

УДК 519.8

К.Г. Розенков, С.А. Федосеев, А.А. Вдовиченко

Пермский национальный исследовательский
политехнический университет, Пермь, Россия

МЕТОД НАСТРОЙКИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ЗАДАЧИ ПЛАНИРОВАНИЯ ПОТОЧНОГО ПРОИЗВОДСТВА

Предложен подход к настройке генетического алгоритма, используемого при решении задачи планирования поточного производства. Для этого введены критерии результативности и эффективности генетического алгоритма. Результативность характеризует способность генетического алгоритма находить глобальный экстремум в поставленной оптимизационной задаче. Эффективность характеризует скорость, с которой генетический алгоритм позволяет решать оптимизационную задачу. Настройка генетического алгоритма осуществляется путем выбора лучшей комбинации из нескольких вариантов генетических операторов. Рассмотрен пример решения задачи планирования для поточной линии по розливу напитков. На данном примере показана реализуемость и практическая значимость предложенного подхода к настройке генетических алгоритмов. В результате проведения значительного количества численных экспериментов удалось найти аргументы в пользу подтверждения гипотезы масштабируемости настроенного генетического алгоритма.

Ключевые слова: генетический алгоритм, генетический оператор, настройка генетического алгоритма, масштабируемость генетического алгоритма, планирование производства.

K.G. Rozenkov, S.A. Fedoseev, A.A. Vdovichenko

Perm National Research Polytechnic University, Perm, Russian Federation

METHOD OF CUSTOMIZING GENETIC ALGORITHM FOR LINE PRODUCTION PLANNING PROBLEM

In this paper an approach for customizing genetic algorithm for solving problem of planning line production is proposed. For this purpose, the criteria of efficiency and effectiveness of the genetic algorithm are introduced. Efficiency characterizes the ability of a genetic algorithm to find the global extremum in a given optimization problem. Effectiveness characterizes the rate at which the genetic algorithm allows to solve the optimization problem. Customizing genetic algorithm is done by selecting the best combination of several variants of genetic operators. For the production line bottling the example of the planning problem solution was considered. This example showed the feasibility and practical relevance of the proposed approach to the customizing of genetic algorithms. As a result of the significant number of numerical experiments were found arguments supporting the hypothesis of scalability customized genetic algorithm.

Keywords: genetic algorithm, genetic operator, genetic algorithms customizing, genetic algorithm scalability, production planning.

Введение

Классической для планирования производства является задача *flow-shop-problem*, сформулированная Джонсоном [1, 2]. Она заключается в построении последовательности партий, обеспечивающей минимальное время завершения обработки всех этих партий. Существенная особенность заключается в том, что состав и последовательность операций для обработки каждой партии одинаковы.

В данной работе рассматривается частный случай задачи *flow-shop-problem*, а именно – задача планирования для многопредметных поточных линий [3], которые активно используются в металлургической, пищевой, химической и других отраслях промышленности [4, 5].

На многопредметных поточных линиях изделия различных наименований (типоразмеров) обрабатываются или собираются последовательно чередующимися во времени партиями. В каждый момент времени на такой линии изготавливается изделие только одного наименования. Для перехода к следующей партии изделий другого наименования требуется переналадка оборудования, зависящая от снимаемой и запускаемой партий и в некоторых случаях занимающая значительное время. Тогда для минимизации длительности производственного цикла необходимо отыскать такую последовательность запуска партий в обработку, при которой суммарное время переналадки будет наименьшим.

1. Пример поточной линии

Примером многопредметной поточной линии служит линия по розливу напитков [6]. На вход этой линии подается упаковка и напиток, а на выходе получается расфасованный напиток в упаковке. При смене напитка предусмотрена технологическая мойка линии. Причем в зависимости от предыдущего и последующего напитка мойка может быть короткой (40 мин) или длинной (300 мин). Например, если после яблочного сока разливается грейпфрутовый, то линия подвергается короткой мойке. Если, наоборот, после грейпфрутового сока разливается яблочный, то линия подвергается длинной мойке.

В табл. 1 представлен фрагмент матрицы технологических моек. Единица в ячейке означает, что после розлива сока, указанного в строке, выполняется короткая мойка для следующего сока, указанного в столбце. Аналогично пустая ячейка означает длинную мойку.

