

Применение OpenMP для оптимального выбора аппаратурного оформления многоассортиментных производств

А. Б. БОРИСЕНКО*, С. В. КАРПУШКИН

Тамбовский государственный технический университет, Россия

*Контактный e-mail: b0ris2004@yandex.ru

Представлен параллельный алгоритм оптимального выбора аппаратурного оформления многоассортиментных производств, основанный на схеме ветвей и границ. Алгоритм реализован с использованием технологии OpenMP 3.0. Дан анализ ускорения работы программы и эффективности использования ядер процессоров.

Ключевые слова: дискретная оптимизация, параллельный метод ветвей и границ, OpenMP, многоассортиментные производства, определение аппаратурного оформления.

Введение

Выбор аппаратурного оформления (АО) химико-технологических систем (ХТС) — одна из основных задач, возникающих как при проектировании новых, так и при модернизации существующих многоассортиментных химических производств (МХП) [1–3], например производств химических реактивов, фармацевтических препаратов, синтетических красителей и полупродуктов. Решением этой задачи является число основных аппаратов на каждой стадии системы, а также рабочий объем или площадь рабочей поверхности каждого аппарата. Рабочие объемы и площади рабочих поверхностей обычно выбираются из множества стандартных дискретных значений. Требуется найти оптимальную комбинацию аппаратов из всего множества возможных вариантов, при этом критерием оптимальности, как правило, служат суммарные капитальные затраты на оборудование [4].

Рассматриваемая задача относится к классу задач частично-целочисленного нелинейного программирования (Mixed Integer Nonlinear Programming — MINLP) [5, 6]. В литературе предложены различные подходы к ее решению, например методы Монте-Карло, генетические алгоритмы, эвристические методы, сведение исходной задачи к задаче нелинейного либо линейного программирования, которые позволяют получить лишь субоптимальные решения [5–7].

Метод ветвей и границ часто используется для решения оптимизационных задач в таких областях, как, например, комбинаторная оптимизация, искусственный интеллект и т. д. Он применяется также для решения задач целочисленного нелинейного программирования [8]. В основе метода лежит идея последовательного разбиения множества допустимых решений на подмножества [9]. Затем определяется нижняя оценка

для целевой функции на каждом подмножестве. Если нижняя оценка не меньше наилучшего из найденных решений, то подмножество отбрасывается и т. д., пока не будет получено оптимальное решение.

Подобный подход для решения задачи выбора аппаратного оформления ХТС предлагался в работе [10], в которой авторы использовали упрощенную математическую модель функционирования ХТС, эвристический подход для вычисления оценочной функции и рассматривали в качестве тестовых примеров задачи небольшой размерности. Вычислительные возможности на момент написания работы (1975 г.) были весьма ограничены.

В работе [11] представлен основанный на схеме ветвей и границ последовательный алгоритм оптимального выбора аппаратного оформления ХТС. Авторами было выполнено исследование быстродействия отдельных фрагментов алгоритма, на основании которого осуществлена его оптимизация. Эксперименты проводились на компьютере следующей конфигурации: процессор Intel Pentium 4 с тактовой частотой 3000 МГц, объем оперативной памяти 2048 МБ, операционная система Scientific Linux 5.3. Для компиляции программы использовался GCC 4.1.2. Для тестового примера (ХТС состоит из 16 стадий, последовательно выпускает 3 продукта, для каждой стадии имеется 5 типоразмеров стандартного оборудования) время выбора оптимального АО составило порядка 46 ч.

В настоящее время одним из наиболее перспективных способов повышения производительности алгоритмов является их распараллеливание, а также использование различных высокопроизводительных вычислительных платформ, таких как многопроцессорные серверы, вычислительные кластеры и гетерогенные вычислительные системы. В работах [12, 13] предложена реализация параллельного алгоритма решения задачи выбора аппаратного оформления ХТС МХП на вычислительных системах с распределенной памятью и с использованием библиотеки Message Passing Interface (MPI) [14]. Алгоритм основан на методе ветвей и границ и использует парадигму мастер-работник (*Master-Worker Paradigm, Master-Slave Paradigm*) и метод назначаемых поддеревьев [15].

