

Повышение эффективности гибридного алгоритма построения иерархической структуры АСУТП через оптимизацию гиперпараметров

Р. М. Закирзянов

ООО «НЕКСТ инжиниринг»

zr@nexteng.ru

Аннотация. Автоматизированные системы управления технологическими процессами, широко применяемые в промышленности, имеют иерархическую многоуровневую древовидную архитектуру и строятся на базе специализированных программно-технических комплексов. Построение системы управления из готовых выпускаемых промышленностью компонентов с предопределенными фиксированными характеристиками является сложной задачей комбинаторной оптимизации с множественными ограничениями. Рассматривается гибридный метод построения, состоящий из детерминированного алгоритма построения древовидной структуры и метаэвристического алгоритма муравьиных колоний для выбора устройств. Эффективность применяемого метаэвристического алгоритма зависит от значений его гиперпараметров, которые предлагается оптимизировать. В работе представлена общая постановка задачи, приведены критерии оптимизации, описан способ решения, решен ряд численных примеров.

Ключевые слова: структурная оптимизация; распределенная система управления; метаэвристический алгоритм

I. ВВЕДЕНИЕ

Промышленные автоматизированные системы управления технологическими процессами (АСУТП) являются ключевым элементом современных промышленных предприятий [1], на которых предъявляются высокие требования к безопасности, надежности и качеству продукции. Для отраслей с непрерывными технологическими процессами, таких как химическая, нефтехимическая и нефтегазовая отрасли, характерна достаточно высокая степень автоматизации. На таких предприятиях управление технологическими процессами происходит в основном в автоматическом режиме, при этом операторы-технологи выполняют функции контроля и задания технологических режимов.

Крупные АСУТП реализуются с использованием специализированного оборудования и программного обеспечения, созданного из серийно выпускаемых промышленностью компонентов, таких как, например, программируемые логические контроллеры (ПЛК), устройства ввода/вывода и сетевые коммутаторы. Каждый компонент имеет предопределенные характеристики, такие как пропускная способность каналов связи, количество и типы коммуникационных интерфейсов, максимальный объем памяти и производительность.

При проектировании системы инженеры должны построить из доступных серийных компонентов такую

структуру, которая удовлетворяет техническим требованиям и имеет при этом минимальную стоимость. Поскольку свойства компонентов жестко заданы заводами-изготовителями, функциональные характеристики системы в значительной степени определяются выбранной структурой, которая, как правило, является иерархической и многоуровневой. Крупные территориально распределенные системы с большим количеством параметров контроля и управления обычно называют распределенными системами управления (PCY). В таких системах структурная оптимизация приобретает особое значение. АСУТП могут быть построены с использованием компонентов одного производителя или путем интеграции оборудования от нескольких различных производителей, например, в рамках открытых архитектур, таких как OPAS [2].

На практике проектные решения часто основываются на опыте проектировщика и рекомендациях заводоизготовителей, что не всегда приводит к оптимальным решениям. На ранних этапах проектирования инженерам необходимо подобрать оборудование, а также оценить стоимость и основные характеристики системы при неполной информации об объекте управления. Эти оценки имеют очень важное значение при реализации проекта и планировании ресурсов.

Структурный синтез АСУТП можно сформулировать как задачу комбинаторной оптимизации с множественными ограничениями. Такие задачи могут быть решены точными или приближенными методами. Точные методы применяются при относительно небольших размерностях задачи из-за их вычислительной сложности. Для крупномасштабных систем чаще применяются приближенные алгоритмы оптимизации, обеспечивающие решения, близкие к оптимальным, за разумное время. Среди таких методов широкое распространение получили так называемые метаэвристические алгоритмы [3], такие как методы локального поиска, методы, основанные на популяциях, методы, вдохновленные эволюционными процессами, и методы роевого интеллекта.

В предыдущих работах [4–6] для построения древовидных иерархических структур АСУТП в условиях ограниченной информации был предложен подход на основе алгоритма муравьиных колоний (Ant Colony Optimization (ACO) – англ.). В данной работе этот подход модифицирован за счет включения адаптивной эвристики и тонкой двухкритериальной настройки

гиперпараметров алгоритма. Применяемый подход проиллюстрирован численными примерами.

