УДК 519.85

АЛГОРИТМ ПОИСКА ВРЕМЕНИ ЗАВЕРШЕНИЯ СЛОЖНОГО ПРОЕКТА

КОСОЛАП А. И.[1*], *д. ф.-м. н., проф.,*
НЕСТЕРЕНКО А. М.[2*], *асп.*

[1*] Кафедра специализированных компьютерных систем, Государственное высшее учебное заведение «Украинский государственный химико-технологический университет», ул. Набережная Победы, 40, 49005, Днепропетровск, Украина, тел. +38 (0562) 7535726, e-mail: anivkos@ua.fm, ORCID ID: 0000-0001-73386707

[2*] Кафедра специализированных компьютерных систем, Государственное высшее учебное заведение «Украинский государственный химико-технологический университет», ул. Набережная Победы, 40, 49005, Днепропетровск, Украина, тел. +38 (0562) 7535726, e-mail: valhaalmainbox@gmail.com, ORCID ID: 0000-0002-0068-3062

Аннотация. ***Постановка проблемы.*** Рассматривается задача минимизации времени построения сложной системы, которая состоит из множества подсистем. Такие задачи возникают на производстве, в строительстве, управлении, информатике и других областях. Построение системы предполагает последовательное построение ее подсистем, состоящих из множества элементов. Эти элементы в заданные моменты времени поступают на вход системы построения. Известный технологический маршрут каждого элемента и каждой подсистемы, включающей заданное множество элементов, а также время обработки каждого элемента и подсистемы. Построение сложной системы представляется в виде сетевого графика, который определяет последовательность ее построения. Этот сетевой график в данном случае представляет дерево дуг, основой которого является завершение построения сложной системы. При заданных временах поступления элементов системы и их соединения в подсистемы минимальное время построения системы равно критическому пути данного сетевого графика. Для его нахождения, необходимо определить время завершения каждой подсистемы. Это время зависит от времени готовности всех элементов подсистемы для ее сборки. Время критического пути может быть уменьшено посредством устранения ожиданий на участках сборки подсистем. Минимизация ожиданий позволяет получить критический путь минимальной длины. Существуют такие моменты времени поступления элементов сложной системы на входе сетевого графика, для которых суммарное ожидание элементов и их подсистем на каждом пункте сборки будет минимальным. Определение таких моментов поступления на вход системы необходимо начинать с конца сетевого графика. Уменьшение времени завершения сборки системы влечет за собой уменьшение времени поступления составляющих ее элементов. Это уменьшение влечет за собой уменьшение времени поступления элементов подсистемы на предшествующий пункт сборки. Такое прохождение сетевого графика от конца к началу с изменением времен поступления элементов подсистем позволяет минимизировать ожидания начала сборки подсистем на каждом участке. Это достигается посредством определения оптимальных времен поступления элементов на входе сетевого графика. После проделанных изменений критический путь необходимо пересчитать и только в том случае, если его значение не будет изменено, будет найдено минимальное время построения сложной системы. Для реализации данного алгоритма вычисления минимального времени построения сложной системы разработана компьютерная программа и проведены численные эксперименты, которые подтверждают эффективность данного алгоритма.

Ключевые слова*: сложная система, сетевой график, критический путь, алгоритм минимизации времени*

АЛГОРИТМ ПОШУКУ ЧАСУ ЗАВЕРШЕННЯ СКЛАДНОГО ПРОЕКТУ

КОСОЛАП А. І.[1*], *д. ф.-м. н., проф.,*
НЕСТЕРЕНКО А. Н.[2*], *асп.*

[1*] Кафедра спеціалізованих комп’ютерних систем, Державний вищий навчальний заклад «Український державний хіміко-технологічний університет», вул. Набережна Перемоги, 40, 49005, Дніпропетровськ, Україна, тел. +38 (0562) 7535726, e-mail: anivkos@ua.fm, ORCID ID: 0000-0001-73386707.

[2*] Кафедра спеціалізованих комп’ютерних систем, Державний вищий навчальний заклад «Український державний хіміко-технологічний університет», вул. Набережна Перемоги, 40, 49005, Дніпропетровськ, Україна, тел. +38 (0562) 7535726, e-mail: valhaalmainbox@gmail.com, ORCID ID: 0000-0002-0068-3062.