В данной статье акцент сделан на реализации параллельного алгоритма решения задачи для вычислительных систем с общей памятью (*shared memory parallel programming*) и с использованием открытого стандарта распараллеливания программ — Open Multi-Processing (OpenMP) [16]. Исследуются возможности его применения для решения прикладной задачи — оптимального выбора аппаратного оформления многоассортиментных химических производств.

1. Постановка задачи

Химико-технологическая система МХП представляет собой совокупность аппаратов (реакторов, емкостей, фильтров, сушилок и другого технологического оборудования), которые предназначены для реализации технологических процессов производства продуктов фиксированного ассортимента. При допущении, что число аппаратов на каждой аппаратурной стадии ХТС фиксировано, задача может быть сформулирована следующим образом.

Химико-технологическая система представляет собой последовательность I аппаратурных стадий. На каждой i -й стадии системы требуется основной аппарат, выбранный из некоторого конечного множества X_i , количество элементов которого равно J_i . Множество X_i представляет собой все возможные варианты определяющих размеров аппа-

ратов стадии i : $X_i = \{x_{i,j}\}, i = \overline{1, I}, j = \overline{1, J_i}$, где $x_{i,j}$ — определяющий размер (рабочий объем, площадь рабочей поверхности) j -го аппарата, которым может быть оснащена i -я стадия ХТС.

Каждый вариант аппаратного оформления ХТС $\Omega_e, e = \overline{1, E}$, где $E = \prod_{i=1}^I J_i$, представляет собой упорядоченную совокупность определяющих размеров аппаратов, выбранных из соответствующих множеств. Например, существует ХТС, состоящая из трех стадий ($I = 3$). Первая стадия системы может быть оборудована аппаратом, определяющий размер которого выбирается из двух вариантов, т. е. $J_1 = 2, X_1 = \{x_{1,1}, x_{1,2}\}$, вторая стадия — из трех вариантов, т. е. $J_2 = 3, X_2 = \{x_{2,1}, x_{2,2}, x_{2,3}\}$, и третья — из двух вариантов, т. е. $J_3 = 2, X_3 = \{x_{3,1}, x_{3,2}\}$. Общее количество всех возможных вариантов аппаратного оформления ХТС $E = J_1 \cdot J_2 \cdot J_3 = 2 \cdot 3 \cdot 2 = 12$.

Порядок переработки партий продуктов основными аппаратами на каждой стадии ХТС фиксирован (заранее определен при разработке структуры ее материальных потоков), т. е., например, варианты $\{x_{1,1}, x_{2,1}, x_{3,2}\}, \{x_{1,2}, x_{2,1}, x_{3,1}\}$ могут быть допустимыми, а варианты $\{x_{3,1}, x_{2,1}, x_{1,2}\}, \{x_{2,2}, x_{3,1}, x_{1,1}\}$ — явно недопустимы.

Для каждого работоспособного варианта АО ХТС должны выполняться условия совместного функционирования оборудования на различных ее стадиях $S(\Omega_e) = 0$ (*ограничение на работоспособность ХТС*): вариант АО ХТС работоспособен, если аппараты всех ее аппаратных стадий без исключения способны переработать партии всех выпускаемых продуктов с учетом максимальных и минимальных степеней заполнения аппаратов, их максимальной производительности.

Кроме того, работоспособный вариант АО ХТС должен обеспечивать выпуск продукции заданного ассортимента в плановых объемах в течение заданного промежутка времени, т. е. $T(\Omega_e) \leq T_{\max}$ (*ограничение на длительность функционирования ХТС*), где T_{\max} — заданная максимальная продолжительность выпуска продукции.

Таким образом, задачу выбора аппаратного оформления химико-технологической системы можно сформулировать следующим образом: необходимо найти такой вариант $\Omega^* \in \Omega_e, e = \overline{1, E}$, аппаратного оформления ХТС, при котором критерий оптимальности — капитальные затраты $Cost(\Omega_e)$ — достигает минимума и выполняются ограничения на работоспособность и длительность функционирования ХТС:

$$\Omega^* = \operatorname{argmin} Cost(\Omega_e), \quad e = \overline{1, E}, \quad (1)$$

$$\Omega_e = \{x_{1,j_1}, x_{2,j_2}, \dots, x_{I,j_I} | j_i = \overline{1, J_i}, i = \overline{1, I}\}, \quad e = \overline{1, E}, \quad (2)$$

$$x_{i,j} \in X_i, \quad i = \overline{1, I}, \quad j = \overline{1, J_i}, \quad (3)$$

$$S(\Omega_e) = 0, \quad e = \overline{1, E}, \quad (4)$$

$$T(\Omega_e) \leq T_{\max}, \quad e = \overline{1, E}. \quad (5)$$

Подробно математическая модель функционирования ХТС, включающая в том числе выражения для проверки ограничений, вычисления критерия оптимизации и т. д., приведена в [17].

