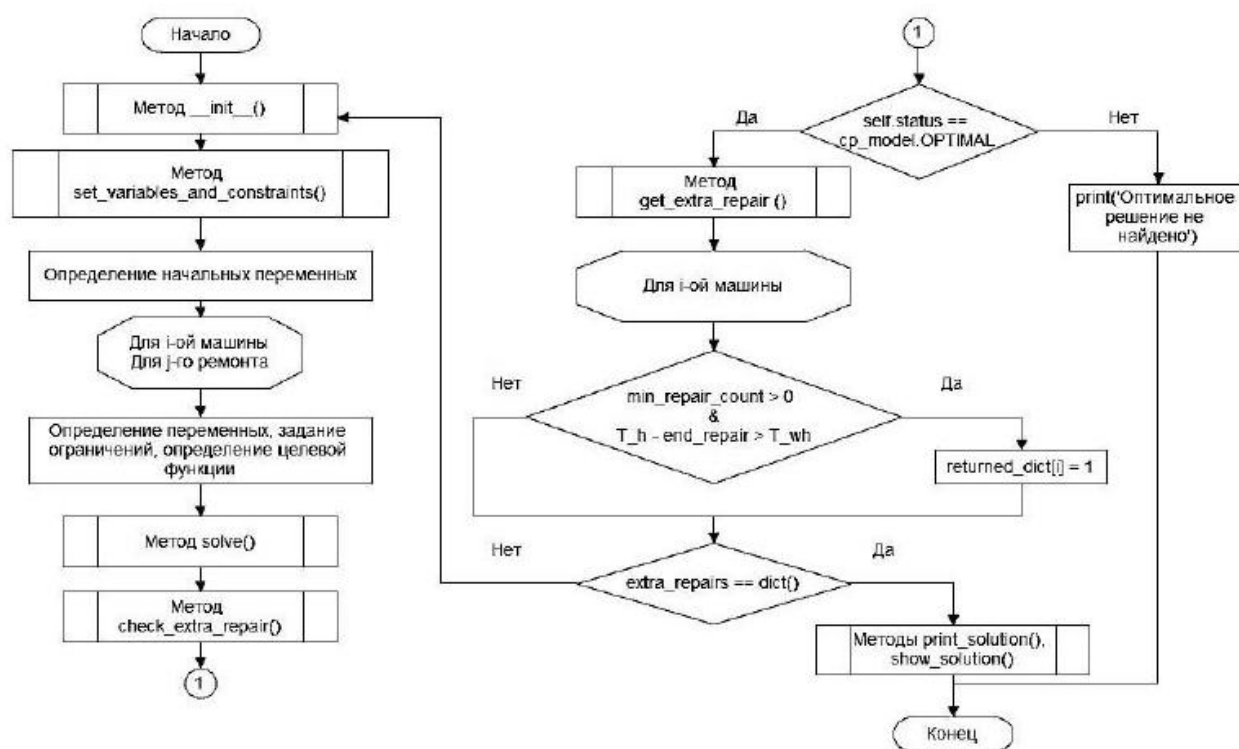
В качестве интервальной переменной используется длительность ремонта. Ограничения можно сформулировать следующим образом: разница между началом следующего ремонта и окончанием предыдущего должна быть больше 0 и меньше значения наработки на отказ.

Третий метод — `solve`, с помощью LCG-алгоритма находит решение задачи. На данном этапе формируется предварительный вариант расписания, которое с помощью методов `check_extra_repair` и `get_extra_repair` необходимо проверить на целесообразность проведения дополнительных ремонтов с помощью двух условий:

– условие ненулевого минимума ремонтов: данное ограничение введено для корректного моделирования сценариев, в которых средняя наработка на отказ превышает заданный пользователем горизонт планирования;

– условие остаточного интервала: длина временного отрезка от момента завершения финального ремонта до конца горизонта планирования не должна превышать статистически ожидаемый период наработки на отказ.

Добавление в расписание дополнительных ремонтов реализовано с помощью рекурсивного подхода: при наличии необходимости в проведении большего числа ремонтных мероприятий производится построение нового экземпляра класса с рассчитанными на предыдущей итерации значениями переменных, и программа заново выполняет все ранее описанные методы. Данное усовершенствование трансформирует алгоритм в адаптивный инструмент, позволяющий формировать расписание без априорного задания конкретного числа ремонтов для каждой единицы оборудования.



Алгоритм составления ремонтных работ

На заключительном этапе работы алгоритма происходит вызов двух методов `print_solution` и `show_solution`, которые выводят оптимальный план-график ремонтных работ в текстовом виде и в виде диаграммы Ганта.

Обсуждение полученных результатов. Разработанный алгоритм, основанный на синтезе программирования в ограничениях и задачи выполнимости булевых формул, позволяет составлять планы-графики без необходимости фиксированного задания точного количества ремонтов, что является ключевым ограничением классических СР-подходов. Таким образом, предложенный инструмент позволяет создавать более автономные и адаптивные системы управления техническим обслуживанием и ремонтом, которые способны работать в условиях неполной входной информации.

Заключение. Предложенный алгоритм может быть использован в комплексных решениях эффективной организации технического обслуживания и ремонта. Для того чтобы планирование ремонтных работ было наиболее точным, целесообразно предусмотреть программный модуль, позволяющий прогнозировать техническое состояние оборудования, на основании которого будут рассчитываться значения наработки на отказ, используемые в качестве одной из входных переменных в разработанном алгоритме.

Список источников

- [1] Щербина О.А. Удовлетворение ограничений и программирование в ограничениях. *Интеллектуальные системы*, 2011, т. 15, № 1–4, с. 53–170.
- [2] Зуенко А.А., Фридман О.В., Зуенко О.Н. Парадигма программирования в ограничениях при решении задач составления расписаний: аналитический обзор. *Вестник ВГУ. Серия: Системный анализ и информационные технологии*, 2022, № 4, с. 156–179.
- [3] Емельянова Д.К. Обзор методов решения задачи удовлетворения ограничений. *Молодой ученый*, 2016, № 1(105), с. 149–153. URL: <https://moluch.ru/archive/105/24883> (дата обращения 15.09.2025).
- [4] Зуенко А.А., Олейник Ю.А. Составление расписаний как задача удовлетворения ограничений (на примере планирования открытых горных работ). *Информатика и автоматизация*, 2024, т. 23, № 5, с. 1290–1310.
- [5] Chu G., Stucke P.J., de la Banda M.G. et al. Symmetries and Lazy Clause Generation. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011, pp. 516–521. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-094>