Анотація. ***Постановка проблеми***. Розглядається задача мінімізації часу побудови складної системи, яка складається з множини підсистем. Такі задачі виникають на виробництві, в будівництві, управлінні, інформатиці та інших прикладних галузях. Побудова системи передбачає послідовну побудову її підсистем, що складаються з безлічі елементів. Ці елементи в задані моменти часу надходять на вхід системи. Відомий технологічний маршрут кожного елемента і кожної підсистеми, що включає задану кількість елементів, а також час обробки кожного елемента і підсистеми. Побудова складної системи подається у вигляді мережевого графіка, який визначає послідовність її побудови. Цей мережевий графік у даному випадку являє собою дерево дуг, основою якого є час завершення побудови складної системи. При заданих моментах надходження елементів системи та їх складання в підсистеми, мінімальний час побудови системи дорівнює критичному шляху даного мережевого графіка. Для його знаходження необхідно визначити час завершення кожної підсистеми. Цей час залежить від часу готовності всіх елементів підсистеми для її складання. Час критичного шляху може бути зменшений за допомогою усунення очікувань на ділянках складання підсистем. Мінімізація очікувань дозволяє отримати критичний шлях мінімальної довжини. Існують такі моменти часу надходження елементів складної системи на вході мережевого графіка, для яких сумарне очікування елементів та їх підсистем на кожному пункті складання буде мінімальним. Визначення таких моментів надходження на вхід системи необхідно починати з кінця мережевого графіка. Зменшення часу завершення складання системи тягне за собою зменшення часу надходження складових її елементів. Це зменшення тягне за собою зменшення часу надходження елементів підсистеми на попередній пункт складання. Таке проходження мережевого графіка від кінця до початку зі зміною часів надходження елементів підсистем дозволяє мінімізувати очікування початку складання підсистем на кожній ділянці. Це досягається за допомогою визначення оптимальних часів надходження елементів на вході мережевого графіка. Після виконаних змін критичний шлях необхідно перерахувати і тільки в тому випадку, якщо його значення не буде змінено, буде знайдене мінімальний час побудови складної системи. Для реалізації даного алгоритму обчислення мінімального часу побудови складної системи розроблено комп'ютерну програму та проведено числові експерименти, які підтверджують ефективність даного алгоритму.

Ключові слова: *складна система, мережевий графік, критичний шлях, алгоритм мінімізації часу*

ALGORITHM FOR SEARCH OF END TIME OF THE COMPLEX PROJECT

KOSOLAP A. I.[1*] , *Dr. Sc. (Phys.-Math.), Prof.*
NESTERENKO A. N.[2*] , *PhD student*
[1*] Department of specialized computer systems, State Higher Education Establishment «Ukrainian State University of Chemical Technology», 40, Naberezhna Peremogy str., Dnipropetrovsk 49005, Ukraine, tel. +38 (0562) 7535726, e-mail: anivkos@ua.fm, ORCID ID: 0000-0001-73386707.
[2*] Department of specialized computer systems, State Higher Education Establishment «Ukrainian State University of Chemical Technology», 40, Naberezhna Peremogy str., Dnipropetrovsk 49005, Ukraine, tel. +38 (0562) 7535726, e-mail: valhaalmainbox@gmail.com, ORCID ID: 0000-0002-0068-3062.

Summary. ***Problem statement.*** The article discusses the problem of minimizing the construction time of a complex system, which consists of the set of subsystems. Such problems arise in manufacturing, construction, management, computer science and other fields. Building a system requires consistent construction of its subsystems, which include numerous elements. These elements are supplied to the system’s entrance at specific moments of time. The technological route of each element and each subsystem consisting of a given number of elements, as well as the processing time of each element and subsystem are known. Construction of a complex system is represented as a network graph, which determines its construction sequence. The network graph in this case is a tree of arcs, the root of which is the completion of a complex system’s construction. With a given times of system elements supply and of their compounding into subsystem, the minimum time of system’s construction is equal to the critical path of network graph. To find the critical path it is necessary to determine the completion time of each subsystem. This time depends on availability time of all elements of subsystems required for its assembly. Time critical path can be reduced by removing the idling in the areas of subsystems assembly. Minimization of idling periods provides a critical path of minimum length. There are such timings of complex system element’s arrival at the entrance of network graph for which the total idle periods of elements and subsystems at each point of the assembly will be minimal. Determining such moments of arriving at system’s entrance requires starting from the end of the network graph. Reducing the completion time of system’s assembly entails a reduction in the time of its elements arrival. This reduction leads to a decrease in the arrival time of the subsystem’s elements on the previous point of assembly. Such passage from end to beginning of the network graph, changing times of subsystems’ elements arrival allows minimizing idle periods before the assembly of sub-systems at each region. This is achieved by determining the optimum arrive timings of elements at the entrance of network graph. After the performed changes the critical path must be recalculated and only if its value is not changed, the minimum construction time of complex system will be found. To implement the computing algorithm of minimization of a complex system’s construction time, a computer program is developed and the numerical experiments that confirm the effectiveness of the algorithm are performed.