II. АНАЛИЗ ЛИТЕРАТУРЫ

Структурная оптимизация технических систем является важной областью современных исследований. Задачи структурного синтеза изучаются в различных областях, в том числе и при проектировании распределенных иерархических технических систем [7].

Среди приближенных методов широкое распространение получили метаэвристические алгоритмы. К настоящему времени разработано большое количество биоинспирированных (вдохновленных природными явлениями) методов, таких как генетические алгоритмы, алгоритм пчелиной колонии, алгоритм серых волков, алгоритм муравьиных колоний, алгоритм роя частиц и другие [8]. Детальный обзор метаэвристических алгоритмов представлен в [9]. Эти методы успешно применяются для решения различных инженерных задач оптимизации.

В последнее время широко изучаются гибридные метаэвристики, которые объединяют различные алгоритмы оптимизации. Пример гибридного алгоритма, сочетающего методы роевого интеллекта и локального поиска, представлен в [10]. Также активно изучаются возможности интеграции метаэвристик с методами машинного обучения. Эффективность метаэвристического алгоритма во многом зависит от настройки его параметров (гиперпараметров).

Алгоритм муравьиных колоний был предложен в 1992 году М. Дориго для решения задачи коммивояжера [11]. В настоящее время разработано множество модификаций данного алгоритма [12]. Алгоритм муравьиных колоний применяется для решения различных задач оптимизации, таких как настройка параметров регулятора, задачи поиска оптимального маршрута и задачи размещения объектов [13–15].

Проблема структурной оптимизации АСУТП, рассматриваемая в данной работе, ранее была сформулирована в [4, 6]. Несмотря на большое количество публикаций, посвященных метаэвристикам и структурной оптимизации, задача оптимизации иерархической структуры АСУТП остается недостаточно изученной.

III. ПОСТАНОВКА ЗАДАЧИ

Детальная формальная постановка задачи синтеза оптимальной архитектуры АСУТП представлена в [5]. Ниже представлены основные положения.

Архитектура АСУТП представлена в виде дерева (ациклического графа) $G = (\mathcal{V}, \mathcal{E})$, где \mathcal{V} – множество вершин, соответствующих устройствам $v \in \mathcal{V}$, а \mathcal{E} – множество ребер, представляющих каналы связи между ними. Пример такой структуры показан на рис. 1.

Количество уровней иерархии задается проектировщиком системы.

Пусть $\mathcal{U} = \{u_1, \dots, u_U\}$ – множество доступных типов устройств. Каждое устройство выполняет циклическую программу, которая включает три основных этапа: сбор данных, обработка данных (т.е. выполнение алгоритмов управления) и передача данных. Устройства

подразделяются на два класса: обработчики и ретрансляторы.

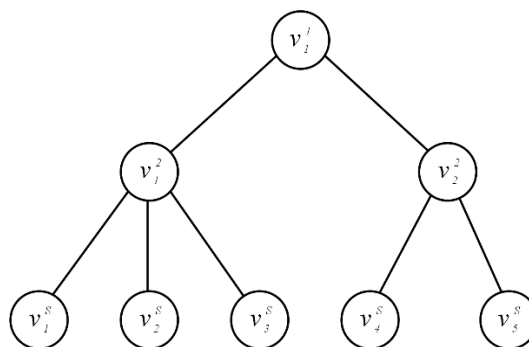


Рис. 1. Иерархическая структура АСУТП

Каждый тип устройства $u_i \in \mathcal{U}$ характеризуется следующим вектором параметров:

$$u_i = (C_i, N_i, R_i, P_i, T_i, y_i, M_i, \tau_i), \quad i = 1, \dots, U, \quad (1)$$

где $C_i \in \mathbb{R}^+$ – стоимость устройства, $N_i \in \mathbb{N}$ – количество физических каналов для подключения контуров управления, $R_i \in \mathbb{R}^+$ – доступная память, $P_i \in [0, 1]$ – вероятность отказа, $T_i \in \mathbb{R}^+$ – время выполнения одной инструкции, $y_i \in \{0, 1\}$ – роль устройства (0 – ретранслятор, 1 – обработчик), $M_i \in \mathbb{N}$ – максимальное количество дочерних узлов и $\tau_i \in \mathbb{R}^+$ – временная задержка передачи сигнала.