2. Последовательный вариант алгоритма

Все множество вариантов аппаратного оформления химико-технологической системы, состоящей из I стадий, может быть представлено в виде дерева, состоящего из I ярусов (рис. 1).

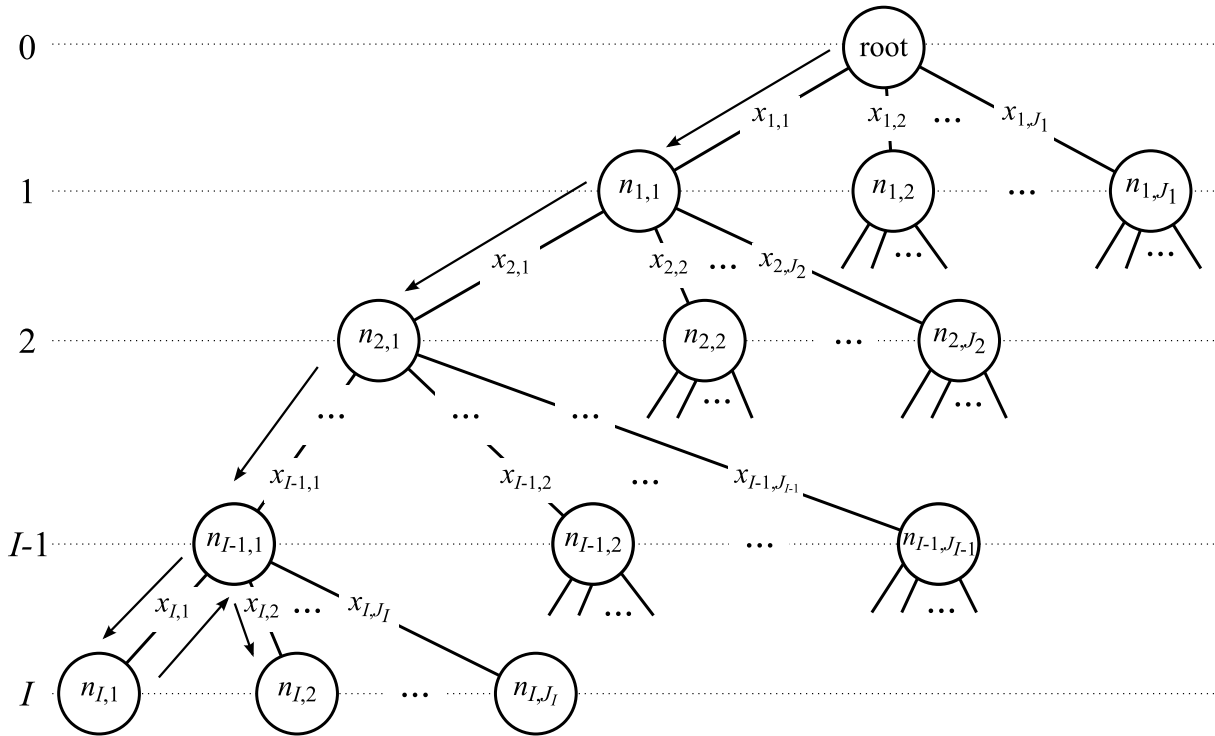


Рис. 1. Обход в глубину дерева вариантов

Каждый ярус дерева соответствует стадии ХТС. Каждое ребро дерева — одному выбранному элементу множества $X_i = \{x_{i,j}\}$, $i = \overline{1, I}$, $j = \overline{1, J_i}$. Например, ребра, исходящие из корневой вершины (нулевой ярус), соответствуют элементам множества X_1 . Каждый узел $n_{i,k}$ яруса $N_i = \{n_{i,1}, n_{i,2}, \dots, n_{i,k}\}$, $i = \overline{1, I}$, $k = \overline{1, K_i}$, $K_i = \prod_{l=1}^i J_l$, соответствует варианту фрагмента ХТС, состоящего из стадий $1 - i$ (из i начальных стадий). Каждый полный путь от корня дерева к одному из его листьев (конечному узлу) представляет собой вариант АО химико-технологической системы.

