

## ИЕРАРХИЧЕСКИЙ САМОКОНФИГУРИРУЕМЫЙ АЛГОРИТМ КООПЕРАТИВНОЙ КОЭВОЛЮЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ

О.Е. Семенкина

semenkinaolga@gmail.com

В.В. Становов

vladimirstanovov@yandex.ru

Е.А. Попов

eropov@bmail.ru

СибГУ им. М.Ф. Решетнева, г. Красноярск, Российская Федерация

---

### Аннотация

Для решения задачи составления расписания при оперативном планировании производства предложено использовать иерархический самоконфигурируемый метод кооперативной коэволюции, основанный на комбинаторных и вещественных бионических алгоритмах оптимизации. Комбинаторная оптимизация осуществлена с помощью алгоритма муравьиных колоний и генетического алгоритма, а также их самоконфигурируемых версий. Классические и самоконфигурируемые известные версии алгоритма дифференциальной эволюции, стайного и вещественного генетического алгоритмов использованы для вещественной оптимизации. Для сравнения с классическими комбинаторными алгоритмами приведены алгоритм умных капель и эвристика Лина — Кернигана. Предложена соответствующая иерархическая постановка задачи составления расписания, где на верхнем уровне стоит комбинаторная задача поиска порядка запуска партий, а вложенная задача состоит в поиске приоритетов оборудования для увеличения гибкости постановки задачи при сохранении универсальности подхода. Рассмотрены также три постановки задачи, состоящие в поиске порядка запуска партий, выборе порядка приоритетов операций и поиске вещественных значений приоритетов операций. Кроме того, использована имитационная модель производства, помогающая включить все необходимые нюансы технологического процесса. Показана эффективность применения данной постановки задачи в сравнении с другими постановками и классическими алгоритмами комбинаторной и веществен-

### Ключевые слова

*Задача составления расписания, оперативное планирование производства, метод самоконфигурирования, кооперативная коэволюция*

---

ной оптимизации. Предложенная постановка задачи имеет большие возможности применения на сложных производствах с технологическими процессами, требующими нестандартных методов описания	Поступила 12.05.2023 Принята 26.05.2023 © Автор(ы), 2023
--	--

---

*Работа выполнена в соответствии с государственным заданием Минобрнауки России (№ FEFE-2023-0004)*

**Введение.** Концепция Индустрия 4.0 [1] предоставляет компаниям множество возможностей для повышения эффективности производства и улучшения качества продукции. Одним из важнейших направлений является создание интеллектуальных систем поддержки принятия решений для производственных систем, включая оперативное планирование производства [2]. Для любого типа планирования необходимо выбрать эффективный вариант, т. е. решить задачу оптимизации. Составление оптимального расписания производства — сложная задача и еще не удалось создать универсальную систему планирования, несмотря на множество исследований и разработок СППР для планирования производства.

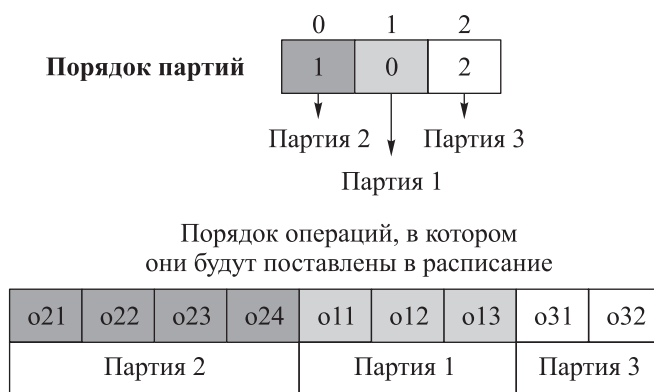
Главной проблемой при создании систем подобного вида остается отсутствие адекватной модели производства, позволяющей решать оптимизационные задачи или имеющей возможность выделения в производственном процессе параметров для оптимизации. Классическая постановка задачи составления расписания, а именно задача планирования проектов с ограниченными ресурсами RCPSP (Resource-Constrained Project Scheduling Problem) может не подходить под особенности производства. Например, могут динамически появляться новые операции, присутствовать точки накопления и перераспределения материала или может возникать необходимость использования нескольких сотрудников и оборудования одновременно в сложном взаимодействии.

В реальных производственных задачах может быть много неопределенностей, например в [3] рассмотрены RCPSP при наличии неопределенности трудозатрат выполнения операций и соответствующая модификация генетического алгоритма. Часто рассматриваются задачи с неопределенной длительностью операции, например в [4] использован подход на основе сценариев, в [5] разработаны несколько эвристик, используемых в подходе на основе робастной оптимизации, в [6] применен метод программной оценки и рассмотрения с экспоненциально распределенной продолжительностью операций. Задача определения последовательности обработки работ и загрузки оборудования так, что минимизируется общее число переключений между оборудованием, рассмотрена в [7].

В настоящей работе авторы предлагают подходить к решению задачи составления расписания в производственной среде комплексно, используя гибкий метод моделирования системы [8] и универсальные постановки задачи. Метод моделирования исходит из того, что производство имеет партийную составляющую, т. е. на вход поступают партии сырья, которые требуется обработать, например, при аффинаже драгоценных металлов. Под партией можно подразумевать заказ на изделие, которое надо произвести по определенному технологическому процессу. Сочетание гибкого метода моделирования и универсальных постановок задачи позволяет учесть все необходимые нюансы конкретного производства в модели, но при этом иметь возможность оценивать различные варианты решений с помощью оптимизационных алгоритмов. Как показано в [9], при оперативном планировании производства могут возникать разнообразные постановки задач, включая варианты с комбинаторными и вещественными переменными, которые хорошо себя показывают, но также имеют ряд ограничений. Универсальная же постановка задачи, применимая к любому производственному процессу, определяемая через поиск порядка запуска партий в производство, может иметь недостаточную гибкость при поиске оптимального решения, так как существенно ограничивает поисковое пространство. В связи с этим предлагается использовать иерархическую постановку задачи, в которой кроме поиска порядка партий определяется приоритет оборудования для большей гибкости метода, но при этом полностью сохраняется универсальность подхода. Необходимость решения иерархической задачи, очевидно, ведет к необходимости использования иерархического метода оптимизации, в качестве которого в настоящей работе предлагается кооперативная коэволюция самонастраивающихся комбинаторных алгоритмов в сочетании с кооперативной коэволюцией самонастраивающихся алгоритмов вещественной оптимизации.

