

УДК 658.512.6:004.021

В.А. Светличная, Н.В. Червинская, Д.А. Хаустова
Донецкий национальный технический университет, г.Донецк,
кафедра автоматизированных систем управления
E-mail: sva@kita.donntu.edu.ua

ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ МУРАВЬИНЫХ КОЛОНИЙ ПРИ РЕШЕНИИ ЗАДАЧ ОПТИМИЗАЦИИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ

Abstract

Svetlichnaya V.A., Chervinskaya N.V., Khaustova D.A. *Usage of ant colony algorithms for scheduling optimization problem solving. In the article the problem of ant algorithms usage for construction work execution calendar planning with use of limits on renewable resources is considered. All works are grouped for ant algorithm application, the problem solving involves several stages, first for components and then for aggregative works. The use of this method made it possible to reduce planned work execution generic cycle significantly.*

Keywords: *ant algorithm, calendar planning, optimum plan, renewable resources, algorithm of the control centralized traffic.*

Анотація

Світлична В.А., Червінська Н.В., Хаустова Д.О. *Використання алгоритмів мурашиних колоній при вирішенні задач оптимізації календарного планування. У статті розглянуто питання використання мурашиних алгоритмів для складання календарного плану виконання будівничих робіт з використанням обмежень на відновлювані ресурси. Для використання мурашиного алгоритму всі роботи поділені на групи, і задача вирішується за декілька етапів, спочатку для складових, а потім для укрупнених робіт. Використання даного методу дозволило значно зменшити загальний цикл виконання планованих робіт.*

Ключові слова: *мурашиний алгоритм, календарне планування, оптимальний план, відновлювані ресурси, алгоритм диспетчеризації.*

Аннотация

Светличная В.А., Червинская Н.В., Хаустова Д.А. *Использование алгоритмов муравьиных колоний при решении задач оптимизации календарного планирования. В статье рассматривается вопрос использования муравьиных алгоритмов для составления календарного плана выполнения строительных работ с использованием ограничений на возобновимые ресурсы. Для использования муравьиного алгоритма все работы разбиты на группы, и задача решается в несколько этапов, сначала для составляющих, а потом для укрупненных работ. Использование данного метода позволило значительно уменьшить общий цикл выполнения планируемых работ.*

Ключевые слова: *муравьиный алгоритм, календарное планирование, оптимальный план, возобновимые ресурсы, алгоритм диспетчеризации.*

Общая постановка проблемы. Многие отрасли современной промышленности требуют использования четкого планирования последовательности выполнения работ, при котором должен быть не только составлен график, отражающий начало и конец (продолжительность) каждой работы, но и учитывающий задействованные ресурсы. Одной из таких отраслей является строительство, для которого существуют свои особенности организации деятельности. К ним следует отнести следующие аспекты [1].

Планирование затрагивает долгий инвестиционный период, а чем дольше период планирования, тем выше неопределенность и тем ниже точность плана.

Строительная деятельность требует жесткой регламентации работ.

При планировании строительной деятельности предприятия должны быть задействованы методы календарного планирования, т.е. использованы составление сетевого графика работ и календарное планирование ресурсов.

Большой объем выполняемых работ при строительстве современных объектов, высокие требования к срокам, динамично изменяющиеся финансовые параметры требуют все более высокого качества при планировании ведения строительных работ, а следовательно использования новых методов, которые дают более эффективные результаты.

Постановка задач исследования. Задача составления оптимального графика выполнения работ является достаточно известной, и для ее решения были использованы разные математические методы. В настоящее время разработаны интеллектуальные методы поиска оптимального решения и современные компьютерные технологии позволяют исследовать возможность применения этих методов в конкретных условиях. В данной работе необходимо было исследовать возможность применения таких методов, для конкретного объекта, определить условия, в которых выбранный метод можно использовать, проанализировать результаты.

Целью создания алгоритма должно быть получение оптимального плана выполнения работ, т.е. синхронизация работы в пределах номенклатуры с учетом наложенных технологических ограничений при максимальной загрузке оборудования и максимальном использовании ресурсов.