Таблица 1

Фрагмент матрицы технологических моек

Вкус	Абрикос	Апельсин	Виноград	Грейпфрут	Мульти-фрукт	Томат	Яблоко
Абрикос	1	1			1		
Апельсин		1		1	1		
Виноград			1	1	1		
Грейпфрут				1	1		
Мультифрукт					1		
Томат						1	
Яблоко	1	1	1	1	1		1

Таким образом, задача составления плана розлива напитков для данной поточной линии сводится к нахождению такой последовательности напитков, которая позволит разлить все требуемые партии напитков с минимальными затратами времени на мойку линии. Иными словами, целевую функцию для задачи оптимизации плана розлива напитков можно записать в следующем виде:

$$F = \sum_{i=2}^q w_{i-1,i} \rightarrow \min, \quad (1)$$

где q – количество разливаемых напитков; $w_{i-1,i}$ – время мойки между предыдущим $i - 1$ -м и следующим i -м напитком.

Данная задача в принципе может быть решена полным перебором всех вариантов последовательностей напитков. Однако если учесть, что в реальности план розлива составляется для порядка трех десятков напитков, то для составления плана розлива придется перебрать $30! \approx 2,65 \cdot 10^{32}$ вариантов, что потребует очень большого количества времени, недопустимого для оперативного планирования производства.

Подобные задачи относятся к классу NP-трудных [7]. На практике такие задачи обычно решаются с использованием различных эвристических алгоритмов, которые позволяют находить субоптимальные, а иногда и оптимальные решения, за приемлемое время. Одной из таких эвристик являются генетические алгоритмы (ГА) [8].

2. Генетические алгоритмы

К достоинствам ГА можно отнести отсутствие направленности на строго определенный класс задач. Недостатком же можно считать то,

что ГА, как и любая другая эвристика, не способны гарантировать нахождение абсолютного экстремума.

Основой ГА являются генетические операторы [9]. Ниже кратко описаны их функциональные особенности.

Оператор репродукции (селекции) (ОР) – это оператор, посредством которого хромосомы (альтернативные решения, в рассматриваемом случае различные варианты последовательности партий), имеющие более высокое значение целевой функции, получают большую возможность для воспроизводства (репродукции) потомков. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков.

В алгоритме в качестве ОР используется комбинированный оператор селекции на основе турнирного и элитного ОР. Суть его заключается в следующем:

- а) из популяции в n особей выбирается s лучших;
- б) эти s лучших решений сохраняются;
- в) над всеми n особями производятся манипуляции с помощью тройки генетических операторов (мутация, инверсия, кроссинговер);
- г) из получившегося набора и сохраненных s особей выбирается n лучших особей, остальные отбрасываются;
- д) переход к шагу a .

Оператор мутации (ОМ) – это языковая конструкция, позволяющая на основе преобразования родительской хромосомы (или ее части) создавать хромосому потомка.

В алгоритме предполагается использование трех различных вариантов ОМ:

1. ОМ, который рассматривает каждую последовательность партий, закодированную в бинарном виде, как одну хромосому, а в качестве гена рассматривает бит. Таким образом этот оператор меняет значения k случайно выбранных битов на противоположные.

2. ОМ, аналогичный варианту 1, но геном является не отдельный бит, а элемент последовательности партий, закодированный в бинарном виде. Таким образом этот оператор в случайно выбранных m генах меняет значения всех битов на противоположные.

3. ОМ, предполагающий первоначальное разделение хромосомы на гены и последующее случайное применение варианта 1 к каждому гену.

В дальнейшем для перечисленных вариантов генетических операторов будем использовать обозначения М1, М2, М3.

Оператор инверсии (ОИ) – это языковая конструкция, позволяющая на основе инвертирования родительской хромосомы (или ее части) создавать хромосому потомка. В данной работе при его реализации случайным образом определяются две точки разреза хромосомы, между которыми гены инвертируются, т.е. меняются местами относительно центра вырезанного (инвертируемого) участка хромосомы.

В алгоритме предполагается использование трех различных вариантов ОИ:

1. ОИ, который рассматривает каждую последовательность партий, закодированную в бинарном виде, как одну хромосому, а в качестве гена рассматривает бит. Таким образом, этот оператор меняет местами некоторые k битов, оказавшихся между точками разреза.

2. ОИ, аналогичный варианту 1, но геном является не отдельный бит, а элемент последовательности партий, закодированный в бинарном виде. Таким образом, этот оператор меняет местами некоторые m генов, оказавшихся между точками разреза.

3. ОИ, предполагающий первоначальное разделение хромосомы на гены и последующее случайное применение варианта 1 к каждому гену.

В дальнейшем для перечисленных вариантов генетических операторов будем использовать обозначения И1, И2, И3.

Оператор кроссинговера (ОК) – это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков. Существует значительное количество типов ОК, в данной работе используется так называемый масочный тип ОК.