**Постановка задачи.** Задача планирования выполнения проектов с ограниченными ресурсами RCPSP [10] является классической постановкой для планирования производства. В этой задаче нужно назначить время начала каждой работы с учетом приоритетов и ограничений на ресурсы, чтобы минимизировать время выполнения проекта. Эта задача является NP-сложной и даже для одной партии на практике трудно найти допустимое решение. Однако для оперативного планирования производства необходимо быстрое решение, что делает использование имитационной модели обоснованным, так как помогает заранее генерировать исключительно допустимые решения. Модель должна содержать несколько оче-

видных сущностей таких, как оборудование, сотрудники, операции и технологические процессы, а каждой партии соответствует технологический процесс, приведенный в виде дерева операций. Схема построения расписания может быть представлена в виде вектора  $n$ -мерного пространства с помощью отображения нелинейной последовательности операций в строку. Таким образом, в оптимизации можно использовать постановку задачи, позволяющую перевести ее решение в последовательность команд для модели, определяющих порядок выполнения всех операций всех партий. Например, если говорить о перестановке порядка запуска операций партий LO (Lot Order), то решение задачи очевидно трансформируется в последовательность операций всех партий по порядку их появления в решении. Этот процесс приведен на рис. 1.



**Рис. 1.** Пример кодирования решения в задаче выбора порядка партий

Подходы, рассмотренные ранее, — это определение вещественных значений для приоритетов операций AP (Activity Priorities) и перестановка приоритетов операций PO (Priority Order). Такие постановки имеют существенно большую гибкость, так как не обязывают проставлять в расписание все операции одной партии, прежде чем перейти к другой. В то же время они теряют универсальность, поскольку не для всякого технологического процесса можно заранее определить последовательность и даже число необходимых операций. В связи с этим предлагается иерархический подход с увеличением гибкости, но с сохранением универсальности. В этом подходе (рис. 2) на верхнем уровне стоит перестановка партий, но дополнительно есть вложенная задача выбора приоритета оборудования LO-MTP (Lot Order and Machine Tool Priority). Получается, что одновременно решаются комбинаторная задача поиска порядка партий и вещественная задача расстановки приоритетов у оборудования для всех опера-

ций для каждого такого порядка. Приоритет в данном случае означает период времени, на который может позже освободиться некоторое оборудование и при этом все равно быть выбранным для простановки операции. Например, есть два оборудования на одной задаче, при этом первое освобождается в момент времени  $t_1$  и имеет приоритет  $p_1$ , а второе освобождается в момент времени  $t_2$  и имеет приоритет  $p_2$ . Если  $t_1 < t_2$ , то построитель расписания обязательно выберет первое оборудование. Однако если при этом приоритеты таковы, что  $t_1 + p_1 > t_2 + p_2$ , то будет выбрано второе оборудование.

Порядок партий	Партия 1	Партия 2	Партия 3	Партия 4
	1	0	3	2

Приоритет оборудования

Операция 1				Операция 2			Операция 3	
o11	o12	o13	o14	o21	o22	o23	o31	o32
2,55	11,28	1,23	5,51	1,56	1,78	4,71	3,76	8,24

**Рис. 2.** Представление решения с порядком партий и приоритетом оборудования

**Методы.** В соответствии с обзором [11], проведенным в 2019 г., генетический алгоритм, алгоритмы роя частиц, пчелиной колонии и муравьиный являются наиболее популярными для решения задач составления расписания. В настоящей работе используются различные алгоритмы для вещественной и комбинаторной оптимизации, в том числе те, что упомянуты в обзоре, так как они хорошо зарекомендовали себя в своей области. Для вещественной оптимизации используются следующие алгоритмы: генетический GA (Genetic Algorithm) [12], роя частиц PSO (Particle Swarm Optimization) [13] и дифференциальной эволюции DE (Differential Evolution) [14]. Для комбинаторной оптимизации — генетический, муравьиный ACO (Ant Colony Optimization) [15], умных капель IWDs (Intelligent Water Drops) [16] и эвристика Лина — Кернигана LKH (Lin — Kernighan Heuristics) [17].

Серьезным недостатком стохастических алгоритмов, к которым относятся и приведенные ранее алгоритмы, часто является то, что они достаточно капризны и требуют настройки большого числа параметров,

существенно влияющих на эффективность и время работы алгоритма. Природа зависимости эффективности алгоритма от выбора его настроек нетривиальна и не поддается простому анализу. В связи с этим для их настройки обычно требуются экспертные знания, а потому настройка проводится произвольно или в лучшем случае после статистических исследований, так как обоснованный выбор настроек является сложной задачей даже для специалистов в данной области. Использование механизмов самоконфигурирования Sc (Self-Configuring) позволяет решить проблему выбора вариантов операторов и настроек алгоритма [18–20], потому что автоматическая настройка параметров алгоритма позволяет пропустить шаг преднастройки и не тратить вычислительные ресурсы. В связи с этим в работе также рассмотрены самоконфигурируемые версии приведенных алгоритмов.

Такой подход кроме экономии ресурсов дает возможность алгоритму быть более гибким и лучше подстраиваться под задачу на разных этапах ее решения. Таким образом, здесь рассматриваются следующие алгоритмы: самоконфигурируемый ScGA (Self-configuring GA), самоконфигурируемый ScPSO и аналогично ScDE для вещественной оптимизации, а также ScGA и ScACO для комбинаторной оптимизации.