Keywords: *complex system, a network graph, critical path, algorithm to minimize time*

Постановка проблемы. С каждым годом строится все больше сложных систем в разных областях человеческой деятельности. Это сложные машины и оборудование, строительные объекты, системы управления, информационные системы и системы искусственного интеллекта. Такие системы часто состоят из сотен тысяч и даже миллионов элементов. Комплекс задач, которые необходимо решать при их построении, уже невозможен без соответствующего информационного обеспечения и использования вычислительной техники [4, 6, 12].

Эффективное решение задач построения сложных систем включает их математическое моделирование. Такие задачи относят к классу календарного планирования [5; 7; 10]. Задачи этого класса являются достаточно сложными и только в простых случаях существуют эффективные алгоритмы. Классическая задача обработки n деталей (элементов) на m станках (участках) эффективно решена только для двух станков (алгоритм Джонсона) [9]. В этой задаче теории расписаний не предполагается сборка деталей. Однако допускается, что на одном станке может последовательно обрабатываться несколько деталей, а деталь может одновременно обрабатываться только на одном станке. Это порождает очереди к станкам, что усложняет ее решение. Число допустимых решений в задаче теории расписаний равно $n!$, поэтому алгоритмы полного перебора вариантов для этой задачи неэффективны. Используются различные схемы сокращения перебора вариантов, в частности, методы ветвей и границ и другие, в том числе стохастические методы [8; 11; 13]. Однако сложность этих методов растет экспоненциально при увеличении размерности задачи. Существуют целочисленные модели задач теории расписаний [1], но и для них не разработано эффективных алгоритмов. В работе [2] предложены кусочно-линейные модели с непрерывными переменными:

$$\min\{T \mid x_{ij} + t_{ij} \le T, i = 1,...,n, j = 1,....,m,$$

$$\sum_{i=1}^{n} g_{ij}(t - x_{ij}) \le 1, j = 1,...,m, \qquad (1)$$

$$\sum_{j=1}^{m} g_{ij}(t - x_{ij}) \le 1, i = 1,...,n, x_{ij} \ge 0, \forall ij, t \ge 0\},$$

где t_{ij} – время обработки i-й детали на j-м станке, а x_{ij} – искомое время начала обработки i-й детали на j-м станке. Функции g_{ij} определяются следующим образом:

$$g_{ij}(t - x_{ij}) = \frac{1}{2}(| t - x_{ij} | / t_{ij} + (| t - x_{ij} - x_{ij} -$$
$$- \min t_{ij} | - | t - x_{ij} - t_{ij} |)(1/t_{ij} + 1/\min t_{ij}) +$$
$$+ (t - x_{ij} - t_{ij} - \min t_{ij})/t_{ij}).$$

Задача (1) содержит континуум ограничений, но, учитывая характер функций, достаточно рассмотреть ограничения только для значений $t = x_{ij} + t_{ij}$. Таким образом, задача (1) будет иметь nm+1 непрерывную переменную и $nm(1 + n + m)$ ограничений.

Для решения задачи (1) можно использовать метод точной квадратичной регуляризации [3] в комбинации с методом последовательного раскрытия модулей [2]. Однако численная эффективность этих методов для решения задач (1) еще не проводилась.

В данной работе предполагается обработка деталей и их сборка. На каждом участке (станке) возможна сборка только одной подсистемы. Это условие не является сильно ограничивающим, так как участок можно разбить на части так, что каждая часть будет обрабатывать или собирать только одну подсистему.

Постановка задачи. Имеется n элементов (деталей), которые в заданные моменты времени $t_i, i = 1,...,n$ поступают на вход цеха, где их необходимо обработать и собрать в заданную сложную систему за минимальное время. Обработка и сборка элементов системы производится на участках цеха, причем каждый из m участков цеха может собирать (обрабатывать) только одну подсистему. Известно время сборки T_j i-й подсистемы ($j = 1,…,m$), а также технологический маршрут для элементов и подсистем, который задается сетевым графиком. Пример сетевого графика представлен на рисунке 1.