Перебор возможных вариантов аппаратного оформления ХТС осуществляется с использованием обхода дерева в глубину [9, 18].

На каждом ярусе дерева осуществляется перебор определяющих размеров аппаратов, доступных для оборудования на этой стадии. Выбранный размер добавляется к начальному фрагменту АО химико-технологической системы, полученному на предыдущих ярусах дерева. Процесс продолжается рекурсивно для всех работоспособных начальных фрагментов АО системы. Таким образом, на последнем ярусе дерева получается полный работоспособный вариант аппаратного оформления ХТС, рекурсивный процесс останавливается и полученный вариант сравнивается с текущим оптимальным. Если получен вариант с меньшим значением критерия, то он заменяет текущее оптимальное решение.

Полный обход дерева вариантов (фиксация на каждой стадии ХТС размера аппарата, выбранного из соответствующего этой стадии ряда стандартных размеров, и формирование полного варианта АО ХТС) дальнейшая проверка ограничений (4) и (5) приведут к значительным вычислительным затратам. Для сокращения числа перебираемых вариантов используется алгоритм, основанный на методе ветвей и границ, псевдо-

```

1 FindSolution () { EnumerateVariants (0); }
2 /* рекурсивный обход дерева */
3 EnumerateVariants(i) {
4   if (i < I) {
5     for (j = 1; j <= J[i]; j++) {
6       /* создание начального фрагмента ХТС (стадии 1 \cdot i) */
7       W[i] = X[i, j];
8       /* проверка работоспособности и
9        стоимости начального фрагмента ХТС */
10      if (PartS(W) == 0 && PartCost (W, i) < minCost ) {
11        /* рекурсивный поиск */
12        EnumerateVariants (i + 1); } } }
13 else { /* достигнут конечный ярус дерева */
14   /* проверка критерия оптимизации */
15   if (Cost (W) < minCost) {
16     /* проверка ограничения (5) на длительность функционирования ХТС */
17     if (T(W) <= Tmax ) {
18       /* сохранение нового оптимального решения */
19       Wopt = W;
20       minCost = Cost(Wopt); } } }
21 }

```

Листинг 1. Последовательный вариант алгоритма

код которого представлен в листинге 1 (имена большинства переменных соответствуют именам, используемым в постановке задачи).

Обход дерева вариантов начинается с вызова процедуры `EnumerateVariants` на нулевом ярусе (строка 1). Эта процедура вызывается рекурсивно вплоть до яруса $I-1$ (строка 4). Затем текущий оптимальный вариант ХТС сохраняется в векторе `Wopt` с числом элементов I , каждый из которых хранит выбранный определяющий размер аппарата для каждой стадии. Рабочий вариант ХТС хранится в аналогичном векторе `W`.

Для сокращения вариантов перебора в оригинальном варианте алгоритма применялась следующая схема. На каждом ярусе i дерева проверяется ограничение (4) (процедура `PartS()`, строка 10) — условие совместного функционирования оборудования ХТС, но не для всей ХТС целиком, а только для ее начального фрагмента, состоящего из уже пройденных вышележащих ярусов и текущего (фрагмент ХТС, состоящий из стадий $1-i$). Если часть схемы уже неработоспособна, то нет смысла перебирать аппараты на нижележащих ярусах. Кроме того, стоимость начального фрагмента АО ХТС (процедура `PartCost()`, строка 10) сравнивается со стоимостью текущего оптимального варианта АО ХТС `minCost`. Таким образом, переход на следующий уровень рекурсии (вызов процедуры `EnumerateVariants ()` для следующего уровня $i+1$, строка 12) осуществляется лишь в том случае, если начальный фрагмент ХТС, состоящий из стадий $1-i$, во-первых, работоспособен и, во-вторых, его стоимость не превышает стоимости наилучшего на текущий момент варианта АО ХТС. В противном случае происходит возврат на предыдущий ярус дерева.