Кроме необходимости выбора настроек и параметров алгоритма остро стоит и вопрос выбора самого алгоритма, поскольку на различных задачах лучше себя показывают разные алгоритмы, следовательно, заранее предсказать лучший выбор невозможно. Хорошим решением этой проблемы является использование метода кооперативной коэволюции CC (Cooperative Coevolution) [21, 22]. Основная идея кооперативной коэволюции заключается в том, чтобы различные алгоритмы работали независимо друг от друга в течение определенного периода времени, а затем оценивались по их эффективности. Каждый алгоритм может использовать свою стратегию для нахождения оптимального решения. После этого периода вычислительные ресурсы перераспределяются таким образом, что лучший алгоритм получает больше ресурсов, чем менее эффективные. Кроме того, алгоритмы обмениваются информацией о лучшем решении, что позволяет им совместно находить наилучшее решение — это называется миграцией.

Метод коэволюции, наиболее похожий на используемый в настоящей работе, в западных источниках носит название «Островная модель» (Island model). Такая коэволюция с обменом данными между разными изолированными популяциями разных алгоритмов показывает хорошую работоспособность и помогает каждой конкретной популяции не впасть

в стагнацию, попав в свой локальный оптимум [23]. Островная модель может состоять из нескольких популяций одного и того же алгоритма, например, наиболее популярный вариант, это генетический алгоритм с различными настройками [24]. В других вариантах реализации используются несколько популяций алгоритмов гармонического поиска [25], искусственной пчелиной колонии [26], дифференциальной эволюции [27], а также множество других вариантов гомогенных островов [28, 29]. В состав островной модели также могут входить и различные алгоритмы. Например, в [23] включены алгоритмы: стайный, поиск волчьей стаи, светлячков, кукушки и летучих мышей. В [30] комбинируются три алгоритма: генетический, стайный и фейерверков. Еще в одном примере можно найти использование следующих алгоритмов: генетического, стайного и социальных пауков [31].

**Обсуждение полученных результатов.** Исследование эффективности приведенных подходов проводилось на шести разработанных задачах, которые были получены на основе опыта изучения ситуаций, возможных на производстве. Для увеличения сложности задач и приближения к реальному положению вещей при разработке задач был использован генератор псевдослучайных величин. В итоге получено 6 задач с общим числом партий от 10 до 60. Итоговая размерность всех задач для различных постановок указана в таблице.

**Размерность задач для различных постановок**

Номер задачи	LO	AP и PO	LO + MTP
1	10	75	10 + 180
2	20	150	20 + 460
3	30	160	30 + 680
4	40	270	40 + 1160
5	50	365	50 + 1260
6	60	355	60 + 1545

Задачи в различных постановках решали с помощью соответствующих алгоритмов. Так, задачи LO и PO решали алгоритмами комбинаторной оптимизации, а именно GA, ScGA, ACO, ScACO, IWDs, LKH и комбинаторным самоконфигурируемым алгоритмом кооперативной коэволюции (ScCCcomb); задача AP — алгоритмами вещественной оптимизации: DE, ScDE, RGA, ScRGA, PSO, ScPSO и вещественным самоконфигурируемым алгоритмом кооперативной коэволюции (ScCCreal). Для задачи LO-MTP использованы только самоконфигурируемые и коэволюционные

алгоритмы, т. е. все сочетания комбинаторных алгоритмов ScGAcomb, ScACOcomb, ScCCcomb с вещественными алгоритмами ScDEreal, ScRGAreal, ScPSOreal и ScCCreal. В том числе сочетание коэволюционных алгоритмов представляет собой иерархический коэволюционный алгоритм, состоящий из ScCCcomb и ScCCreal. Результаты работы классических версий алгоритмов усреднялись по всем рассмотренным вариантам настройки каждого из них.

Все алгоритмы сравнивались на всех шести задачах с фиксированным числом вычислений целевой функции, равным 1 000 000. Результаты экспериментов усреднялись по 50 прогонам, а также были нормализованы следующим образом. Поскольку решения различных задач сильно различаются по абсолютной величине, то для каждой задачи был выбран худший из всех алгоритмов результат, а затем все результаты всех алгоритмов были поделены на этот худший. После этого получено, что все результаты лежат в шкале от 0 до 1. Поскольку решается задача минимизации по критерию общей длины составленного плана, то чем ближе значение к нулю, тем лучше.

Результаты работы перечисленных алгоритмов для различных постановок задачи приведены на рис. 3. Независимо от постановки очевидна тенденция, что самоконфигурируемая версия алгоритма превосходит работу усредненного по настройкам алгоритма. Учитывая то, что при решении реальных практических задач нет возможности тратить вычислительные ресурсы на выбор лучших настроек, использование метода самоконфигурирования является в данном случае предпочтительным. Это обосновано также тем, что полученные в ходе экспериментов результаты показывают, что на различных задачах наиболее эффективными оказываются разные варианты настроек. Более того, на различных задачах лучшие результаты могут показывать разные алгоритмы. Коэволюционный алгоритм показывает себя наравне с входящими в него самоконфигурируемыми алгоритмами, а в случае задачи AP даже немного лучше. Это во многом связано с тем, что кооперативная коэволюция имеет максимальную гибкость при решении задачи и может адаптироваться к каждому этапу как выбором алгоритма, так и выбором его настроек.