Алгоритм решения задачи. Исходными данными для решения задачи являются значения параметров $t_i = 0, i = 1,...,n$ и $T_j, j = 1,...,m$ и сетевой график технологических маршрутов. Для каждой вершины сети (участка) определим время завершения обработки детали (подсистемы) $Z_j, j=1,...,m$ по формуле

$$Z_j = \max Z_k + T_j,$$

где индекс k пробегает все входящие в вершину j дуги $k \in S(j)$. Тогда $Z_m - \min t_i$ будет равно критическому пути в данном сетевом графике.

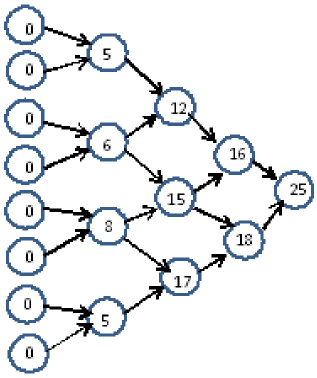

Рис. 1. Пример сетевого графика системы

На следующем шаге рассмотрим вершину m и $Z_j, j \in S(m)$. Найдем

$$Z_r = \min\{Z_j \mid j \in S(m)\}$$

и положим

$$Z_r = T_r + \min\{Z_j \mid j \in S(r)\} - \max\{Z_j \mid j \in S(r)\}$$

а для всех остальных вершин $Z_j = Z_r j \in S(m)$. Далее, изменим значения $Z_i, j \in S(j)$ так, чтобы расчетные значения $Z_j, j \in S(m)$ совпали с заданными. Продолжим этот процесс изменения значений Z_i до достижения начальных вершин сети. После этого пересчитаем критический путь измененного сетевого графика. Этот процесс изменений Z_i будем повторять до тех пор, пока значение критического пути не будет меняться. Если все $t_i = 0, i=1,...,n$ и $T_j, j=1,...,m$ целые, то через конечное число итераций будет найден минимальный критический путь в данном сетевом графике $T = Z_m - \max t_i$. Таким образом, будет определено минимальное время построения сложной системы.

Пример. Рассмотрим сетевой график системы, представленный на рисунке 2.

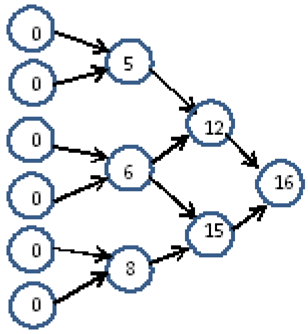

Рис. 2. Сетевой график системы

Для этого сетевого графика первоначальное значение критического пути равно 39. Преобразованный сетевой график по рассмотренному алгоритму имеет вид (рис. 3)

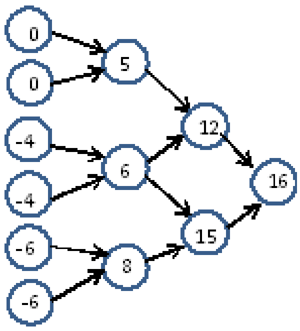

Рис. 3. Минимальный критический путь

Критический путь для преобразованного сетевого графика будет равен 17 + 16 = 33. Данный алгоритм позволяет построить сетевой график таким образом, чтобы завершить построение системы к указанному сроку.

Выводы. В данной работе для класса задач оптимального календарного планирования построения сложной системы приведена математическая модель и разработан эффективный алгоритм для расчета минимального критического пути. Этот алгоритм может быть использован при построении сложных систем большой размерности.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Корбут А. А. Дискретное программирование / А. А. Корбут, Ю. Ю. Финкельштейн ; под ред. Д. Б. Юдина. – Москва : Наука, 1969. – 368 с.
2. Косолап А. И. Методы глобальной оптимизации / А. И. Косолап. – Днепропетровск : Наука и образование, 2013. – 316 с.