Если достигнут конечный ярус дерева ($i = I$, строка 13), т.е. рассматриваемый вариант АО ХТС работоспособен, то его стоимость (результат выполнения процедуры

`Cost()`) сравнивается со стоимостью текущего наилучшего варианта АО ХТС `minCost` (строка 15).

Если найдено лучшее решение, проверяется ограничение (5) на длительность функционирования ХТС (процедура `T()`, строка 17). Если ограничение (5) выполняется, то найден новый оптимальный вариант аппаратного оформления ХТС и он заменяет текущее оптимальное решение `Wopt` (строка 19), а его стоимость сохраняется в переменной `minCost` (строка 20).

Начальным значением переменной `minCost` служит сумма максимальных стоимостей аппаратов каждой стадии.

3. Параллельный вариант алгоритма

В методе ветвей и границ пространство поиска представляет собой дерево, что обеспечивает хорошие предпосылки для распараллеливания алгоритма, так как различные ветви дерева могут обрабатываться одновременно.

В предлагаемом варианте алгоритма все узлы дерева $N_i = \{n_{i,1}, n_{i,2}, \dots, n_{i,k}\}$, $i = \overline{1, I}$, $k = \overline{1, K_i}$, $K_i = \prod_{l=1}^i J_l$, для каждого яруса i рассматриваются как независимые задачи (*tasks*), которые могут выполняться параллельно. Общее количество таких задач, равное $N_{tasks} = \sum_{i=1}^G K_i$, $1 \leq G \leq I$, может быть очень большим числом. Параметр G (*granularity* — гранулярность) алгоритма позволяет ограничивать степень параллелизма до определенного уровня дерева: поддеревья ниже этого уровня не разделяются на параллельные задачи и обрабатываются последовательно.

```

1 FindSolution() { EnumerateVariants(0); }
2
3 EnumerateVariants(level) {
4   if (level < I) {
5     for (j=1; j <= J[level]; j++) {
6       /* создание начального фрагмента ХТС (стадии 1 - i) */
7       W[level] = X[level, j];
8       /* проверка работоспособности и
9        стоимости начального фрагмента ХТС */
10      if (S(W) == 0 && PartCost(W, level) < minCost) {
11        if (level < G) { /* если текущая глубина поиска меньше G */
12          /* создание параллельной задачи */
13          CREATE TASK: EnumerateVariants(level + 1); }
14        else {
15          /* последовательный рекурсивный обход дерева */
16          EnumerateVariants(level + 1); } } } }
17      else { /* достигнут конечный ярус дерева */
18        ... }
19 }

```

Листинг 2. Параллельный вариант алгоритма

Задачи (*tasks*) являются альтернативой традиционным потокам (*threads*), позволяющей реализовать лучший баланс нагрузки, эффективное использование имеющихся ресурсов и высокий уровень абстракции, дающий возможность программисту рассматривать логический параллелизм своих приложений, не заботясь о непосредственном управлении параллельным выполнением [19].

Псевдокод предлагаемого параллельного алгоритма представлен в листинге 2.

Основное отличие параллельного варианта алгоритма от представленного выше последовательного заключается в том, что каждый вызов рекурсивной процедуры обхода дерева `EnumerateVariants` представляет собой отдельную независимую задачу (строка 13). Следует отметить, что при параллельной обработке этих задач необходимо обеспечить создание и передачу каждой задаче копии массива W — текущего начального фрагмента ХТС.

4. Результаты экспериментов

Для проведения численных экспериментов были разработаны последовательный и параллельный варианты программ на языке программирования C++. Параллельный вариант программы реализован с использованием открытого стандарта распараллеливания программ — Open Multi-Processing (OpenMP) — версии 3.0 [16]. Начиная с версии 3.0, в OpenMP появилась поддержка директивы `#task`, позволяющей создавать отдельные независимые задачи.

В качестве тестового примера использовался расчет аппаратного оформления для ХТС, состоящей из 16 аппаратных стадий, для каждой из которых доступно 5 стандартных типоразмеров оборудования. Химико-технологическая система должна в течение 7400 ч обеспечить выпуск трех красителей объемом 2250, 100 и 50 т (подробнее см. в [11]).

Эксперименты проводились с использованием вычислительной системы на базе четырех восьмиядерных процессоров Intel Xeon E7550 2 ГГц (всего 32 ядра) и 128 ГБ ОЗУ. Для компиляции программы использовался Intel C++ Compiler 11.1.