Если внимательно рассмотреть результаты на рис. 3, г, то можно отметить, что использование кооперативной коэволюции в иерархической задаче показывает хорошие результаты как отдельно на верхнеуровневой или вложенной задаче, так и на обеих одновременно, т. е. при использовании иерархического кооперативного коэволюционного алгоритма



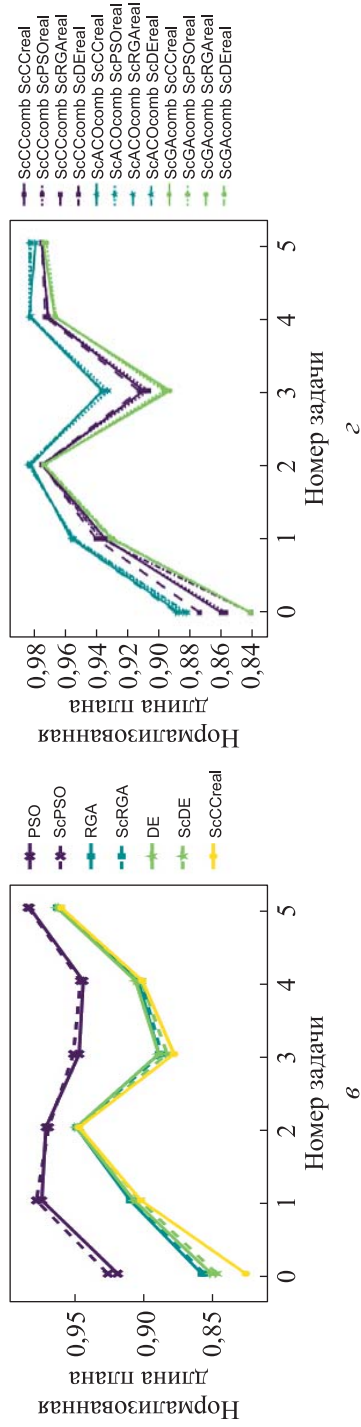
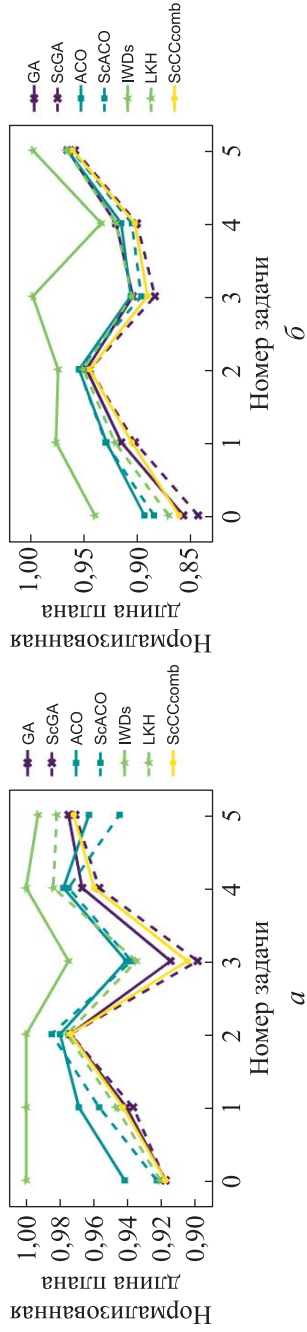
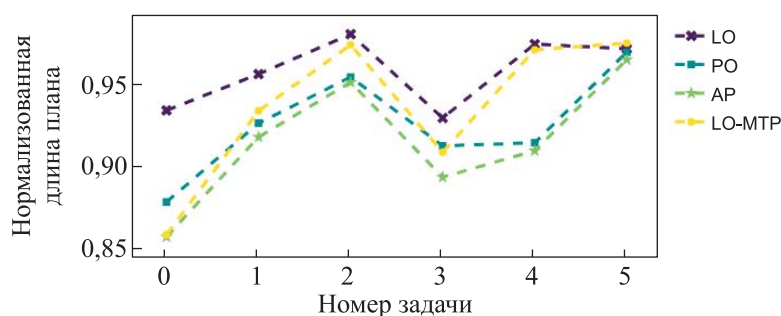


Рис. 3. Сравнение результатов алгоритмов для задач в постановках LO (a), PO (б), AP (в) и LO-MTP (z) соответственно

ScCCcomb+ScCCreal. Таким образом, можно заключить, что иерархический кооперативный коэволюционный алгоритм ScCCcomb+ScCCreal является эффективным для решения задач составления расписания, позволяющим получить результаты, сравнимые с алгоритмами, входящими в его состав. При этом этот алгоритм имеет максимальную гибкость, позволяющую полностью подстраивать решение под задачу в процессе поиска оптимального решения. Данный метод подразумевает также, что при необходимости можно расширять список используемых алгоритмов и таким образом комбинировать их сильные стороны, увеличивая или уменьшая располагаемые вычислительные ресурсы в зависимости от потребности на каждом этапе решения задачи.

Отдельно проведено сравнение различных постановок задачи, а именно выбор порядка партий (LO), перестановка приоритетов операций (PO), вещественные приоритеты операций (AP) и порядок партий со вложенной задачей выбора приоритетов оборудования (LO+MTP). На рис. 4 приведен результат для каждой задачи, усредненный по всем алгоритмам, которые были использованы для конкретной постановки.



**Рис. 4.** Усредненные по всем алгоритмам результаты при различных постановках задачи

Задача в постановке LO имеет существенно меньшую размерность, так как включает в себя исключительно порядок партий, фиксируя их строгую приоритезацию, тогда как задачи в постановках PO и AP имеют размерность в виде суммы числа всех операций во всех партиях, что больше в десятки раз. Задача в постановке LO-MTP в свою очередь имеет еще большую размерность, поскольку включает в себя выбор приоритетов всего оборудования для всех операций. Тем не менее несмотря на существенный рост размерности, постановка LO-MTP в некоторых случаях может давать сравнимые с постановкой LO и даже лучшие результаты за счет гибкости при составлении расписания. Отметим, что очевидно постановка LO существенно менее гибкая по сравнению со всеми осталь-

ными постановками, так как не позволяет рассматривать более разнообразные решения. Тем не менее постановка LO конкурирует с остальными на задачах большей размерности.

