УДК 004.023

РАЗРАБОТКА КОМПЬЮТЕРИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ ПОДСИСТЕМЫ ОПЕРАТИВНО-КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С ПОМОЩЬЮ МУРАВЬИНЫХ АЛГОРИТМОВ

Черный М.С., Скобцов Ю.А.
Донецкий национальный технический университет,
кафедра автоматизированных систем управления
E-mail: chernyy@mls-automatization.com, ya-skobtsov@list.ru

В статье рассматривается вопрос использования муравьиных алгоритмов для оперативно-календарного планирования на машиностроительном предприятии.

5

Общая постановка проблемы

Качество функционирования современного производства во многом определяется решениями, принимаемыми на этапах календарного планирования и оперативного управления. Планирование обеспечивает взаимодействие совокупности элементов управляемого объекта для достижения заданных целей, главная из которых заключается в организации согласованного во времени и маршрутно-ориентированном пространстве движения частей и изделий в производстве. В рамках оперативного управления одной из важнейших проблем является проблема планирования загрузки оборудования, т.е. упорядочение работ на выбранной структуре производственных модулей (календарное планирование) [1]. Особенностью машиностроительной сферы является тот факт, что в процессе работы некоторое оборудование может выйти из строя, вследствие этого необходимо иметь возможность динамически перераспределять нагрузки между оставшимся оборудованием, чтобы не останавливать все производство.

Постановка задачи исследования

Проблема оперативно-календарного планирования (JSP) может быть характеризована как n деталей, которые должны быть обработаны на m станках, множество n деталей можно описать как $J = \{J_1,...,J_n\}$, тогда как множество m станков $M = \{M_1,...,M_m\}$. Каждая деталь представляет собой план технологического процесса, который определяет ограничения предшествования. Обработку детали на одном станке называют операцией, а обработку детали i на станке j обозначают u_{ij}. Так что множество операций можно описать как $O = \{u_{ij} \mid i \in [1,n], j \in [1,m]\}$, где n обозначает число деталей, а m число станков. Как только операция началась, она не может быть остановлена или выполнятся одновременно с другой. Это означает, что операция u_{ij} не может начаться, пока не закончится u_{ij-1}.

Каждая операция $u_{ij} \in O$ должна иметь свое время выполнения (время обработки i детали на j станке) $p_{ij}(p_{ij} > 0)$, так же известное как ограничение машинной обработки. Значение $C_{ij} = C_{ij} + p_{ij}$ – это время завершения операции u_{ij} в последовательности операций $u_{ik} \to u_{ij}$. p_{ij} заранее установлено, так что проблема заключается в определении времени завершения $C_{ij}(\forall u_{ij} \in O)$, что минимизирует данное соотношение:

$$C_{max} = \max_{u_{ij} \in O}(C_{ij}) = \max_{u_{ik} \to u_{ij}} (C_{ik} + p_{ij}) \quad (1)$$

В JSP существует 2 ограничения: ограничение предшествования и ограничение машинной обработки.

1. Ограничение предшествование означает, что для каждой детали есть фиксированный технологический процесс, при этом выполнение операции не может быть прервано или выполнятся одновременно с другой.

$$\neg\exists k(u_{ip} \to u_{ik} \vee u_{ik} \to u_{iq}), \text{ где } u_{ip} \to u_{iq}, k \neq p \wedge k \neq q \quad (2)$$

Учитывая задержки в работе, такие как время ожидания станка во время выполнения операции, мы получаем:

$$C_{ij} \geq C_{ik} + p_{ij}, \text{ где } u_{ik} \to u_{ij} \quad (3)$$

2. Ограничение машинной обработки означает, что только

5

одна деталь может обрабатываться в текущее время на данном станке:

$$C_{ij} \geq C_{kj} + p_{ij}, \text{ где } u_{kj} \rightarrow u_{ij}$$

Операции должны быть разнесены во времени таким образом, чтобы как только одна операция началась, она должна быть закончена.