Пусть $\mathcal{A} = \{a_1, \dots, a_A\}$ – множество контуров управления. Каждый контур управления определяется следующим образом:

$$a_j = (n_j, r_j, w_j), \quad j = 1, \dots, A, \quad (2)$$

где $n_j \in \mathbb{N}$ количество физических сигналов в контуре, $r_j \in \mathbb{R}^+$ – требуемый объем памяти для данного контура, $w_j \in \mathbb{N}$ – количество программных инструкций, необходимых для обработки данного контура.

Для описания подключения и назначения контуров вводятся бинарные переменные $x_{va} \in \{0, 1\}$ и $z_{va} \in \{0, 1\}$. $x_{va} = 1$ означает, что контур a физически подключен к узлу v , иначе $x_{va} = 0$. $z_{va} = 1$ означает, что контур a назначен на обработчик v , иначе $z_{va} = 0$.

Формальная постановка задачи включает ограничения, связанные с техническими характеристиками устройств (ограничения по объему памяти, надежности и производительности), а также структурные ограничения системы.

Задача оптимизации состоит в определении допустимой иерархической архитектуры G , которая минимизирует общую стоимость системы:

$$C^* = \min_{G \in \mathcal{G}} \sum_{v \in \mathcal{V}(G)} C_v \quad (3)$$

при соблюдении наложенных ограничений. Здесь \mathcal{G} обозначает множество всех допустимых иерархических графов, которые могут быть построены из доступных типов устройств.

IV. ОПИСАНИЕ МЕТОДА

Задача синтеза архитектуры АСУТП представляет собой задачу комбинаторной оптимизации с множественными ограничениями. Прямое использование для решения такой задачи метаэвристических алгоритмов на основе популяций (например, генетического алгоритма или алгоритма роя частиц) обычно приводит к большому количеству недопустимых решений из-за нарушений ограничений. Это приводит к неэффективному поиску и увеличению вычислительных затрат. Для решения этой проблемы предлагается гибридный подход, сочетающий детерминированный алгоритм построения дерева с алгоритмом выбора устройств на основе метаэвристики.

Детерминированный алгоритм поэтапно строит древовидную структуру, начиная со ствола дерева, затем контуры управления подключаются к листовым узлам. При нарушении ограничений по памяти или производительности производится добавление новых устройств или ветвей дерева. Так как при построении дерева структурные ограничения соблюдаются автоматически, в значительной степени удается избежать недопустимых структур. В определенных точках принятия решения, где необходимо выбрать тип устройства, выбор осуществляется с помощью алгоритма муравьиных колоний (АСО), который хорошо подходит для решения подобных конструктивных задач.

На каждом шаге принятия решения устройство-кандидат i выбирается вероятностным образом:

$$P_i = \frac{\tau_i^\alpha \eta_i^\beta}{\sum_{k=0}^N \tau_k^\alpha \eta_k^\beta}, \quad (4)$$

где τ_i характеризует влияние феромона, η_i характеризует влияние эвристики, α и β – весовые параметры. Значения феромона обновляются итеративно:

$$\tau_i \leftarrow (1 - \rho)\tau_i + \Delta\tau_i, \quad (5)$$

где $\rho \in (0,1)$ – параметр, характеризующий интенсивность испарения феромона.

По сравнению с методом случайного жадного поиска и методом Монте-Карло алгоритм АСО демонстрирует более высокое качество решения, но все еще испытывает трудности вблизи границ допустимости, где допустимых решений становится мало. Это обосновывает использование адаптивной эвристики.

Базовая эвристика, основанная на стоимости (6), как правило, отдает предпочтение недорогим устройствам, но может привести к неоптимальным структурам.

$$\eta_i = \frac{1}{c_i}. \quad (6)$$

Для решения этой проблемы вводится адаптивная эвристика, зависящая от текущего уровня иерархии. Для листовых узлов ($s = S$) приоритет отдается устройствам с большим количеством каналов:

$$\eta_i = \frac{N_i^2}{c_i}, \quad s = S. \quad (7)$$

Для внутренних уровней ($s < S$) дополнительно учитывается возможность ветвления (8).

$$\eta_i = \frac{N_i^2 M_i}{c_i}, \quad s < S. \quad (8)$$

Эвристический параметр вычисляется на основе количества каналов и перспективы ветвления.