Отметим, что хотя постановки PO и AP и показывают в среднем лучшую эффективность, они не во всех ситуациях применимы на практике, также как и классическая постановка RCPSP, где вектором решений являются точки начала выполнения всех операций всех партий. Это связано с тем, что на реальном производстве зачастую технологический процесс производства одной партии не может быть заранее predetermined пооперационно. Например, часто встречаются накопления материала до определенной массы, следовательно, нет возможности заранее до построения расписания предсказать какие именно партии будут объединены, так как это определяется порядком их прихода, а значит порядком запуска в производство. Кроме того, многие производственные процессы могут быть описаны в имитационной модели в виде динамически возникающих событий, предсказать число и время которых заранее также невозможно.

**Заключение.** Рассмотрены различные постановки задачи составления расписания при оперативном планировании производства. Постановка задачи через определение порядка партий, несмотря на худшие в среднем результаты, неплохо показывает себя на задачах большой размерности и может сравниться с более гибкими постановками. Кроме того, эта постановка имеет большие возможности применения на сложных производствах с технологическими процессами, требующими нестандартных методов описания. Иерархическая постановка задачи с использованием выбора порядка партий и подбором приоритетов оборудования улучшает результаты, сохраняя гибкость и универсальность использования на реальных производствах.

Кооперативная коэволюция самоконфигурируемых бионических алгоритмов показывает одни из лучших результатов и, кроме того, может даже показывать себя лучше, чем ее отдельные компоненты. Такая интеграция двух подходов существенно расширяет возможности применения оптимизационных алгоритмов данного класса на практике, так как не требует привлечения экспертов в области бионических алгоритмов оптимизации. Использование иерархического самоконфигурируемого метода кооперативной коэволюции на основе комбинаторных и вещественных алгоритмов оптимизации для решения иерархической задачи составления расписания показывает хорошие результаты.

## ЛИТЕРАТУРА

- [1] Oztemel E., Gursev S. Literature review of Industry 4.0 and related technologies. *J. Intell. Manuf.*, 2020, vol. 31, no. 4, pp. 127–182.  
DOI: <https://doi.org/10.1007/s10845-018-1433-8>
- [2] Li Y., Goga K., Tadei R., et al. Production scheduling in Industry 4.0. *CISIS 2020*. Cham, Springer Nature, 2020, pp. 355–364.  
DOI: [https://doi.org/10.1007/978-3-030-50454-0\\_34](https://doi.org/10.1007/978-3-030-50454-0_34)
- [3] Шевляков А.О., Матвеев М.Г. Решение RCPSP при нечетких трудозатратах выполнения операций. *Вестник ВГУ. Сер. Системный анализ и информационные технологии*, 2015, № 4, с. 121–125.
- [4] Zaman F., Elsayed S.M., Sarker R.A., et al. Scenario-based solution approach for uncertain resource constrained scheduling problems. *IEEE CEC*, 2018.  
DOI: <https://doi.org/10.1109/CEC.2018.8477756>
- [5] Chakraborty R.K., Sarker R.A., Essam D.L. Resource constrained project scheduling with uncertain activity durations. *Comput. Ind. Eng.*, 2017, vol. 112, pp. 537–550.  
DOI: <https://doi.org/10.1016/j.cie.2016.12.040>
- [6] Creemers S. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *J. Sched.*, 2015, vol. 18, no. 3, pp. 263–273.  
DOI: <https://doi.org/10.1007/s10951-015-0421-5>
- [7] Calmels D. The job sequencing and tool switching problem: state-of-the-art literature review, classification, and trends. *Int. J. Prod. Res.*, 2019, vol. 57, no. 15-16, pp. 5005–5025. DOI: <https://doi.org/10.1080/00207543.2018.1505057>
- [8] Семенкина О.Е., Рыжиков И.С., Липинский Л.В. и др. Метод моделирования производственных систем для оперативного планирования производства. *Системы управления и информационные технологии*, 2021, № 3, с. 17–24.
- [9] Семенкина О.Е. Кооперативная коэволюция самоконфигурируемых бионических алгоритмов в задачах составления расписания. *Системы управления и информационные технологии*, 2022, № 1, с. 63–68.
- [10] Аничкин А.С., Семенов В.А. Современные модели и методы теории расписаний. *Труды Института системного программирования РАН*, 2014, т. 26, № 3, с. 5–50. DOI: [https://doi.org/10.15514/ISPRAS-2014-26\(3\)-1](https://doi.org/10.15514/ISPRAS-2014-26(3)-1)
- [11] Liu Y., Wang L., Wang X., et al. Scheduling in cloud manufacturing: state-of-the-art and research challenges. *Int. J. Prod. Res.*, 2019, vol. 57, no. 15-16, pp. 4854–4879.  
DOI: <https://doi.org/10.1080/00207543.2018.1449978>
- [12] Bäck T., Fogel D.B., Michalewicz Z., eds. *Handbook of evolutionary computation*. Oxford, Oxford University Press, 1997.
- [13] Bansal J.C. Particle swarm optimization. *Evolutionary and swarm intelligence algorithms*. Cham, Springer Nature, 2019, pp. 11–23.  
DOI: [https://doi.org/10.1007/978-3-319-91341-4\\_2](https://doi.org/10.1007/978-3-319-91341-4_2)
- [14] Lampinen J., Storn R. Differential evolution. *New optimization techniques in engineering*. Berlin, Springer Heidelberg, 2004, pp. 123–166.  
DOI: [https://doi.org/10.1007/978-3-540-39930-8\\_6](https://doi.org/10.1007/978-3-540-39930-8_6)