Решение задачи и результаты исследований. Для решения поставленной задачи был проведен анализ с одной стороны всех этапов планирования при строительстве современных объектов, а с другой стороны, математических методов, которые возможно было бы при этом использовать. В результате был выбран метод муравьиных алгоритмов.

Идея муравьиного алгоритма — моделирование поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь. При своём движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьев находить новый путь, если старый оказывается недоступным. Очевидная положительная обратная связь быстро приведёт к тому, что кратчайший путь станет единственным маршрутом движения большинства муравьев. Моделирование испарения феромона — отрицательной обратной связи — гарантирует нам, что найденное локально-оптимальное решение не будет единственным — муравьи будут искать и другие пути. Если мы моделируем процесс такого поведения на некотором графе, рёбра которого представляют собой возможные пути перемещения муравьев, в течение определённого времени, то наиболее обогащённый феромоном путь по рёбрам этого графа и будет являться решением задачи, полученным с помощью муравьиного алгоритма.

Обобщенная схема работы алгоритма сводится к следующему:

1. Создаём муравьев

– Стартовая точка, куда помещается муравей, зависит от ограничений, накладываемых условиями задачи, потому что для каждой задачи способ размещения муравьев является индивидуальным и определяющим. Все муравьи могут помещаться в одну точку, либо в разные точки с повторениями или без повторений.

– На этом же этапе задаётся начальный уровень феромона. Он инициализируется небольшим положительным числом для того, чтобы на начальном шаге вероятности перехода в следующую вершину не были нулевыми.

2. Ищем решения

– Вероятность перехода из вершины i в вершину j определяется по следующей формуле:

$$P_{ij}(t) = \frac{\tau_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{j \in \text{разрешенны м_узлам}} \tau_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta} \quad (1)$$

где $\tau(t)$ — уровень феромона; d — эвристическое расстояние; α, β — константные параметры.

– При $\alpha=0$ выбор ближайшего узла наиболее вероятен, то есть алгоритм становится жадным. При $\beta=0$ выбор происходит только на основании феромона, что приводит к неоптимальным решениям. Поэтому необходим компромисс между этими величинами, который находится экспериментально.

3. Обновляем феромон

Уровень феромона обновляется в соответствии с формулой

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{\substack{k \in \text{колонии, исп} \\ \text{ребро } (i,j)}} \frac{Q}{L_k} \quad (2)$$

где ρ — интенсивность испарения; $L(t)$ — цена текущего решения для k -го муравья; Q — параметр, имеющий значение порядка цены оптимального решения; $\frac{Q}{L_k(t)}$ — феромон,

откладываемый k -ым муравьём, использующим ребро (i,j) .

4. Дополнительные действия

Обычно здесь используется алгоритм локального поиска, однако он может также появиться и после поиска всех решений. Для того чтобы построить подходящий муравьиный алгоритм для решения какой-либо задачи, нужно пройти этапы, представленные на рис. 1.