В алгоритме предполагается использование трех различных вариантов ОК:

1. ОК, который рассматривает каждую последовательность партий, закодированную в бинарном виде, как одну хромосому, а в качестве гена рассматривает бит. Таким образом, этот оператор меняет местами некоторые k битов между хромосомами родителей.

2. ОК, аналогичный варианту 1, но геном является не отдельный бит, а элемент последовательности партий, закодированный в бинарном виде. Таким образом, этот оператор меняет местами некоторые m генов между хромосомами родителей.

3. ОК, предполагающий первоначальное разделение хромосом родителей на гены и последующее случайное применение варианта 1 к некоторым парам генов родителей.

В дальнейшем для перечисленных вариантов генетических операторов будем использовать обозначения К1, К2, К3.

Следует отметить, что благодаря своей гибкости ГА, с одной стороны, обладают широким спектром применимости, но с другой – как правило, нуждаются в специальной настройке применительно к различным классам задач.

3. Критерии результативности и эффективности генетического алгоритма

Необходимым элементом ГА является условие его завершения в каждом численном эксперименте. Здесь и далее под численным экспериментом понимается однократное решение задачи оптимизации. В данной работе использованы два условия завершения ГА:

1) завершение по достижении максимального наперед заданного числа генераций популяции в каждом численном эксперименте, например 1000;

2) завершение при отсутствии улучшения, например более чем на 5 %, средневзвешенного значения целевой функции лучших особей в течении фиксированного числа генераций популяции, например 20.

В данной работе под результативностью ГА понимается способность алгоритма находить оптимальное решение (абсолютный экстремум) или субоптимальные решения (локальные экстремумы) задачи оптимизации плана производства. Следует отметить, что ГА не относится к классу полных (точных) алгоритмов, т.е. не гарантирует нахождение оптимального решения задачи.

Если оптимальное решение задачи является заранее известным, то в качестве значения критерия результативности может быть использована частота нахождения этого решения при проведении заданного количества численных экспериментов.

Если оптимальное решение задачи неизвестно, то в качестве оценки критерия результативности может быть использовано средневзвешенное значение целевой функции

$$\bar{F} = \frac{1}{N} \sum_{i=1}^k n_i F_i, \quad (2)$$

где N – заданное количество численных экспериментов; k – количество найденных оптимальных и субоптимальных значений целевой функции; F_i – оптимальное или субоптимальное значение целевой функции; n_i – частота нахождения значения целевой функции F_i , $\sum_{i=1}^k n_i = N$.

В данной работе под эффективностью ГА понимается скорость, с которой алгоритм находит оптимальное или субоптимальное решение.

В качестве значения критерия эффективности может быть использовано средневзвешенное значение времени проведения численных экспериментов при заданном их количестве. В связи с тем, что время одного численного эксперимента достаточно мало, колеблется в пределах 1 с, критерий эффективности ГА можно считать второстепенным.

Следует отметить, что критерии эффективности и результативности в общем случае могут противоречить друг другу. В частности, численные эксперименты могут довольно быстро завершаться (ГА функционирует эффективно), но в окрестности не абсолютного, а локального экстремума (ГА функционирует нерезультативно).

4. Настройка генетического алгоритма

Настройка ГА заключается в подборе такого сочетания генетических операторов, которые для данного класса задач обеспечивают достижение требуемых значений критериев эффективности и результативности ГА.

В данной работе настройка ГА осуществлялась на основе генетических операторов М1, М2, М3, И1, И2, И3, К1, К2, К3. В силу их незначительного количества возможен полный перебор всех вариантов сочетаний $M_i I_j K_k$ путем осуществления некоторого количества численных экспериментов.

Результат численных экспериментов при измерении критерия результативности как частоты нахождения оптимального решения при проведении 10 000 численных экспериментов для 30 партий напитков приведен на рисунке, где F – значение целевой функции, измеренное в часах; n – частота нахождения значения целевой функции. Лучшим по критерию результативности стало сочетание генетических операторов М3И2К2. В частности, из рисунка видно, что сочетание генетических операторов М3И2К2 обеспечивает лучшую результативность ГА, чем сочетание ге-

нетических операторов М1И2К2, так как в первом случае гораздо чаще были найдены существенно меньшие значения целевой функции.

