Оптимизация размещения буферных устройств в автоматических линиях

А.Б.Долгий, А.В.Еремеев, А.А.Колоколов, В.С.Сигаев

University of Technology of Troyes,

Омский филиал Института математики им. С.Л.Соболева СО РАН,

Омский государственный университет

Аннотация

Рассматривается задача оптимизации размещения буферных устройств между единицами оборудования в автоматических технологических линиях параллельно-последовательной структуры. Предложен генетический алгоритм поиска решений данной задачи, использующий оценку параметров линии с помощью марковских моделей.

Ключевые слова: буферные устройства, оптимизация, генетический алгоритм.

Введение

В работе рассматривается задача оптимизации размещения буферных устройств для временного хранения обрабатываемых деталей между единицами оборудования (ЕО) в автоматических технологических линиях. Предполагается, что линия имеет параллельно-последовательную призводственно-технологическую структуру, и в процессе работы происходят случайные по моменту возникновения и длительности остановки оборудования, вызванные отказами. Последствия отказов имеют тенденцию распространяться на смежные операции из-за невозможности передать деталь на следующую операцию или отсутствия деталей на входе ЕО. Наличие промежуточных буферных емкостей между ЕО позволяет снизить влияние отказов на соседние операции и повысить среднюю производительность линии, однако установка буферов связана с дополнительными капитальными затратами и может повлечь повышение стоимости складирования деталей.

В данной работе предлагается генетический алгоритм (ГА) оптимизации объемов буферов, управляющий эвристикой локального поиска посредством выбора начальных решений. Проводится сравнение качества приближенных решений, полученных ГА и эвристикой локального поиска со случайными начальными данными. Оценка параметров технологической линии при различных расположениях буферов осуществляется с помощью моделей марковского типа [1].

1 Постановка задачи

Предположим, что в буфере на входе линии всегда имеется достаточное число деталей, и обработанные детали помещаются в выходной буфер неограниченного объема. Структура линии представляет собой ориентированный параллельно-последовательный граф G, где вершины $b_0, ..., b_n$ отвечают буферным устройствам, а дуги $a_1, ..., a_m$ - единицам оборудования. Каждая дуга направлена от вершины входного буфера соответствующей ЕО к вершине выходного и возможны кратные дуги, т.е. параллельно работающие ЕО с общими буферами. Единственная вершина b_0 без входящих дуг соответствует входному буферу линии и единственная вершина b_n без выходящих дуг - выходному. Каждой вершине $b_j, j = 0, ..., n$ приписано число вмещаемых деталей h_j в буфер j, причем $h_0 = h_n = \infty$.

Любая ЕО может находиться в состояниях "работа", "отказ" (ЕО восстанавливается), "блокировка" (отсутствие возможности передать обработанную деталь на следующую операцию) и "простой" (отсутствие деталей на входе ЕО). Находясь в состоянии "работа", ЕО с номером i имеет постоянное время обработки детали, равное u_i, $i = 1, ..., m$. В состояниях "блокировка" и "простой" ЕО не отказывает. Предполагается, что потоки отказов и восстановлений - простейшие, интенсивность отказов i-й ЕО равна $\lambda_i = 1/T_O^i$, интенсивность восстановления равна $\mu_i = 1/T_B^i$, где T_O^i, T_B^i - время наработки на отказ и среднее время восстановления ЕО, $i = 1, ..., m$. Отказы и восстановления различных ЕО независимы. Таким образом, каждая дуга a_i графа G характеризуется тройкой параметров (u_i, λ_i, μ_i).

В литературе предложен ряд алгоритмов для различных постановок задачи оптимизации вектора размещения буферов $H = (h_1, ..., h_{n-1})$, включая алгоритмы динамического программирования, имитации отжига, ГА и др. (см., например, [4]). В отличие от данной работы в известных нам публикациях рассматриваются ЕО со случайным временем обработки деталей с распределением специального вида.