Параметры АСО α , β и ρ влияют как на допустимость решения, так и на стабильность поиска глобального минимума, поэтому настройка гиперпараметров формулируется как многокритериальная задача оптимизации. Такая задача может быть решена с использованием подхода Парето. Такой подход позволяет найти компромисс между допустимостью и стабильностью.

Кроме этого, к лучшему найденному решению применяется дополнение в виде локального поиска (Local Search (LS) – англ.), который производит замену случайно выбранного устройства на альтернативный, но допустимый тип. Замена принимается, если она улучшает целевую функцию, сохраняя при этом допустимость решения.

V. РЕЗУЛЬТАТЫ

Для реализации описанного метода было разработано программное обеспечение на языке Python. В численном примере, параметры которого приведены в таблице 1, рассмотрено шесть типов устройств (два обработчика и четыре ретранслятора). Обработчики имеют относительно высокую стоимость и не имеют каналов для физического подключения полевых устройств. Ретрансляторы позволяют подключать полевые устройства и имеют коммуникационные интерфейсы.

ТАБЛИЦА 1. Типы устройств

Тип	Параметры устройства							
	C_i	N_i	R_i	P_i	T_i (с)	y_i	M_i	τ_i (с)
u_1	1300	0	512	0,001	0,007	1	4	0
u_2	950	0	256	0,003	0,003	1	3	0
u_3	110	6	-	0,0015	-	0	6	0,02
u_4	90	4	-	0,002	-	0	8	0,03
u_5	70	1	-	0,002	-	0	8	0,02
u_6	55	2	-	0,002	-	0	4	0,04

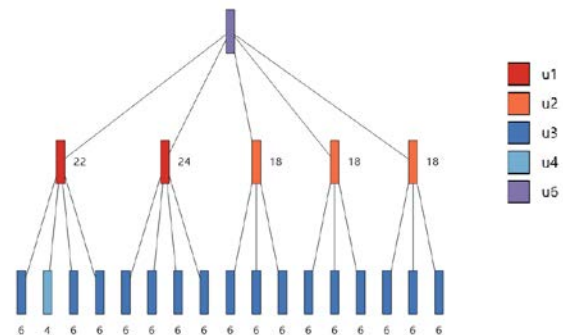


Рис. 2. Структура АСУТП для $A = 100$ и $S = 3$

Пример оптимальной структуры для $A = 100$ контуров управления и $S = 3$ уровней иерархии показан на рис. 2.

Было рассмотрено несколько модификаций АСО, включая классическую муравьиную систему (AS), систему Min-Max (MMAS) и систему с элитными муравьями (EAS), каждая из которых была протестирована как с базовой эвристикой, основанной на стоимости, так и с предложенной адаптивной эвристикой. В таблице результатов запуска алгоритма с адаптивной эвристикой обозначены как «new».

Эффективность алгоритма оценивалась с помощью показателя реализуемости W (процент успешных запусков) и коэффициента вариации CV^* , нормированного по лучшему решению. Поскольку производительность АСО зависит от его параметров, была применена оптимизация на основе алгоритма NSGA-II [16]. Полученный фронт Парето был использован для выбора значений параметров с приоритетом реализуемости, в результате чего были получены значения $\alpha = 2,2546$, $\beta = 8,3215$, и $\rho = 0,7856$. В таблице результатов запуска алгоритма с данными параметрами обозначены как «Optimized».

ТАБЛИЦА II. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

№	Алгоритм	C_{min}	C_{avg}	$CV^*(\%)$
1	AS	7355	7411	1.62
2	MMAS	7355	7370	0.25
3	Elite MMAS	7355	7361	0.15
4	Elite MMAS new	7355	7368	0.24
5	AS new Optimized	8090	8274	3.05
6	AS new Optimized + LS	7740	7880	2.84
7	Elite MMAS new Optimized	8090	8141	1.34
8	Elite MMAS new Optimized + LS	7355	7384	1.05

Результаты эксперимента показывают, что только оптимизированные варианты сохраняют высокую осуществимость во всем диапазоне количества контуров управления. Однако усиление влияния эвристических или феромонных компонентов может снизить стабильность решения и привести к преждевременной сходимости. Для решения этой проблемы был включен этап локального поиска, описанный в главе IV.