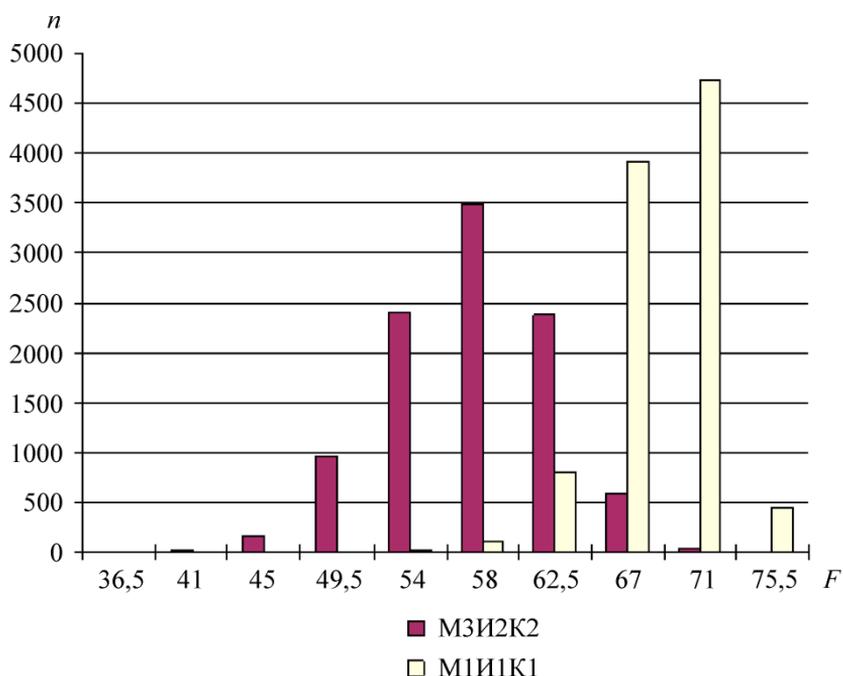


Рис. Сравнение сочетаний генетических операторов по критерию результативности (авторские результаты)

5. Гипотеза масштабируемости настроенного генетического алгоритма

В табл. 2 частично отражены результаты численных экспериментов при измерении критерия результативности по формуле (2) в часах. Наглядно видно, что лидерами по критерию результативности поочередно являются два сочетания генетических операторов МЗИ2К2 и М2И2К2. В действительности эти сочетания генетических операторов захватывают лидерство при количестве партий не 26, а гораздо раньше, а именно – при количестве партий, равном 10. Таким образом, данное обстоятельство можно считать аргументом в пользу гипотезы о масштабируемости настроенного ГА.

С практической точки зрения подтверждение гипотезы о масштабируемости настроенного ГА может означать, что для рассматриваемого класса задач оптимизации плана производства поточных линий ГА может быть настроен по критерию результативности на небольшом

количестве партий, т.е. настроен за достаточно небольшое количество времени, а затем применен в таком настроенном состоянии для решения задач с гораздо большим количеством партий.

Таблица 2

Результаты численных экспериментов

Число партий	Место				
	1	2	3	4	5
26	М2И2К2 50,8732	М3И2К2 50,9079	М2И2К3 51,0426	М3И2К3 51,0539	М1И2К2 51,0578
27	М3И2К2 52,4048	М2И2К2 52,4876	М1И2К2 52,5292	М3И2К3 52,5335	М1И2К3 52,5859
28	М3И2К2 54,2341	М2И2К2 54,2536	М2И2К3 54,3628	М1И2К2 54,3767	М1И2К3 54,3866
29	М3И2К2 56,258	М2И2К2 56,2762	М1И2К2 56,2913	М3И2К3 56,3017	М2И2К3 56,3958
30	М2И2К2 57,8095	М3И2К2 57,8138	М1И2К2 57,9248	М3И2К3 57,9551	М2И2К3 58,0539

Заключение

На основании проведенного исследования можно сделать следующие выводы:

1. ГА могут быть успешно применены для решения задач оптимизации плана производства поточных линий не только в пищевой, но и в других отраслях промышленности.

2. Настройка ГА под конкретный класс задач может быть осуществлена путем подбора соответствующего сочетания генетических операторов, обеспечивающих достижение требуемых значений критериев эффективности и результативности ГА.

3. На основании значительного количества численных экспериментов удалось получить аргументы в пользу подтверждения гипотезы о масштабируемости настроенного ГА, т.е. сохранения оптимальности сочетания генетических операторов при значительном увеличении количества партий, производство или обработка которых планируется.

Список литературы

1. Зак Ю.А. Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения отдельных заданий и времена работы машин. Ч. 1. Точные методы решения // Проблемы управления. – 2010. – № 3. – С. 17–25.

2. Johnson S.M. Optimal-two-and tree stage production schedules with setup times included // *Research Logistics Quarterly*. – 1954. – No. 1. – P. 61–68.

3. Производственный менеджмент: учеб. пособие / под ред. В.А. Козловского. – М.: ИНФРА-М, 2003. – 574 с.