- [15] Dorigo M., Stutzle T. Ant colony optimization. Overview and recent advances. *Handbook of metaheuristics*. Cham, Springer Nature, 2010, pp. 227–263.  
DOI: [https://doi.org/10.1007/978-1-4419-1665-5\\_8](https://doi.org/10.1007/978-1-4419-1665-5_8)
- [16] Shah-Hosseini H. Problem solving by intelligent water drops. *Proc. IEEE Congress on Evolutionary Computation*, 2007, pp. 3226–3231.  
DOI: <https://doi.org/10.1109/CEC.2007.4424885>
- [17] Lin S., Kernigan B.W. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 1973, vol. 21, no. 2, pp. 498–516.
- [18] Semenkin E., Semenkina M. Self-configuring genetic algorithm with modified uniform crossover operator. *ICSI 2012*. Berlin, Springer Heidelberg, 2012, pp. 414–421.  
DOI: [https://doi.org/10.1007/978-3-642-30976-2\\_50](https://doi.org/10.1007/978-3-642-30976-2_50)
- [19] Meyer-Nieberg S., Beyer H.-G. Self-adaptation in evolutionary algorithms. *Parameter setting in evolutionary algorithm*. Cham, Springer Nature, 2007, pp. 47–75.  
DOI: [https://doi.org/10.1007/978-3-540-69432-8\\_3](https://doi.org/10.1007/978-3-540-69432-8_3)
- [20] Семенкин Е.С., Семенкина М.Е. Проектирование ансамблей интеллектуальных информационных технологий самоконфигурируемым алгоритмом генетического программирования. *Вестник СибГАУ*, 2012, № 4, с. 89–96.
- [21] Аплеснин С.С., Жуков В.Г., Попов Е.А. и др. Исследование коэволюционного алгоритма генетического программирования и его применение в задаче моделирования фазовых границ магнитного состояния кристалла. *Вестник СибГАУ*, 2011, № 5, с. 9–14.
- [22] Емельянова М.Н., Семенкин Е.С. Исследование эффективности коэволюционного алгоритма. *Вестник СибГАУ*, 2004, № 6, с. 28–34.
- [23] Ахмедова Ш.А., Становов В.В., Семенкин Е.С. Кооперация бионического и эволюционного алгоритмов для задач проектирования искусственных нейронных сетей. *Журнал СФУ. Математика и физика*, 2018, т. 11, № 2, с. 148–158.
- [24] Brester C., Ryzhikov I., Semenkin E., et al. On island model performance for cooperative real-valued multi-objective genetic algorithms. *ICSI 2018*. Cham, Springer Nature, 2018, pp. 210–219. DOI: [https://doi.org/10.1007/978-3-319-93815-8\\_21](https://doi.org/10.1007/978-3-319-93815-8_21)
- [25] Al-Betar M.A. Island-based harmony search algorithm for non-convex economic load dispatch problems. *J. Electr. Eng. Technol.*, 2021, vol. 16, no. 4, pp. 1985–2015.  
DOI: <https://doi.org/10.1007/s42835-021-00758-w>
- [26] Awadallah M.A., Al-Betar M.A., Bolaji A.L., et al. Island artificial bee colony for global optimization. *Soft Comput.*, 2020, vol. 24, no. 17, pp. 13461–13487.  
DOI: <https://doi.org/10.1007/s00500-020-04760-8>
- [27] Duarte G.R., de Lima B.S.L.P. Differential evolution variants combined in a hybrid dynamic island model. *IEEE CEC*, 2020.  
DOI: <https://doi.org/10.1109/CEC48606.2020.9185579>
- [28] Abed-Alguni B.H., Paul D. Island-based Cuckoo Search with elite opposition-based learning and multiple mutation methods for solving optimization problems. *Soft Comput.*, 2022, vol. 26, no. 7, pp. 3293–3312.  
DOI: <https://doi.org/10.1007/s00500-021-06665-6>

[29] Alvarez-Mamani E., Enciso-Rodas L., Ayala-Rincón M., et al. Parallel social spider optimization algorithms with island model for the clustering problem. *SIMBig 2020*. Cham, Springer Nature, 2020, pp. 122–138.

DOI: [https://doi.org/10.1007/978-3-030-76228-5\\_9](https://doi.org/10.1007/978-3-030-76228-5_9)

[30] Li J., Gonsalves T. Parallel hybrid island metaheuristic algorithm. *IEEE Access*, 2022, vol. 10, pp. 42268–42286. DOI: <https://doi.org/10.1109/ACCESS.2022.3165830>

[31] Da Silveira L.A., Soncco-Álvarez J.L., de Lima T.A., et al. Behavior of bioinspired algorithms in parallel island models. *IEEE CEC*, 2020.

DOI: <https://doi.org/10.1109/CEC48606.2020.9185732>

**Семенкина Ольга Евгеньевна** — инженер НИЛ Института информатики и телекоммуникаций СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т им. газеты Красноярский рабочий, д. 31).

**Становов Владимир Вадимович** — канд. техн. наук, доцент кафедры высшей математики СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т им. газеты Красноярский рабочий, д. 31).

**Попов Евгений Александрович** — д-р физ.-мат. наук, профессор кафедры системного анализа и исследования операций СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т им. газеты Красноярский рабочий, д. 31).