Представим нашу задачу с помощью дизъюнктивного графа $D = (N, A, B)$. N – набор узлов, соответствующий всем операциям. A – множество коньюктивных направленных дуг, основанных на правилах предшествования, и B – множество дизъюнктивный ненаправленных дуг, соединяющие операции различных деталей, которые должны быть выполнены на данном станке. При применении данного графа к проблеме JSP, N, A и B можно описать так:

1) $N = O \cup \{u_0\} \cup \{u_{N+1}\}$, $\{u_0\}$ и $\{u_{N+1}\}$ – специальные фиктивные узлы, которые определяют начало и окончание всех списков операций для каждой детали. N – общее число всех действительных операций, где N=n*m.

2) $A = \{(u_{ij}, u_{ij+1}) \mid u_{ij} \rightarrow u_{ij+1}$ последовательность операций для детали $J_i, 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{(u_0, u_{i1})$: u_{i1} – первая операция для детали $J_i, 1 \leq i \leq n\} \cup \{(u_{im}, u_{N+1})$: u_{im} – последняя операция для детали $J_i, 1 \leq i \leq n\}$.

3) $B = \{(u_{ij}, u_{hj}) \mid 1 \leq i \leq n, 1 \leq j \leq m, h \neq i\}$.

Вес ребер ассоциируется со временем обработки p_{ij} для каждой операции $u_{ij} \in O(p_0 = p_{N+1} = 0)$. Длина пути определяется как сумма весов для всех операций в этом пути, начиная с начальной точки и до точки завершения обработки детали.

Для описания технологического процесса используем матрицу T, а для описания длительности операций возьмем матрицу P. Каждая строка T показывает порядок обработки на станках, на которых должны выполняться все операции данной детали, тогда как эта же строка матрицы P показывает время выполнения данных операций.

Приведем пример для 2-х деталей и 3-х станков.

$$T = \begin{bmatrix} M_1 & M_2 & M_3 \\ M_3 & M_1 & M_2 \end{bmatrix}$$

$$P = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \end{bmatrix}$$

На основе данных матриц можно построить граф.

На данном графе узлы 1-6 представляют собой матрицу T, каждому узлу соответствует операция. Направленные связи между узлами 0-1-2-3-7 и 0-4-5-6-7 показывают ограничения предшествования, описанные в матрице T, а двунаправленные связи между остальными операциями, показывают, что между ними нет ограничения предшествования. Узлы 0 и 7, являющиеся входом и выходом соответственно, являются фиктивными узлами.

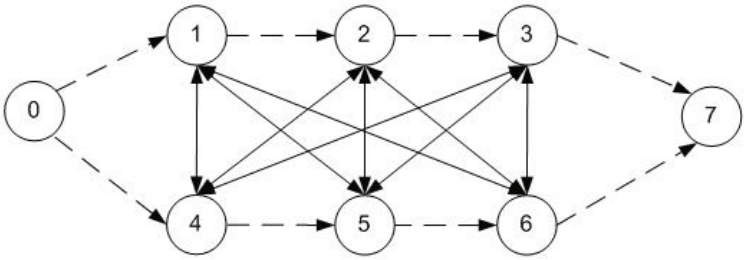

Рисунок 1 – Пример графа

Муравьиные алгоритмы применительно к задаче JSP

Строго говоря, в начальный момент времени концентрация феромона для каждой дуги графа нулевая, но для удобства каждой дуге присваивается небольшое случайное число $\tau_{ij}(0)$.