Параллельная реализация алгоритма является развитием последовательной программы, в исходный код которой добавлены директивы OpenMP.

Начальный вызов рекурсивной процедуры `EnumerateVariants` (строка 1 листинга 2) осуществляется внутри директивы `#pragma omp single`, вложенной в директиву `#pragma omp parallel`, что позволяет лишь одному из доступных параллельных потоков (*parallel threads*) выполнять рекурсивный обход дерева.

Последующие рекурсивные вызовы процедуры `EnumerateVariants` осуществляются внутри директивы `#pragma omp task` (строка 13 листинга 2). Таким образом, первый поток программы в процессе рекурсивного обхода дерева на каждом уровне i , $0 < i < G$, создает новые независимые задачи, которые автоматически помещаются в пул задач, из которого ее может взять любой свободный поток для выполнения.

На рис. 2 представлены результаты исследования зависимости времени выполнения параллельной программы на различном числе процессоров от параметра гранулярности G . Следует отметить, что при значении параметра G больше 10 количество задач было настолько большим, что для их хранения не хватило памяти (наблюдалось аварийное завершение выполнения программы).

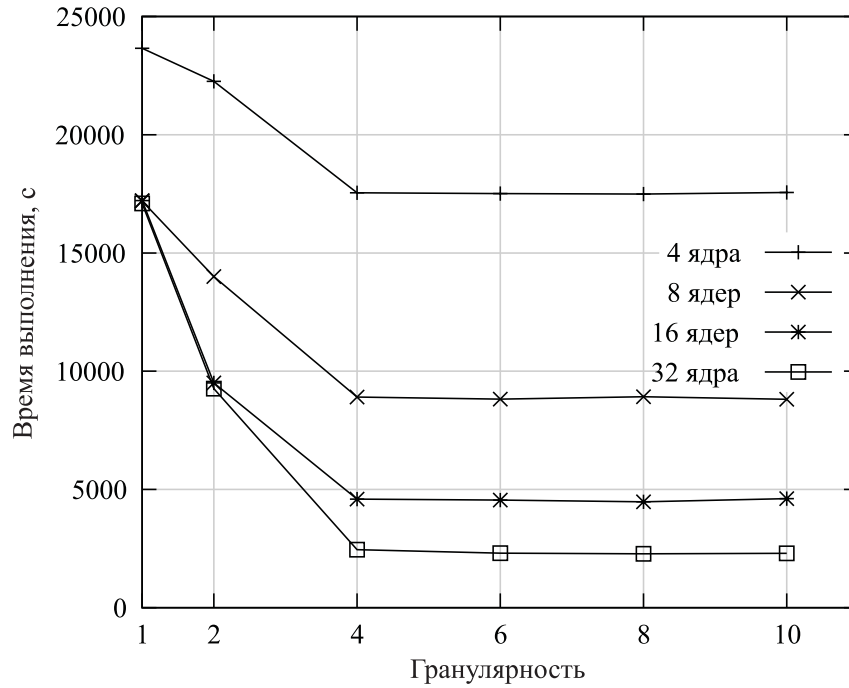
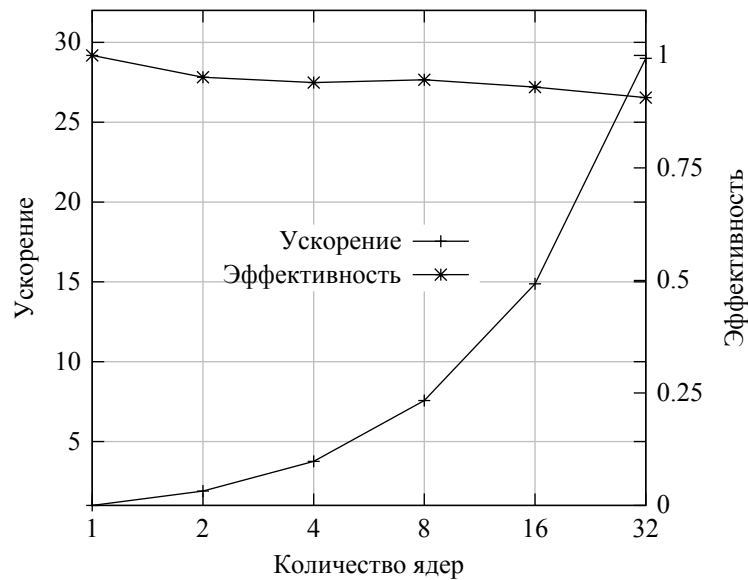
Рис. 2. Зависимость времени выполнения программы от гранулярности G 