**Просьба ссылаться на эту статью следующим образом:**

Семенкина О.Е., Становов В.В., Попов Е.А. Иерархический самоконфигурируемый алгоритм кооперативной коэволюции для решения задачи составления расписания. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2023, № 4 (145), с. 131–148. DOI: <https://doi.org/10.18698/0236-3933-2023-4-131-148>

**COOPERATIVE COEVOLUTION HIERARCHICAL  
SELF-CONFIGURING ALGORITHM FOR SOLVING  
THE SCHEDULING PROBLEM**

**O.E. Semenkina**

[semenkinaolga@gmail.com](mailto:semenkinaolga@gmail.com)

**V.V. Stanovov**

[vladimirstanovov@yandex.ru](mailto:vladimirstanovov@yandex.ru)

**E.A. Popov**

[epopov@bmail.ru](mailto:epopov@bmail.ru)

**Reshetnev Siberian State University of Science and Technology,  
Krasnoyarsk, Russian Federation**

**Abstract**

To solve the problem of scheduling during operational production planning, the paper proposes to introduce a cooperative coevolution hierarchical self-configuring method based on the combinatorial and material bionic

**Keywords**

*Scheduling problem, operational production planning, self-configuration method, cooperative coevolution*

optimization algorithms. Combinatorial optimization was performed using the ant colony algorithm and the genetic algorithm, as well as their self-configuring versions. Similarly, classical and self-configuring well-known versions of the differential evolution algorithm and of the flocking and material genetic algorithms were used in material optimization. For comparison with the classical combinatorial algorithms, the smart drop algorithm and the Lin — Kernighan heuristic were included. A corresponding hierarchical formulation of the scheduling problem was proposed, where at the top level, the combinatorial problem of finding the batch launching order was set, and the attached task consisted in finding the equipment priorities to increase the problem formulation flexibility while maintaining the approach universality. Three problem formulations were also considered. They included finding the batch launching order, selecting the operation priority order and finding material values of the operation priorities. In addition, production simulation model was introduced to include necessary nuances of the technological process. Effectiveness of using this methodology was demonstrated in comparison with other problem formulations and classical algorithms of combinatorial and material optimization. The proposed formulation of the problem provides significant possibilities for application in complex industries with technological processes requiring non-standard methods of description

Received 12.05.2023

Accepted 26.05.2023

© Author(s), 2023

---

*The work was supported by the Ministry of Science and Higher Education of the Russian Federation (project no. FEFE-202300-0004)*

## REFERENCES

- [1] Oztemel E., Gursev S. Literature review of Industry 4.0 and related technologies. *J. Intell. Manuf.*, 2020, vol. 31, no. 4, pp. 127–182.  
DOI: <https://doi.org/10.1007/s10845-018-1433-8>
- [2] Li Y., Goga K., Tadei R., et al. Production scheduling in Industry 4.0. *CISIS 2020*. Cham, Springer Nature, 2020, pp. 355–364.  
DOI: [https://doi.org/10.1007/978-3-030-50454-0\\_34](https://doi.org/10.1007/978-3-030-50454-0_34)
- [3] Shevlyakov A.O., Matveev M.G. Solving RCPSP with uncertain duration times of activities. *Vestnik VGU. Ser. Sistemnyy analiz i informatsionnye tekhnologii* [Proceedings of Voronezh State University. Ser. Systems Analysis and Information Technologies], 2015, no. 4, pp. 121–125 (in Russ.).

- [4] Zaman F., Elsayed S.M., Sarker R.A., et al. Scenario-based solution approach for uncertain resource constrained scheduling problems. *IEEE CEC*, 2018.  
DOI: <https://doi.org/10.1109/CEC.2018.8477756>
- [5] Chakraborty R.K., Sarker R.A., Essam D.L. Resource constrained project scheduling with uncertain activity durations. *Comput. Ind. Eng.*, 2017, vol. 112, pp. 537–550.  
DOI: <https://doi.org/10.1016/j.cie.2016.12.040>
- [6] Creemers S. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *J. Sched.*, 2015, vol. 18, no. 3, pp. 263–273.  
DOI: <https://doi.org/10.1007/s10951-015-0421-5>
- [7] Calmels D. The job sequencing and tool switching problem: state-of-the-art literature review, classification, and trends. *Int. J. Prod. Res.*, 2019, vol. 57, no. 15-16, pp. 5005–5025. DOI: <https://doi.org/10.1080/00207543.2018.1505057>
- [8] Semenkin O.E., Ryzhikov I.S., Lipinskiy L.V., et al. A method of production system modeling for operational planning. *Sistemy upravleniya i informatsionnye tekhnologii*, 2021, no. 3, pp. 17–24 (in Russ.).
- [9] Semenkin O.E. Cooperative co-evolution of self-configuring bio-inspired algorithms for the scheduling problem. *Sistemy upravleniya i informatsionnye tekhnologii*, 2022, no. 1, pp. 63–68 (in Russ.).
- [10] Anichkin A.S., Semenov V.A. A survey of emerging models and methods of scheduling. *Trudy Instituta sistemnogo programirovaniya RAN* [Proceedings of ISP RAS], 2014, vol. 26, no. 3, pp. 5–50 (in Russ.).  
DOI: [https://doi.org/10.15514/ISPRAS-2014-26\(3\)-1](https://doi.org/10.15514/ISPRAS-2014-26(3)-1)
- [11] Liu Y., Wang L., Wang X., et al. Scheduling in cloud manufacturing: state-of-the-art and research challenges. *Int. J. Prod. Res.*, 2019, vol. 57, no. 15-16, pp. 4854–4879.  
DOI: <https://doi.org/10.1080/00207543.2018.1449978>
- [12] Bäck T., Fogel D.B., Michalewicz Z., eds. Handbook of evolutionary computation. Oxford, Oxford University Press, 1997.
- [13] Bansal J.C. Particle swarm optimization. *Evolutionary and swarm intelligence algorithms*. Cham, Springer Nature, 2019, pp. 11–23.  
DOI: [https://doi.org/10.1007/978-3-319-91341-4\\_2](https://doi.org/10.1007/978-3-319-91341-4_2)
- [14] Lampinen J., Storn R. Differential evolution. *New optimization techniques in engineering*. Berlin, Springer Heidelberg, 2004, pp. 123–166.  
DOI: [https://doi.org/10.1007/978-3-540-39930-8\\_6](https://doi.org/10.1007/978-3-540-39930-8_6)
- [15] Dorigo M., Stutzle T. Ant colony optimization. Overview and recent advances. *Handbook of metaheuristics*. Cham, Springer Nature, 2010, pp. 227–263.  
DOI: [https://doi.org/10.1007/978-1-4419-1665-5\\_8](https://doi.org/10.1007/978-1-4419-1665-5_8)
- [16] Shah-Hosseini H. Problem solving by intelligent water drops. *Proc. IEEE Congress on Evolutionary Computation*, 2007, pp. 3226–3231.  
DOI: <https://doi.org/10.1109/CEC.2007.4424885>
- [17] Lin S., Kernigan B.W. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 1973, vol. 21, no. 2, pp. 498–516.