Пусть $V(H)$ – производительность линии, т.е. среднее число обрабатываемых деталей в единицу времени в стационарном режиме работы; $B(H)$ – суммарный объем буферов; $q_j(H)$ – среднее число деталей в промежуточных буферах в стационарном режиме $(j = 1, ..., n-1)$. Кроме того, предположим, что максимальный объем буфера j ограничен сверху величиной $d_j, j = 1, ..., n-1$. Рассматриваемые далее алгоритмы могут применяться для задач вида

$$\max\{\phi_G(V(H), B(H), Q(H)) : 0 \leq h_j \leq d_j, j = 1, ..., n-1\},$$

где $Q(H) = (q_1(H), q_2(H), ..., q_{n-1}(H))$. Ограничения на минимальную или максимальную допустимую производительность, максимальный общий объем буферов и т.д. могут учитываться посредством "штрафов" через функцию ϕ_G. Далее мы используем критерий

$$\phi_G(V(H), B(H), Q(H)) = T_{am}S(V(H)) - J(B(H)) - T_{am}\sum_{j=1}^{n-1} c_j q_j(H),$$

где T_{am}-период амортизации линии; $S(V(H))$-доход от продажи готовой продукции за единицу времени; c_j-затраты на хранение одной детали в буфере j в единицу времени; $J(B(H))$-капитальные затраты на установку буферов. В функциях $S(V(H))$ и $J(B(H))$ может также учитываться "штраф" за нарушение ограничений по общему объему буферов, по требуемой производительности линии и т.д. Для вычисления $\phi_G(H)$ будем пользоваться эффективным приближенным методом анализа линии [1, 2], основанным на редукции графа G с применением марковских моделей для пар последовательных и параллельных ЕО.

2 Описание генетического алгоритма

Процесс работы ГА представляет собой последовательную смену популяций, состоящих из фиксированного числа *особей* – пробных точек пространства решений, при этом каждая особь представлена строкой символов некоторого алфавита, называемой *генотипом* (элементы этих строк принято называть *генами*). Особи, отвечающие решениям с бóльшим качеством (более пригодные), получают преимущество в порождении потомков в следующей популяции.

Положим, что строка генотипа H состоит из $n-1$ гена и значение гена j – объем соответствующего буферного устройства $h_j, j = 1, ..., n-1$. Обозначим генотипы текущей популяции $H^{(1)}$, $H^{(2)}, ..., H^{(N)}$, тогда $h_j^{(i)}$ - значение гена j в генотипе особи i.

В начальную популяцию помещаются особи, полученные процедурой локального поиска $LS(H)$, где генотип H определяет начальное решение для поиска и имеет равномерное распределение на множестве допустимых решений. Эвристика $LS(H)$ имеет следующий вид.

Алгоритм $LS(H)$

1. Положить $f := \phi_G(H)$.
2. Для всех j от 1 до $n-1$ выполнять:
   2.1 Положить $h := h_j, \delta := 1$.
   2.2 Обозначить $H(\delta) = (h_1, ..., h_{j-1}, h+\delta, h_{j+1}, ..., h_{n-1})$.
   2.3 Если $h - \delta \geq 0$ и $\phi_G(H(-\delta)) > \phi_G(H)$, то положить $\delta := -1$.
   2.4 Пока $0 \leq h + \delta \leq d_j$ и $\phi_G(H(\delta)) > \phi_G(H)$, выполнять:
      2.4.1 Положить $H := H(\delta); h := h + \delta$.
3. Если $\phi_G(H) > f$, то перейти на шаг 1.
4. Результат содержится в генотипе H.

ГА основан на так называемой стационарной схеме воспроизводства [3], при которой на каждой итерации генерируется пара новых особей, заменяющих пару "бесперспективных" особей в популяции. После заданного числа итераций t_{max} алгоритм завершает работу и лучшее найденное решение возвращается в качестве результата.

Создание пары новых генотипов "потомков" начинается с выбора пары "родительских" особей из текущей популяции посредством вероятностного оператора селекции. К генотипам выбранных особей применяется оператор *кроссинговера*, заменяющий часть генов одного генотипа на гены другого. Обе полученные после кроссинговера строки подвергаются действию оператора *мутации*, при котором некоторое подмножество элементов строки измененяется случайным образом.

Для выбора номера очередного "родительского" генотипа используется так называемый оператор s-турнирной селекции: $Select_s = \text{argmax}\{\phi_G(H^{(i)}) : i \in U_s\}$, где случайное подмножество $U_s \subseteq \{1, 2, ..., N\}$ составлено из значений s независимых случайных величин с равномерным распределением на $\{1, 2, ..., N\}$.