Рис. 3. Зависимость ускорения и эффективности использования ядер от количества ядер

Было проведено исследование ускорения и эффективности использования ядер. Ускорение S_p определяется как отношение процессорного времени T_1 , требуемого для выполнения последовательной программы на одном ядре, ко времени T_p выполнения вычислений параллельной программой на p -ядерной машине. Эффективность использования ядер E_p определяется как отношение полученного ускорения S_p к количеству ядер p . Результаты представлены на рис. 3. Значение параметра G при этом равно 10.

Выводы

Предложен основанный на схеме ветвей и границ параллельный алгоритм оптимального выбора АО химико-технологической системы. Алгоритм был реализован в виде параллельной программы для вычислительных систем с общей памятью (*shared memory parallel programming*) на языке программирования C++ с использованием открытого стандарта распараллеливания программ OpenMP 3.0.

Программная реализация алгоритма с использованием OpenMP существенно проще, чем параллельный вариант алгоритма для вычислительных систем с распределенной памятью, использующий библиотеку MPI и реализующий парадигму мастер-работник (*master-worker*). Отличие от последовательного варианта программы заключается лишь в добавлении директив OpenMP.

Значение параметра гранулярности G рекомендуется выбирать таким, чтобы количество задач значительно превышало количество физических ядер вычислительного устройства и было, как минимум, не меньше количества используемых ядер. Количество задач определяется величиной параметра G , для оценки значения которого предлагается эмпирическая формула $G \geq \log_{\max\{J_i\}} N_p i = \overline{1, I}$, где $\max\{J_i\}$ — максимальная мощность множества доступных размеров аппаратов на стадии, N_p — количество используемых ядер. Следует отметить, что при очень большом количестве задач (в нашем случае при $G > 10$) может наблюдаться нехватка памяти. Точно определить количество требуемой памяти представляется затруднительным. Однако по результатам, представленным на рис. 2, видно, что начиная с определенного порогового значения величины G (в нашем примере для $G = 4$) скорость выполнения программы практически не изменяется, несмотря на увеличение количества задач. Поэтому предлагается сначала выполнить вычисления для задачи небольшой размерности, найти пороговое значение G , обеспечивающее приемлемое время решения задачи, а затем проводить вычисления для задач большей размерности.

Исследование ускорения и эффективности работы ядер показало, что разработанная параллельная программа обладает ускорением, близким к линейному, а также достаточно высокой масштабируемостью. Значение эффективности использования ядер 0.9–0.95.

Таким образом, предлагаемый в данной работе параллельный вариант алгоритма выбора оптимального аппаратного оформления химико-технологических систем позволяет значительно сократить время расчета по сравнению с последовательным вариантом и повысить качество проектных решений за счет получения глобального оптимального решения.

Благодарности. Работа выполнена при финансовой поддержке Германской службы академических обменов (DAAD) и Министерства образования и науки РФ в рамках совместной программы “Михаил Ломоносов II”.

Список литературы / References

- [1] Борисенко А.Б., Карпушкин С.В. Иерархия задач аппаратного оформления технологических систем многоассортиментных химических производств // Изв. РАН. Теория и системы управления. 2014. № 3. С. 113–123.
Borisenko, A.B., Karpushkin, S.V. Hierarchy of processing equipment configuration design problems for multiproduct chemical plants // Journal of Computer and Systems Sciences International. May 2014, Vol. 53, No. 3. P. 410–419.