4. Глухов В.В. Организация прокатного производства. – СПб.: Лань, 2001. – 368 с.

5. Рудольф В.В., Орещенко А.В., Яшнова П.М. Производство безалкогольных напитков: справ. – СПб.: Профессия, 2000. – 360 с.

6. Управление качеством продукции на современных промышленных предприятиях: моногр. / С.А. Федосеев, М.Б. Гитман, В.Ю. Столбов, А.В. Вожаков. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2011. – 229 с.

7. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: пер. с англ. – М.: Мир, 1982. – 416 с.

8. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Харьков: ОСНОВА, 1997. – 112 с.

9. Панченко Т.В. Генетические алгоритмы: учеб. пособие / под ред. Ю.Ю. Тарасевича. – Астрахань: Изд-во Астрахан. ун-та, 2007. – 87 с.

References

1. Zak Iu.A. Reshenie obobshhennoi zadachi Dzhonsona s ogranicheniiami na sroki vypolneniia ot del'nykh zadaniy i vremena raboty mashin. Chast' 1. Tochnye metody reshenii [Solving of the Johnson's problem, wich is generalized to the case when the restrictions on the time for execution of individual tasks, and the time of equipment work are set machines. Part 1. Exact methods for solving the problem]. *Problemy upravleniia*, 2010, no. 3, pp. 17-25.

2. Johnson S.M. Optimal-two-and tree stage production schedules with setup times included. *Research Logistics Quarterly*, 1954, no. 1, pp. 61-68.

3. Proizvodstvennyi menedzhment [Industrial management]. Ed. V.A. Kozlovskii. Moscow: INFRA-M, 2003. 574 p.

4. Glukhov V.V. Organizatsiia prokatnogo proizvodstva [Industrial engineering of the rolling production]. Saint-Petersburg: Lan', 2001. 368 p.

5. Rudol'f V.V., Oreshchenko A.V., Iashnova P.M. Proizvodstvo bezalkogol'nykh napitkov [Production of soft drinks]. Saint-Petersburg: Professiia, 2000. 360 p.

6. Fedoseev S.A., Gitman M.B., Stolbov V.Ju., Vozhakov A.V. Upravlenie kachestvom produktsii na sovremennykh promyshlennykh predpriatiiakh [Product quality management on modern industrial companies]. Perm: Permskii natsional'nyi issledovatel'skii universitet, 2011. 229 p.

7. Geri M., Dzhonson D. Vychislitel'nye mashiny i trudnoreshaemye zadachi [Computing machinery and difficult solving tasks]. Moscow: Mir, 1982. 416 p.

8. Voronovskii G.K., Makhotilo K.V., Petrashev S.N., Sergeev S.A. Geneticheskie algoritmy, iskusstvennye neironnye seti i problemy virtual'noi real'nosti [Genetic algorithm, artificial neural networks and problems of virtual reality]. Kharkov: OSNOVA, 1997. 112 p.

9. Panchenko T.V. Geneticheskie algoritmy [Genetic algorithm]. Ed. Iu.Iu. Tarasevich. Astrakhan': Astrahanskii universitet, 2007. 87 p.

Получено 02.04.2015

Об авторах

Розенков Константин Григорьевич (Пермь, Россия) – студент Пермского национального исследовательского политехнического университета (614990, г. Пермь, Комсомольский пр., 29, e-mail: alter_kos@mail.ru).

Федосеев Сергей Анатольевич (Пермь, Россия) – доктор технических наук, профессор кафедры «Математическое моделирование систем и процессов» Пермского национального исследовательского политехнического университета (614990, г. Пермь, Комсомольский пр., 29, e-mail: fsa@gelicon.biz).

Вдовиченко Андрей Александрович (Пермь, Россия) – студент Пермского национального исследовательского политехнического университета (614990, г. Пермь, Комсомольский пр., 29, e-mail: ramul2b@mail.ru).

About the authors

Konstantin G. Rozenkov (Perm, Russian Federation) – Student, Perm National Research Polytechnic University (29, Komsomolsky av., Perm, 614990, Russian Federation, e-mail: alter_kos@mail.ru).

Sergei A. Fedoseev (Perm, Russian Federation) – Doctor of Technical Sciences, Professor, Department of Mathematical Modeling of Systems and Processes, Perm National Research Polytechnic University (29, Komsomolsky av., Perm, 614990, Russian Federation, e-mail: fsa@gelicon.biz).

Andrei A. Vdovichenko (Perm, Russian Federation) – Student, Perm National Research Polytechnic University (29, Komsomolsky av., Perm, 614990, Russian Federation, e-mail: ramul2b@mail.ru).