В результате применения оператора кроссинговера $(H', H'') = Cross(H^{(i)}, H^{(k)})$ с вероятностью P_{cross} получаем

$$H' = (H_1^{(i)}, ..., H_\chi^{(i)}, H_{\chi+1}^{(k)}, ..., H_{n-1}^{(k)}), H'' = (H_1^{(k)}, ..., H_\chi^{(k)}, H_{\chi+1}^{(i)}, ..., H_{n-1}^{(i)}),$$

причем позиция χ выбрана с равномерным распределением между 1 и $n-2$; с вероятностью $1 - P_{cross}$ полагаем $H' = H^{(i)}, H'' = H^{(k)}$. Действие вероятностного оператора мутации определяется функцией $Mut(H) = (h_1 + \xi_1, h_2 + \xi_2, ..., h_{n-1} + \xi_{n-1})$, где ξ_j - целочисленные случайные величины, равномерно распределенные от $\max\{-h_j, -\Delta\}$ до $\min\{d_j - h_j, \Delta\}, j = 1, ..., n-1$.

Процедура $LS(H)$ применяется на каждой итерации ГА к обоим решениям, полученным после мутации. Найденный локальный оптимум H помещается в популяцию на место особи с наименьшим значением целевой функции, если в популяции отсутствует решение H. Иначе в популяцию помещается генотип $Mut(H)$. Параметры ГА были выбраны нами экспериментально: $\Delta = 5$, $P_{cross} = 0.5, s = 5, N = 50$.

В вычислительных экспериментах ГА сравнивался с алгоритмом локального поиска $LS(H)$, в котором начальный вектор H имел равномерное распределение на множестве допустимых решений. На случайные запуски локального поиска отводилось время, равное времени работы ГА на соответствующей задаче и в качестве результата рассматривался лучший из найденных локальных оптимумов.

В эксперименте для каждой из 16 тестовых задач [2] выполнялось 300 независимых запусков обоих алгоритмов. При этом время одного запуска алгоритма на ПК Pentium, Celeron 460 МГц для наиболее трудоемкой задачи на линии из 14 ЕО составляло примерно 175 сек., а для задач на линиях из 5 ЕО – около 4 сек. Для 13 задач, в которых известно оптимальное решение, ГА обнаруживал его при каждом запуске. При решении остальных задач среднее значение целевой функции на решениях, найденных ГА, статистически значимо превышало среднее для локального поиска (с уровнем доверия 0,95).

Список литературы

[1] Долгий А.Б., Свирин Ю.П. Модели оценки вероятностной производительности автоматизированных технологических комплексов // Вести АН Беларуси. 1995. N 1. С. 59-67.

[2] Dolgui A., Eremeev A., Kolokolov A., Sigaev V. Optimization of buffer allocation in production line. To appear in Proc. of Joint Meeting of IEMS and ICC&IE, Cocoa Beach, Florida, USA, March 5-7, 2001.

[3] Reeves C.R. Genetic Algorithms for the Operations Researcher// INFORMS Journal on Computing. 1997. V. 9, N 3, P.231-250.

[4] Spinellis D., Papadopoulos C. Stochastic algorithms for buffer allocation in reliable production lines// Mathematical Problems in Engineering. 2000, V. 5. P. 441-458.

e-mail: $dolgui@univ-troyes.fr$, $eremeev@iitam.omsk.net.ru$, $kolo@iitam.omsk.net.ru$

Optimization of buffer devices allocation in automatic lines

A.B.Dolgui, A.V.Eremeev, A.A.Kolokolov, V.S.Sigaev
(University of Technology of Troyes, Troyes, France,
Omsk Branch of Sobolev Institute of Mathematics, Omsk, Russia,
Omsk State University, Omsk, Russia)

In this paper we consider a problem of buffer devices allocation in flow-line manufacturing system of machines. It is supposed that the line has a series-parallel structure and the equipment is subject to breakdowns. The failure and repair times of machines are supposed to be exponentially distributed and the processing time at each mashine is deterministic. The buffer allocation problem consists in determining the buffer capacities with respect to given optimality criterion, which depends on the average production rate, the buffer allocation cost and the inventory cost of holding parts in buffers. For this problem we propose a genetic algorithm which controls a local search heuristic by choosing its starting solutions. The tentative solutions are estimated by means of approximate decomposition and aggregation method of flow-line performance evaluation based on the Markov models.

Keywords: buffer devices, optimization, genetic algorithm.