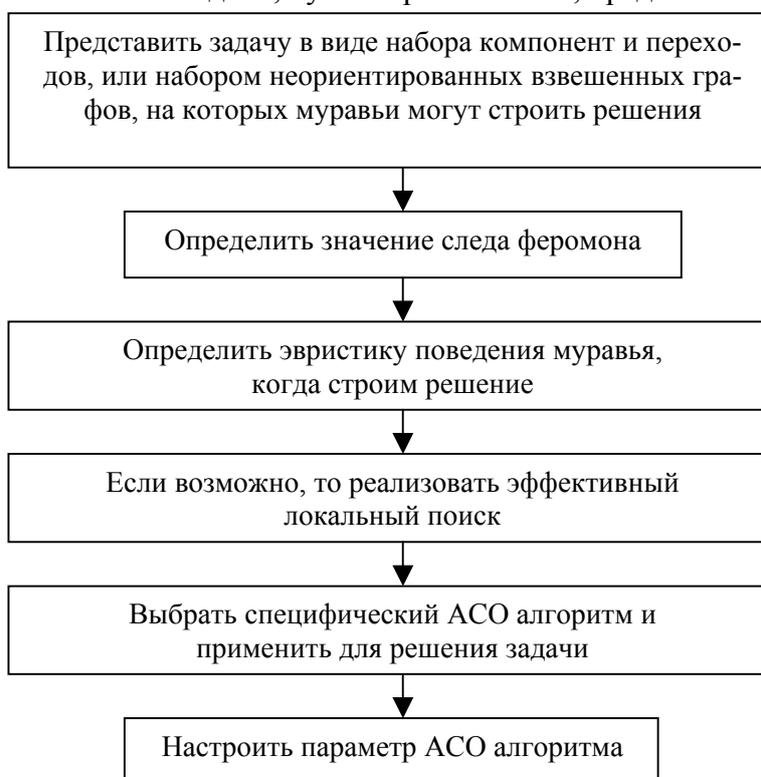


Рисунок 1 — Этапы решения задачи при помощи муравьиных алгоритмов

Определяющими при построении алгоритма являются:

- количество муравьёв;

- баланс между изучением и использованием;
- сочетание с жадными эвристиками или локальным поиском;
- момент, когда обновляется феромон.

Ряд экспериментов показывает [3], что эффективность муравьиных алгоритмов растёт с ростом размерности решаемых задач оптимизации. Качество получаемых решений во многом зависит от настроечных параметров в вероятностно- пропорциональном правиле выбора пути на основе текущего количества феромона и от параметров правил откладывания и испарения феромона. Важную роль играет и начальное распределение феромона, а также выбор условно оптимального решения на шаге инициализации.

Поскольку в основе муравьиного алгоритма лежит моделирование передвижения муравьёв по некоторым путям, то такой подход может стать наиболее эффективным способом поиска рациональных решений для задач оптимизации, допускающих графовую интерпретацию.

Решение задачи оптимизации затрагивает два этапа:

1. Минимизация времени выполнения каждой укрупненной работы (подпроекта) при помощи алгоритма List Scheduling(LS);
2. Применение муравьиного алгоритма на минимизированных по времени укрупненных работах для составления оптимального плана.

Можно составлять расписание без учета доступных ресурсов, но при учете количества доступных ресурсов задача составления расписания усложняется. Более того, здесь возникает проблема построения оптимального расписания проекта с учетом ограничения на ресурсы. В данной статье рассматривается задача с целевой функцией C_{min} — минимизация общего времени выполнения проекта. При этом будем учитывать только возобновимые ресурсы.

В результате работы алгоритма LS строятся только активные расписания, то есть такие, при которых каждая работа выполняется как можно раньше. Таким образом, оптимальное решение задачи построения расписания проекта с учетом ограничения на ресурсы следует искать среди активных расписаний.

Каждому активному расписанию соответствует некая перестановка (j_1, j_2, \dots, j_n) — последовательность постановки работ в расписание. Следовательно, задача построения оптимального расписания сводится к задаче нахождения оптимальной перестановки.

Выполним математическую постановку проблемы RCPSP.

Дано множество работ $N = \{i = 1 \dots n\}$ и r возобновимых ресурсов $k = 1 \dots r$. В каждый момент времени t доступно r_k единиц ресурса k . Заданы продолжительности обслуживания p_i для каждого требования $i = 1 \dots n$. Во время обслуживания требования i требуется r_{ik} единиц ресурса k .

Между некоторыми парами требований заданы ограничения предшествования: $i \rightarrow j$ означает, что обслуживание требования j начинается не раньше окончания требования i . Обозначим множество отношений предшествования $C = \{i \rightarrow j \mid i, j \in N\}$.

Обслуживание требований начинается в момент времени $t = 0$. Прерывание требований запрещены. Необходимо определить моменты времени начала обслуживания требований $S_i, i=1 \dots n$ так, чтобы:

1. В каждый момент времени t "занятое" количество ресурса k было меньше/равно r_k .
2. Не нарушались отношения предшествования, то есть $S_i + p_i \leq S_j$, если $i \rightarrow j$.
3. Минимизировалось время выполнения проекта.

Существует задача, в которой учитывается задержка между выполнением работ d_{ij} .