Муравей выбирает следующую дугу пути следующим образом. Множество муравьев k=1,…,n_k помещаются в начальную вершину. В каждой итерации МА каждый муравей пошагово строит путь до конечной вершины. При этом в каждой вершине каждый муравей должен выбрать следующую дугу пути. Если муравей k находится в вершине i, он выбирает следующую вершину $j \in N_i^k$ на основе вероятностей перехода

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{u\in N_i^K(t)}\tau_{iu}^{\alpha}\eta_{iu}^{\beta}(t)} & if\ \ j\in N_i^k\\ 0 & if\ \ j\notin N_i^k\end{cases} \qquad (4)$$

N_i^k – представляет множество возможных вершин, связанных с вершиной i, для муравья k. Если для любого узла i и муравья k $N_i^k \equiv \varnothing$, тогда предшественник узла i включается в N_i^k. Для каждого муравья создается и отслеживается табу-список. Вершины из этого списка удаляются из N_i^k , поскольку каждая вершина может посещаться только один раз;

τ_{ij} – представляет эффективность перехода из вершины i в j, которая определяется интенсивностью феромона для соответствующей дуги;

η_{ij} - представляет эффективность перехода из i в j на основе некоторой эвристики. В качестве эвристики используем кратчайшее расстояние:

$$\eta_{ij}=\frac{1}{d_{ij}} \qquad (5)$$

d_{ij} – расстояние между вершинами i и j. Очевидно, что в этом случае предпочтительней короткая дуга, исходящая из вершины i;

α – положительная константа, которая определяет влияние концентрации феромона;

β – положительная константа, которая определяет влияние эвристики.

Когда все муравьи построили полный путь, каждый муравей помечает свой построенный путь, откладывая для каждой дуги феромон.

Обновление феромона проходит в 2 этапа:

– локальное обновление;

– глобальное обновление.

1) Локальное обновление. Коррекция концентрации феромона осуществляется в соответствии со следующим правилом:

$$\tau_{ij}(t) = (1 - p_1)\tau_{ij} + p_1\tau_0 \quad (6)$$

p_1 - коэффициент представляющий испарение феромона (0< p_1<1);

τ_0 – малая положительная константа.

Эксперименты при решении задачи коммивояжера показали, что значение $\tau_0 = (n_G L)^{-1}$ дает хорошие результаты, где n_G – число узлов в графе и L – длина тура.

2) Глобальное обновление. Коррекцию феромона разрешается проводить только лучшим (в глобальном смысле) муравьям, которые построили кратчайший путь $x^+(t)$ (лучший путь, найденный с первой по текущую итерацию), в соответствии со следующим правилом:

$$\tau_{ij}(t+1) = (1 - p_2)\tau_{ij}(t) + p_2\Delta\tau_{ij}(t) \quad (7)$$

$$\Delta\tau_{ij}(t) = \begin{cases} \dfrac{1}{f(x^+(t))} & if \quad (i,j) \in x^+(t) \\ 0 \end{cases} \quad (8)$$

p_2 - коэффициент представляющий испарение феромона (0< p_2 <1).

Эта стратегия отдает предпочтение эксплуатации пространства поиска и применяется после того, как решение построено. Для более быстрого поиска значение p_2 изменяться в процессе поиска решения: на начальной стадии используются большие значения, а на конечной – малые.

Проиллюстрируем данный алгоритм с помощью блок-схемы, представленной на рис. 2.

Для муравья k множество N_i^k формируется динамически в соответствии с ограничением предшествования операций и состояния станков (станок занят или свободен). В начале каждой итерации N_1^k инициализируется первой операцией для детали $N_1^k = \{u_{i1} \mid i \in [1,n]\}$, где n – число деталей. Когда муравей k переходит из вершины u_{ip} в u_{iq}, u_{ip} удаляется, а u_{iq} добавляется

