

Выбор параметров модифицированного алгоритма муравьиных колоний для оптимизации структуры автоматизированной системы управления крупным промышленным объектом

Приводится описание задачи поиска оптимальной структуры распределенной системы управления технологическими процессами крупного промышленного объекта. Предлагаются различные способы решения задачи. Дается описание метаэвристического алгоритма муравьиных колоний для построения оптимальной древовидной структуры системы управления. Рассматриваются варианты выбора параметров указанного алгоритма.

Ключевые слова: распределенная система управления, метаэвристический алгоритм, автоматизированная система управления, структурная оптимизация.

Технологические процессы современных промышленных предприятий требуют надежных и функциональных систем управления. Особенности требования предъявляются к автоматизированным системам управления технологическими процессами (АСУТП) крупных промышленных объектов, характеризующихся большим количеством параметров и территориальной распределенностью.

Как правило, АСУТП таких объектов строятся на базе программно-технических комплексов (ПТК). ПТК АСУТП состоит из большого количества серийно выпускаемых промышленностью компонентов, соединенных различными каналами связи. Поскольку характеристики компонентов задаются заводами-изготовителями, функциональные возможности полученной системы во многом определяются ее структурой.

Оптимизация структуры производственной системы является важной задачей в различных отраслях [1]. Актуальной является задача построения такой структуры системы управления, которая бы надлежащим образом решала задачу управления промышленным объектом, удовлетворяла требуемым ограничениям и имела бы при этом минимальную стоимость.

Критерии оптимизации и ограничения для системы сформулированы в [2].

Постановка задачи

Современные АСУТП, как правило, имеют древовидную иерархическую структуру. Такую структуру можно представить в виде графа $G = (V, E)$ (рис. 1), где $v \in V$ – компоненты системы, устройства (вершины графа), E – ребра графа (каналы связи, соединяющие устройства).

Иерархия содержит S уровней: уровень 1 – корень дерева, уровень S – листья дерева.

Пусть задано множество возможных типов устройств $U = \{u_1, \dots, u_U\}$, состоящее из U типов устройств. Данное множество представляет собой различные серийно выпускаемые промышленностью компоненты системы управления.

Каждое устройство может обладать функцией обработки информации (процессор, обработчик),

либо только функцией передачи информации на вышестоящий или нижестоящий уровень (ретранслятор).

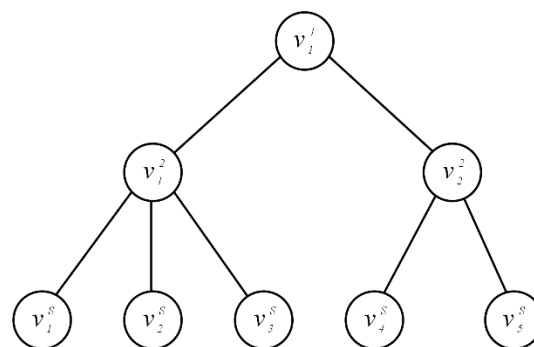


Рис. 1. Структура системы управления в виде графа

Каждый тип $u_i \in U$ узла структуры характеризуется рядом параметров, таких как стоимость устройства, количество подключаемых физических каналов, максимальный объем памяти, вероятность отказа устройства, производительность (время выполнения одной программной инструкции), режим работы (обработчик/ретранслятор), максимальное количество дочерних устройств, задержка передачи (только для ретрансляторов).

Для характеристики объекта управления введем множество контуров управления $A = \{a_1, \dots, a_A\}$, состоящее из A контуров.

Каждый контур физически должен быть подключен только к одному устройству иерархии, находящемуся на нижнем уровне S – уровне листьев дерева. В то же время каждый контур должен быть назначен обработчику, который может находиться на любом уровне иерархии, включая корень.

Для соответствия модели реальным техническим системам вводится ряд ограничений: по надежности системы, времени обработки контура и др. Полная постановка задачи приведена в [3]. Целевая функция, определяющая стоимость системы как сумму стоимостей ее компонентов, будет иметь вид:

$$C_0 = \min_g \sum_{v \in V} C_v, \quad (1)$$

Таким образом, необходимо подобрать такую иерархическую структуру из узлов заранее заданных типов, чтобы доставить минимум целевой функции (1) при заданных ограничениях.

Методы решения

Поставленная задача является задачей дискретной оптимизации с множественными ограничениями. Задача может быть решена точными, приближенными методами или методами, основанными на нейросетевых технологиях.

Точные методы дают точное решение, но требуют больших вычислительных ресурсов, поэтому применимы только при небольших размерностях задачи.

Приближенные методы, такие как метаэвристические биоинспирированные алгоритмы [4], широко применяются для решения подобного класса задач. Данные методы не дают точного решения, однако менее требовательны к вычислительным ресурсам, и при определенных условиях могут быть использованы для решения описанной в настоящей статье задачи.

Для решения задачи оптимизации структуры системы управления был выбран алгоритм муравьиных колоний. Данный алгоритм находит в настоящее время широкое применение для решения различных задач оптимизации [5, 6].