На практике количество выделенных на проект ресурсов может меняться со временем. В соответствии с практической задачей будем считать, что величины r_k зависят от времени t , т.е. $r_k(t)$.

Решение задачи RCPSP представляется в виде набора $S = (S_1 \dots S_n)$. А допустимому решению $(S_1 \dots S_n)$ соответствует некоторая перестановка $p = (j_1 \dots j_n)$ из n требований.

Для каждого требования i определим временное окно $[r_i, d_i]$, в котором требование i должно быть обслужено при любом допустимом расписании S :

$$C_{min}(S) \leq UB \tag{3}$$

где: UB — некоторая верхняя граница для C_{\min} .

Алгоритм List Scheduling диспетчеризации с учетом введенных обозначений будет состоять из следующих этапов.

1. Пусть EL — список всех работ без предшественников;
2. while $EL \neq \emptyset$ do
3. Выберем работу $j \in EL$;
4. $t = \max_{i \rightarrow j \in C} \{S_i + p_i\}$
5. while Существует ресурс k , что $r_{ij} > R_k(\tau)$ для некоторого $\tau \in [t, t+p_j)$ do
6. Вычислим минимальное значение $t_k > t$, что работа j может быть выполнена на интервале $[t_k, t_k + p_j)$, если рассматривается только ресурс k ;
7. $t = t_k$;
8. end while
9. Назначим выполнение работы j на интервал $[S_j, C_j) = [t, t+p_j)$;
10. Резервируем ресурсы под работу $R_k(\tau) = R_k(\tau) - r_{jk}$, $k = 1, \dots, r$, $\tau \in [t, t+p_j)$;
11. $EL = EL \setminus \{j\}$. Добавляем в EL последователей требования j , для которых все предшественники расставлены;
12. end while

Полученное решение зависит от выбора работы j на шаге 3 алгоритма. На основании данного алгоритма диспетчеризации можно построить точный алгоритм ветвей и границ, ветвление в котором происходит при выборе работы j . Применение муравьиных алгоритмов для решения этой задачи основано на идее последовательного приближения к оптимальному решению. Каждая итерация — "запуск искусственного муравья", который "пытается" по некоторому правилу выбрать наилучший маршрут к "пище" (к оптимуму функции), используя метки своих предшественников.

Каждый муравей выполняет алгоритм List Scheduling, выбирая требование $j \in EL$ исходя из значений параметров:

а) η_{ij} — эвристическая информация о том, насколько хорошим кажется постановка работы j на место i в перестановке (j_1, j_2, \dots, j_n) т.к. каждому активному расписанию соответствует некая перестановка). То есть насколько хорошим кажется выбор работы $j \in EL$ на i -ом срабатывания цикла. Этот параметр можно вычислить по правилу d: $\eta_{ij} = \frac{1}{d_j}, i = 1, \dots, n$, или

по правилу CriticalPatch (критический путь): $\eta_{ij} = d_j - r_j, i = 1, \dots, n$

б) τ_{ij} — "след" (в природе: след феромона). После каждой итерации этот параметр корректируется. Параметр показывает насколько "хорошим" для работы j оказалась позиция i в перестановке (j_1, j_2, \dots, j_n) . То есть это накопленная статистическая информация о качестве выбора для позиции i работы j , в то время как η_{ij} характеризует предполагаемую выгоду такой постановки при недостатке накопленной информации.

Параметры η_{ij} рассчитываются один раз перед первой итерацией.

На каждом шаге вычисляется матрица вероятностей перехода:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in EL} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta}, & j \in EL \\ 0, & j \notin EL \end{cases} \quad (4)$$

Правило, по которому на позицию i выбирается требование j определяется следующим образом:

$$\begin{cases} j = \arg \max_{h \in EL} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta, & q < q_0 \\ j \text{ определяется случайным образом, согласно распределению вероятностей } \rho_{ij}, & q \geq q_0 \end{cases}$$

где: $0 \leq q_0 \leq 1$ — параметр алгоритма, q_0 вычисляется случайным образом на каждом шаге.