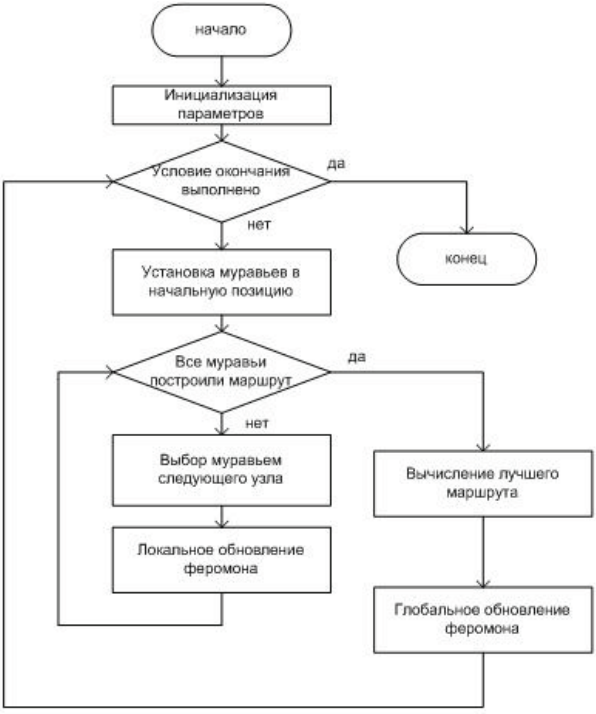

Рисунок 2 – Обобщенная блок-схема алгоритма решения задачи

в N_i^k, если станок для выполнения операции u_{iq} свободен. Все следующие операции остальных деталей, прошлые операции которых закончились, а новые не начались, также включаются в N_i^k.

В алгоритме могут быть использованы различные критерии окончания:

1. Окончание при превышении заданного числа итераций;
2. Окончание по найденному приемлемому решению $f(x^k(t)) \le \varepsilon$;
3. Окончание, когда на протяжении нескольких итераций не было отмечено повторных изменений;
4. Окончание, когда все муравьи следуют одним и тем же путем.

Решение задачи и результаты исследований

Для обеспечения высокой эффективности работы производственных участков необходимо составлять близкие к оптимальному расписания работы оборудования. Поэтому необходимо разработать компьютеризированную подсистему оперативно-календарного планирования работы производственного участка машиностроительного производства для повышения эффективности работы за счет составления субоптимальных расписаний работы, на основе выбранных критериев оптимизации.

В результате исследований был разработан алгоритм решения задачи, выбраны эвристики для определения ключевых параметров.

Выводы

В свете быстрых темпов развития современного производства и высоких требованиях к гибкости и способности адаптироваться к изменяющимся условиям оперативно-календарное планирование является неотъемлемой частью любого производства. Математический аппарат муравьиных алгоритмов способен удовлетворить все требования, предъявляемые современным производством, вследствие того, что он реализует поведение реальных муравьев, чье способность к адаптации к любым условиям столетиями оттачивала природа. Благодаря своей простоте, масштабируемости и динамичности муравьиные алгоритмы целесообразно использовать для решения задачи оперативно-календарного планирования.

Литература

[1] Система оперативно-календарного планирования автоматизированного механообрабатывающего мелкосерийного производства на основе комплексных моделей : диссертация д-ра техн. наук Загидуллина Равиля Рустэмбековича: 05.13.06 Уфа, 2006 448 с. РГБ

ОД, 71:07-5/452.

[2] Implementation of an Ant Colony Optimization technique for job shop scheduling problem, Jun Zhang, Xiaomin Hu, X. Tan, J.H. Zhong and Q. Huang, 2006.

[3] Applying Ant Colony Optimization (ACO) algorithm to dynamic job shop scheduling problems, Rong Zhou, Heow Pueh Lee and Andrew Y.C. Nee, 2008.

[4] Swarm Intelligence. Focus on Ant and Particle Swarm Optimization, 2007 I-Tech Education and Publishing.

[5] Роевые алгоритмы, курс лекций, Скобцов Ю.А, 2009 г.

[6] M. Dorigo and G. Di Caro. The Ant Colony Optimization Meta-Heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 11-32. McGraw-Hill, 1999.

[7] Donald W. Fogarty, Yohn H. Blackstone, Yr. And Thomas R. Hoffman. Production and Inventory management (Cincinnati: South - Western Publishing, 1991). P. 452-453.