- [18] Semenkin E., Semenkina M. Self-configuring genetic algorithm with modified uniform crossover operator. *ICSI 2012*. Berlin, Springer Heidelberg, 2012, pp. 414–421. DOI: [https://doi.org/10.1007/978-3-642-30976-2\\_50](https://doi.org/10.1007/978-3-642-30976-2_50)
- [19] Meyer-Nieberg S., Beyer H.-G. Self-adaptation in evolutionary algorithms. *Parameter setting in evolutionary algorithm*. Cham, Springer Nature, 2007, pp. 47–75. DOI: [https://doi.org/10.1007/978-3-540-69432-8\\_3](https://doi.org/10.1007/978-3-540-69432-8_3)
- [20] Semenkin E.S., Semenkina M.E. Integration of intelligent information technologies ensembles with self-configuring genetic programming algorithm. *Vestnik SibGAU [Vestnik SibSAU]*, 2012, no. 4, pp. 89–96 (in Russ.).
- [21] Aplesnin S.S., Zhukov V.G., Popov E.A., et al. Research of coevolution genetic programming algorithm and its application in the problem of model analysis of phase boundaries of magnetic state of a crystal. *Vestnik SibGAU [Vestnik SibSAU]*, 2011, no. 5, pp. 9–14 (in Russ.).
- [22] Emelyanova M.N., Semenkin E.S. Effectiveness investigation of coevolutionary genetic algorithm for constrained optimization problems. *Vestnik SibGAU [Vestnik SibSAU]*, 2004, no. 6, pp. 28–34 (in Russ.).
- [23] Akhmedova Sh.A., Stanovov V.V., Semenkin E.S. Cooperation of bio-inspired and evolutionary algorithms for neural network design. *Zhurnal SFU. Matematika i fizika [Journal of Siberian Federal University. Mathematics & Physics]*, 2018, vol. 11, no. 2, pp. 148–158 (in Russ.).
- [24] Brester C., Ryzhikov I., Semenkin E., et al. On island model performance for cooperative real-valued multi-objective genetic algorithms. *ICSI 2018*. Cham, Springer Nature, 2018, pp. 210–219. DOI: [https://doi.org/10.1007/978-3-319-93815-8\\_21](https://doi.org/10.1007/978-3-319-93815-8_21)
- [25] Al-Betar M.A. Island-based harmony search algorithm for non-convex economic load dispatch problems. *J. Electr. Eng. Technol.*, 2021, vol. 16, no. 4, pp. 1985–2015. DOI: <https://doi.org/10.1007/s42835-021-00758-w>
- [26] Awadallah M.A., Al-Betar M.A., Bolaji A.L., et al. Island artificial bee colony for global optimization. *Soft Comput.*, 2020, vol. 24, no. 17, pp. 13461–13487. DOI: <https://doi.org/10.1007/s00500-020-04760-8>
- [27] Duarte G.R., de Lima B.S.L.P. Differential evolution variants combined in a hybrid dynamic island model. *IEEE CEC*, 2020. DOI: <https://doi.org/10.1109/CEC48606.2020.9185579>
- [28] Abed-Alguni B.H., Paul D. Island-based Cuckoo Search with elite opposition-based learning and multiple mutation methods for solving optimization problems. *Soft Comput.*, 2022, vol. 26, no. 7, pp. 3293–3312. DOI: <https://doi.org/10.1007/s00500-021-06665-6>
- [29] Alvarez-Mamani E., Enciso-Rodas L., Ayala-Rincón M., et al. Parallel social spider optimization algorithms with island model for the clustering problem. *SIMBig 2020*. Cham, Springer Nature, 2020, pp. 122–138. DOI: [https://doi.org/10.1007/978-3-030-76228-5\\_9](https://doi.org/10.1007/978-3-030-76228-5_9)

[30] Li J., Gonsalves T. Parallel hybrid island metaheuristic algorithm. *IEEE Access*, 2022, vol. 10, pp. 42268–42286. DOI: <https://doi.org/10.1109/ACCESS.2022.3165830>

[31] Da Silveira L.A., Soncco-Álvarez J.L., de Lima T.A., et al. Behavior of bioinspired algorithms in parallel island models. *IEEE CEC*, 2020. DOI: <https://doi.org/10.1109/CEC48606.2020.9185732>

**Semenkina O.E.** — Engineer, Research Laboratory, Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology (Imeni gazety Krasnoyarskiy rabochiy prospekt 31, Krasnoyarsk, 660037 Russian Federation).

**Stanovov V.V.** — Cand. Sc. (Eng.), Assoc. Professor, Department of Higher Mathematics, Reshetnev Siberian State University of Science and Technology (Imeni gazety Krasnoyarskiy rabochiy prospekt 31, Krasnoyarsk, 660037 Russian Federation).

**Popov E.A.** — Dr. Sc. (Phys.-Math.), Professor, Department of Systems Analysis and Operations Research, Reshetnev Siberian State University of Science and Technology (Imeni gazety Krasnoyarskiy rabochiy prospekt 31, Krasnoyarsk, 660037 Russian Federation).

**Please cite this article in English as:**

Semenkina O.E., Stanovov V.V., Popov E.A. Cooperative coevolution hierarchical self-configuring algorithm for solving the scheduling problem. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2023, no. 4 (145), pp. 131–148 (in Russ.). DOI: <https://doi.org/10.18698/0236-3933-2023-4-131-148>