После того, как работа j была поставлено на позицию i в перестановке, пересчитывается "локальный след":

$$\tau_{ij} = (1 - p)\tau_{ij} + p\tau_0, \quad (5)$$

где: $\tau_0, p \in [0,1]$ — некие параметры алгоритма.

После каждой итерации "глобальный след" τ_{ij} корректируется по правилу: если в "лучшем" найденном расписании на позиции i перестановки находится работа j .

$$\tau_{ij} = (1 - p)\tau_{ij} + p / C_{\min}^* \quad (6)$$

Иначе:
$$\tau_{ij} = (1 - p)\tau_{ij} \quad (7)$$

Значение C_{\min}^* — время выполнения проекта для "лучшего" найденного расписания.

У пользователя есть возможность настроить параметры алгоритма, выбрать эвристики для расчета η_{ij} определить влияние каждой эвристики на вероятностей перехода p_{ij} . В алгоритме предлагается на выбор 6 эвристик $\eta_{ij}^1, \eta_{ij}^2, \eta_{ij}^3, \eta_{ij}^4, \eta_{ij}^5, \eta_{ij}^6$

Параметры p_{ij} вычисляются следующим образом:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}^1]^{\beta^1} [\eta_{ij}^2]^{\beta^2} [\eta_{ij}^3]^{\beta^3} [\eta_{ij}^4]^{\beta^4} [\eta_{ij}^5]^{\beta^5} [\eta_{ij}^6]^{\beta^6}}{\sum_{h \in EL} [\tau_{ih}]^\alpha [\eta_{ih}^1]^{\beta^1} [\eta_{ih}^2]^{\beta^2} [\eta_{ih}^3]^{\beta^3} [\eta_{ih}^4]^{\beta^4} [\eta_{ih}^5]^{\beta^5} [\eta_{ih}^6]^{\beta^6}}, & j \in EL \\ 0, & j \notin EL \end{cases} \quad (8)$$

У пользователя есть возможность настроить параметры $\alpha, \eta_{ij}^1, \eta_{ij}^2, \eta_{ij}^3, \eta_{ij}^4, \eta_{ij}^5, \eta_{ij}^6, p, m, q_0$.

То есть можно настроить количество муравьев и количество запусков алгоритма.

В процессе работы алгоритма выводятся сообщения о построенных расписаниях и найденных значениях C_{\min} . В любой момент пользователь может прерваться и сохранить полученное расписание или перенастроить параметры алгоритма.

Выводы.

Алгоритм был применен к планированию работ для строительной организации. Весь процесс строительства был разбит на укрупненные работы, каждая из которых, в свою очередь, состоит из подрбот. Алгоритм применялся последовательно в два этапа. В первую очередь минимизируется время выполнения каждой группы работы. Следующий этап — применение муравьиного алгоритма на минимизированных по времени укрупненных работах для составления оптимального плана.

Для выбранных работ с помощью муравьиного алгоритма было найдено расписание, при котором план выполнения работ занял 4,5 месяца, в то время как обычно используемый алгоритм строит расписание, при котором $C_{\min} = 6$ месяцев. Основываясь на приведенных результатах можно сделать вывод о целесообразности использования метода муравьиных алгоритмов при составлении оптимального по длительности плана строительных работ.

Литература

1. Малярченко, И.С. Планирование и оптимизация: от Вергилия до... APS-системы [Текст] / И.С. Малярченко // PC WEEK/RE. — 2006. — №27(537). — С. 21–25.
2. Кузьменко В. М., Таран С. В. Анализ современных методов искусственного интеллекта применительно к задачам календарного планирования [Текст] / В. М. Кузьменко, С.В. Таран // Вісник Донбаської державної машинобудівної академії. — 2006. — № 1Е. — С. 38–42.
3. Штовба, С.Д. Муравьиные алгоритмы [Текст] / С. Д. Штовба // Exponenta Pro: Математика в приложениях. — 2003. — №4. — С. 9–15.

Здано в редакцію:
24.02.2009р.

Рекомендовано до друку:
д.т.н, проф. Чичикало Н.І.