Алгоритм муравьиных колоний

Описание алгоритма муравьиных колоний (Ant Colony Optimization (ACO) – англ.) дано в [7]. Особенностью алгоритма является то, что в нем происходит запоминание лучших решений посредством испаряющихся с определенной скоростью следов – феромонов [5].

Выбор каждого шага производится согласно правилу:

$$P_i = \frac{\tau_i^\alpha \eta_i^\beta}{\sum_{k=0}^N \tau_k^\alpha \eta_k^\beta}, \quad (2)$$

где P_i – вероятность выбора i -го варианта, τ – количество феромона, η – стоимость варианта, α – параметр, характеризующий вес феромона («стадность»), β – параметр, характеризующий вес эвристики («жадность»), N – общее количество допустимых вариантов.

Обновление количества феромона происходит по формуле (3):

$$\tau_{i+1} = (1 - \rho)\tau_i + \Delta\tau_i, \quad (3)$$

где ρ – коэффициент испарения феромона, $\Delta\tau_i$ – количество отложенного феромона (депозит).

В данной задаче каждый шаг муравья разбивается на несколько действий (выбор типа нового устройства, выбор количества дочерних устройств, назначение контуров и т.д.). Каждый выбор делается с вероятностью, определяемой выражением (2).

Муравей последовательно стоит дерево от корня к листьям путем добавления вершин графа. Каждая итерация алгоритма завершается, если построено допустимое дерево. После построения производится

проверка ограничений. В случае нарушения ограничений решение отбрасывается.

Для реализации алгоритма было разработано программное обеспечение на языке Python, с помощью которого был решен ряд численных примеров.

Примененный метаэвристический алгоритм не дает точное решение. Близость полученного решения к оптимальному во многом определяется способом задания начальных условий и выбором параметров алгоритма. Как видно из рис. 2, алгоритм имеет склонность застревать на локальных оптимумах.

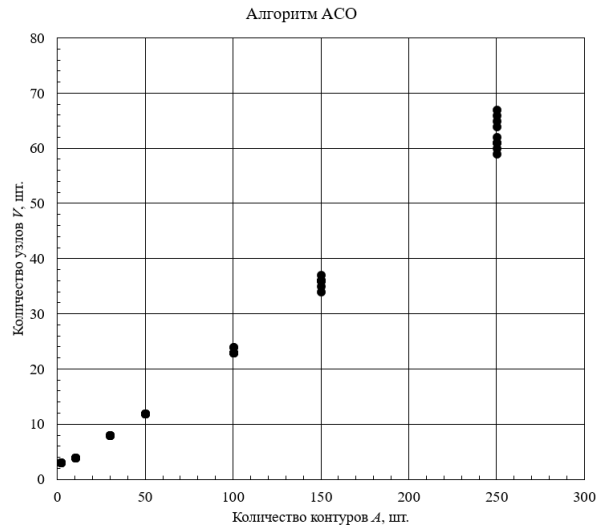


Рис. 2. Результат работы алгоритма до модификации

Модификация алгоритма

Разработано достаточно большое количество модификаций муравьиного алгоритма, решающих те или иные присущие алгоритму проблемы [8].

Для предотвращения застревания на локальных оптимумах в данном исследовании алгоритм был модифицирован путем добавления локального поиска вблизи предполагаемого оптимума. Локальный поиск производит следующие действия: замена типа устройства, слияние соседних узлов, удаление устройства, перенос контура на соседний узел.

Результат работы алгоритма после добавления локального поиска представлен на рис. 3.

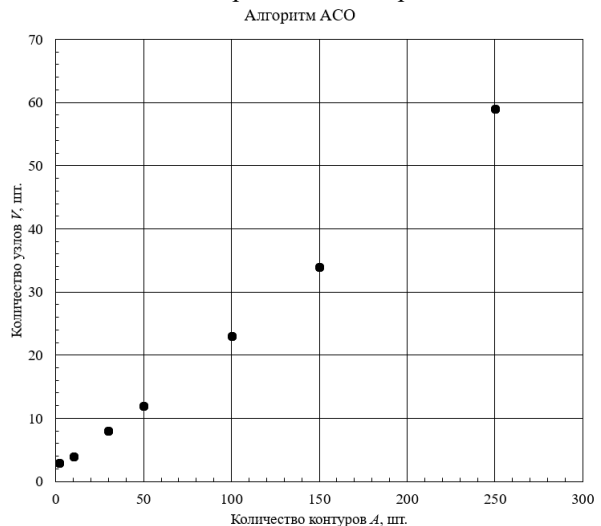


Рис. 3. Результат работы алгоритма после модификации

Выбор параметров алгоритма

Выбор параметров алгоритма в значительной мере влияет на сходимость. С помощью разработанного программного обеспечения решен ряд тестовых примеров для оценки влияния параметров на сходимость алгоритма. Вычисления проводились для системы с тремя уровнями иерархии ($S = 3$), количеством типов устройств $U = 4$ и количеством контуров управления $A = 39$. Для эксперимента были выбраны параметры, указанные в таблице 1. Разработанное программное обеспечение позволяет задавать параметры алгоритма как в виде констант, так и в виде функций от аргумента n , где n – номер итерации алгоритма.