3. Косолап А. И. Глобальная оптимизация. Метод точной квадратичной регуляризации / А. И. Косолап. – Днепропетровск : ПГАСА, 2015. – 164 с.
4. Кузин Б. И. Оптимальное календарное планирование на поточных линиях и предметных участках / Б. И. Кузин. – Ленинград : ЛГУ, 1969. – 167 с.
5. Теория расписаний и вычислительные машины : сб. ст. / под. ред. Э. Г. Коффмана ; пер.с англ. В. М. Амочкина, под ред. Б. А. Головкина. – Москва : Наука, 1984. – 336 с.
6. Baker K. R. Principles of sequencing and scheduling / K. R. Baker, D. Trietsch. – Hoboken ; New Jersey : A John Wiley & Sons, 2009. – 510 p.
7. Brucker P. Scheduling algorithms / P. Brucker. – 5th ed. – Berlin ; Heidelberg ; New York : Springer, 2007. – 377 p.
8. Horst R. Global optimization. Deterministic approaches / R. Horst, H. Tuy. – 3rd ed. – Berlin ; Heidelberg ; New York : Springer, 1996. – 727 p.
9. Johnson S. M. Optimal two and three stage production schedules with setup times included / S. M. Johnson // Naval research logistics quarterly. – 1954. – Vol. 1. – P. 61-68.
10. Pinedo M. L. Scheduling : theory, algorithms, and systems; fourth edition / M. L. Pinedo. – 4th ed. – Berlin ; Heidelberg ; New York : Springer, 2012. – 696 p.
11. Scholz D. Deterministic global optimization. Geometric branch-and-bound methods and their applications / D. Scholz. – Berlin ; Heidelberg ; New York : Springer, 2010. – 153 p.
12. Uher T. Programming and scheduling techniques / Thomas Uher, Adam S. Zantis. – 2nd ed. – Kensington : NewSouth Publishing, 2011. – 285 p.
13. A review of methods and algorithms for optimizing construction scheduling / J. Zhou, P. E. D. Love, X. Wang, K. L. Teo, Z. Irani // Journal of the Operational Research Society. – 2013. – Vol. 64. – P. 1091-1105.

REFFERENCES

1. Korbut A.A. and Finkelshteyn Yu.Yu. *Diskretnoe programmirovanie* [Discrete programming]. Moskow: Nauka, 1969, 368 p. (in Russian).
2. Kosolap A.I. *Metody global'noy optimizatsii* [Methods of global optimization]. Dnepropetrovsk: Nauka i obrazovanie, 2013, 316 p. (in Russian).
3. Kosolap A.I. *Global'naya optimizatsiya. Metod tochnoy kvadratichnoy regulyarizatsii* [Globaloptimisation. A method of exact quadraticregularization]. Dnepropetrovsk: PGASA, 2015, 164 p. (in Russian).
4. Kuzin B.I. *Optimal'noe kalendarnoe planirovanie na potochnykh liniyakh i predmetnykh uchastkakh* [Optimal scheduling on production lines and subject areas]. Leningrad: LGU, 1969, 167 p. (in Russian).
5. Koffman E.G. ed. *Teoriya raspisaniy i vychislitel'nye mashiny* [Scheduling theory and computers]. Moskow: Nauka, 1984, 336 p. (in Russian).
6. Baker K.R. and Trietsch D. *Principles ofsequencing andscheduling.* Hoboken, New Jersey: A John Wiley& Sons, 2009, 510 p.
7. Brucker P. *Scheduling algorithms*. 5th ed. Berlin, Heidelberg, New York: Springer, 2007, 377 p.
8. Horst R. and Tuy H. *Global Optimization. Deterministic Approaches*. 3rd ed. Berlin, Heidelberg, New York: Springer, 1996, 727 p.
9. Johnson S.M. *Optimal two and three stage production schedules with setup times included. Naval research logistics quarterly,* 1954, vol. 1, pp. 61-68.
10. Pinedo M.L. *Scheduling : theory, algorithms, and systems; fourth edition*. 4th ed. Berlin, Heidelberg, New York: Springer, 2012, 696 p.
11. Scholz D. *Deterministic globaloptimization. Geometric branch-and-bound methods and their applications*. Berlin, Heidelberg, New York: Springer, 2010, 153 p.
12. Uher T. and Adam S. Zantis. *Programmingand schedulingtechniques*. 2nd ed. Kensington: NewSouth Publishing, 2011, 285 p.
13. Zhou J., Love P.E.D., Wang X., Teo K.L., Irani Z. *A review of methods and algorithms for optimizing construction scheduling. Journal of the Operational Research Society.* 2013, vol. 64, pp. 1091-1105.

Рецензент: д-р т. н., проф. Н. М. Ершова
Надійшла до редколегії: 23.09.2015 р. Прийнята до друку: 29.09.2015 р.