VI. ЗАКЛЮЧЕНИЕ

Задача поиска оптимальной структуры АСУТП при ограниченной исходной информации об объекте управления может быть рассмотрена как задача комбинаторной оптимизации с множественными ограничениями. Для решения данной задачи предложен гибридный метод, сочетающий детерминированный алгоритм построения дерева иерархии и метаэвристический выбор устройств. Возможными направлениями дальнейших исследований являются повышение точности модели для более точного отражения характеристик реальных промышленных устройств, сравнение различных алгоритмов оптимизации, а также изучение интеграции модели с методами машинного обучения.

СПИСОК ЛИТЕРАТУРЫ

- [1] Закирзянов Р.М., Латыпов А.Р., Шпилев А.И. Опыт импортозамещения АСУТП непрерывного производства в химической промышленности // Автоматизация в промышленности. 2025. № 12. С. 43-48. EDN MWSHQV.
- [2] R.D. Bartusiak, S. Bitar, D.L. DeBari, B.G. Houk, D. Stevens, B. Fitzpatrick, P. Sloan "Open Process Automation: A standards-based, open, secure, interoperable process control architecture", Control Engineering Practice, vol. 121, 2022, doi: 10.1016/j.conengprac.2021.105034.
- [3] 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. EDN: KNHQTN.
- [4] R. Zakirzyanov "A Method For Optimizing The Structure Of The Software And Hardware Complex Of A Distributed Process Control System For Large Industrial Enterprises", 18th International Conference on Management of Large-Scale System Development (MLSD), Moscow, Russian Federation, 2025, pp. 1-5, doi: 10.1109/MLSD65526.2025.11220659.
- [5] Закирзянов Р.М. Использование биоинспирированного метаэвристического алгоритма оптимизации для генерации иерархической структуры программно-технического комплекса АСУТП // Вестник Самарского государственного технического университета. Серия «Технические науки», 2026. (в печати)
- [6] Закирзянов Р.М. Применение метаэвристических алгоритмов для оптимизации структуры промышленной системы управления // Всероссийская научная конференция по проблемам управления в технических системах. 2025. Т. 1. С. 35-37. EDN QETZLN.
- [7] J.H. Bussemaker, P. Saves, N. Bartoli "System architecture optimization strategies: dealing with expensive hierarchical problems", Journal of Global Optimization, vol. 91, pp. 851-895, 2025, doi: 10.1007/s10898-024-01443-8.
- [8] A. Shaban, S. Almufti, R. Asaad "Metaheuristic Algorithms for Engineering and Combinatorial Optimization: A Comparative Study Across Problems Categories and Benchmarks", International Journal of Scientific World, 2025, no. 11, pp. 38-49, doi: 10.14419/0hndc578.
- [9] K. Sörensen, M. Sevaux, F. Glover "A history of metaheuristics", 2018. doi: 10.1007/978-3-319-07124-4_4.
- [10] L.M. Gambardella, M. Dorigo "An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem", INFORMS Journal on Computing, 2000, no. 12(3), pp. 237-255, doi: 10.1287/ijoc.12.3.237.12636.
- [11] M. Dorigo, T. Stützle "Ant Colony Optimization", MIT Press, 2004. doi: 10.7551/mitpress/1290.001.0001.
- [12] S.M. Almufti, R.P. Maribojoc, A.V. Pahuriray "Ant Based System: Overview, Modifications and Applications from 1992 to 2022", Polaris Global Journal of Scholarly Research and Trends, 2022, no. 1(1), pp. 29-37, doi: 10.58429/pgjsrt.v1n1a85.
- [13] A. Abushawish, M. Hamadeh, A. Nassif "PID Controller Gains Tuning Using Metaheuristic Optimization Methods: A survey", International Journal of Computers, 2020, vol. 14, no. 14, pp. 87-95, doi: 10.46300/9108.2020.14.14.
- [14] X. Ma, C. Liu "Improved Ant Colony Algorithm for the Split Delivery Vehicle Routing Problem", Applied Sciences, vol. 14, no. 12, doi: 10.3390/app14125090.
- [15] K. Schiff "Ant colony optimisation algorithm for the facility localisation problem", Czasopismo Techniczne, 2018, vol. 1, pp. 103-112, doi: 10.4467/2353737XCT.18.008.7959.
- [16] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.