Таблица 1

Номер эксперимента	Параметры алгоритма		
	ρ	α	β
1	0,25	1,0	0,0
2	0,25	$1 + 0,1n$	$1/n$
3	$0,25 + 0,05n$	$1 + 0,2n$	$1/(2n)$
4	0,25	$1 + 1/n$	$0,02n$

По полученным в результате вычислительных экспериментов данным построены графики сходимости алгоритма. Графики показаны на рис. 4.

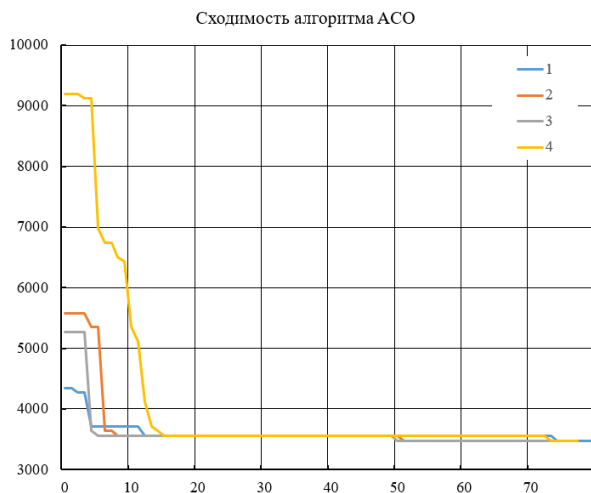


Рис. 4. Сходимость алгоритма

Вычисления показывают, что при всех заданных параметрах алгоритм сходится к оптимальному значению стоимости $C_0 = 3478$, полученному при решении задачи точным методом. При этом наилучшие результаты достигаются при использовании постоянных параметров при $\beta = 0$. При увеличении «жадности» алгоритма происходит замедление сходимости. Однако, как показали другие эксперименты, при

больших размерностях задачи общими рекомендациями будет увеличение «жадности» на начальном этапе работы алгоритма с увеличением влияния феромонов по мере приближения к целевому значению.

Заключение

В статье описана задача оптимизации иерархической структуры промышленной системы управления, построенной на базе программно-технического комплекса, приведены методы решения. Проведено исследование решения задачи с использованием метаэвристического алгоритма оптимизации. Даны рекомендации по модификации алгоритма, проанализирован выбор параметров алгоритма.

Литература

1. Резчиков А. Ф. Алгоритм оптимизации структур робототехнических сборочных комплексов / А. Ф. Резчиков, А. А. Большаков, Д. Ю. Петров, Д. Д. Яковлев // Автоматизация в промышленности. – 2025. – № 5. – С. 3-10.
2. Закирзянов Р. М. Критерии выбора оптимальной структуры распределенной системы управления технологическими процессами крупных промышленных предприятий / Р. М. Закирзянов // Математические методы в технологиях и технике. – 2024. – № 10. – С. 17-23.
3. Закирзянов Р. М. Применение метаэвристических алгоритмов для оптимизации структуры промышленной системы управления / Р. М. Закирзянов // VI Международная научная конференция по проблемам управления в технических системах (ПУТС-2025). Сборник докладов. Санкт-Петербург. 26 – 28 сентября 2025 г. СПб.: СПбГЭТУ «ЛЭТИ». – 2025. – С. 35-37.
4. Гладков Л. А. Интеллектуальные системы: модели и методы метаэвристической оптимизации / Л. А. Гладков, Ю. А. Кравченко, В. В. Курейчик, С. И. Родзин. – Чебоксары: ООО «Издательский дом «Среда», 2024. – 228 с.
5. Shyu, Shyong & Yin, Peng-Yeng & Lin, Bertrand & Haouari, Mohamed. (2003). Ant-Tree: An ant colony optimization approach to the generalized minimum spanning tree problem. J. Exp. Theor. Artif. Intell. 15. 103-112. 10.1080/0952813021000032699.
6. Lebedev B. K. Modified swarm algorithm «ant tree» in the problem of diversification of tractor resources spaces / B. K. Lebedev, O. B. Lebedev, O. A. Purchina // IOP Conference Series: Materials Science and Engineering, Novosibirsk, 22–27 мая 2020 года. Vol. 918. – Novosibirsk, 2020. – P. 012082.
7. M. Dorigo, T. Stützle, 2004. Ant Colony Optimization, MIT Press. ISBN 0-262-04219-3.
8. Курейчик В. М. О некоторых модификациях муравьиного алгоритма / В. М. Курейчик, А. А. Кажаров // Известия ЮФУ. Технические науки. – 2008. – № 4(81). – С. 7-12.

Закирзянов Руслан Марселевич

Технический директор ООО «НЕКСТ инжиниринг»

Эл. почта: zr@nexteng.ru