- [2] **Карпушкин С.В., Краснянский М.Н., Борисенко А.Б.** Методика выбора основной аппаратуры технологических систем при проектировании многоассортиментных химических производств. Ч. I. Постановки задач и схема их совместного решения // Информационные технологии в проектировании и производстве. 2012. № 3. С. 52–58.
Karpushkin, S.V., Krasnyansky, M.N., Borisenko, A.B. Method for choosing of engineering systems' main apparatuses when designing of multi-product chemical plants. Pt I. Statements of tasks and scheme of their joint decision // Information Technologies in Design and Production. 2012. No. 3. P. 52–58. (in Russ.)
- [3] **Карпушкин С.В., Краснянский М.Н., Борисенко А.Б.** Модернизация химико-технологических систем действующего многоассортиментного производства при изменении планов выпуска продукции // Химическая промышленность сегодня. 2012. № 1. С. 26–32.
Karpushkin, S.V., Krasnyansky, M.N., Borisenko, A.B. Modernization of operating multi-product plant's chemical-technological systems in case of products turnout plan's variation // Khimicheskaya Promyshlennost Segodnya. 2012. No. 1. P. 26–32. (in Russ.)
- [4] **Карпушкин С.В., Краснянский М.Н., Борисенко А.Б.** Методика оценки эффективности аппаратного оформления химико-технологических систем многоассортиментного производства // Информационные системы и технологии. 2011. № 5(67). С. 96–105.
Karpushkin, S.V., Krasnyansky, M.N., Borisenko, A.B. Technique of the estimation of efficiency of hardware registration of himiko-technological systems of poliassopriment manufactures // Information Systems and Technologies. 2011. No. 5(67). P. 96–105. (in Russ.)
- [5] **Ponsich, A., Azzaro-Pantel, C., Domenech, S., Pibouleau, L.** Mixed-integer nonlinear programming optimization strategies for batch plant design problems // Industrial & Engineering Chemistry Research. 2007. Vol. 46(3). P. 854–863.
- [6] **Hamzaoui, Y.E., Hernandez, J.A., Cruz-Chavez, M.A., Bassam, A.** Search for optimal design of multiproduct batch plants under uncertain demand using gaussian process modeling solved by heuristics methods // Chemical Product and Process Modeling. 2010. Vol. 5, No. 1. P. 1934–2659.
- [7] **Малыгин Е.Н., Карпушкин С.В., Борисенко А.Б.** Методика определения аппаратного оформления многопродуктовых химико-технологических систем // Химическая промышленность сегодня. 2003. № 5. С. 43–50.
Malygin, E.N., Karpushkin, S.V., Borisenko, A.B. Procedure for determination of multiproduct chemical-engineering systems equipment // Himicheskaya Promyshlennost Segodnya. 2003. No. 5. P. 43–50. (in Russ.)
- [8] **Leyffer, S., Linderoth, J., Luedtke, J., Miller, A., Munson, T.** Applications and algorithms for mixed integer nonlinear programming // Journal of Physics: Conference Series. 2009. Vol. 180, No. 1. P. 12–14.
- [9] Алгоритмы. Построение и анализ. Третье издание / **Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн.** М.: Вильямс, 2013.
Introduction to Algorithms. 3rd Edition / **T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein.** The Massachusetts Institute of Technology Press, 2009. 1312 p.
- [10] **Sparrow, R.E., Forder, G.J., Rippin, D.W.T.** The choice of equipment sizes for multiproduct batch plants. Heuristics vs. branch and bound // Industrial & Engineering Chemistry Process Design and Development. 1975. Vol. 14, No. 3. P. 197–203.

- [11] **Борисенко А.Б., Карпушкин С.В.** Применение метода ветвей и границ для оптимального выбора аппаратурного оформления химико-технологических систем // Вычисл. технологии. 2012. Т. 17, № 1. С. 35–43.
Borisenko, A.B., Karpushkin, S.V. Application of branch-and-bound method for optimal selection of process equipment of chemical-engineering systems // Computational Technologies. 2012. Vol. 17, No. 1. P. 35–43. (in Russ.)
- [12] **Борисенко А.Б., Кутузов Д.В., Осовский А.В.** Применение параллельных вычислений для расчета аппаратурного оформления химико-технологических систем // Вестник ТГТУ. 2011. Т. 17, № 2. С. 493–496.
Borisenko, A.B., Kutuzov, D.V., Osovskiy, A.V. Application of parallel computing for the calculation of hardware design of chemical-technological systems // Transactions of the TSTU. 2011. Vol. 17. No. 2. P. 493–496. (in Russ.)
- [13] **Борисенко А.Б., Карпушкин С.В.** Параллельный алгоритм оптимального выбора аппаратурного оформления многоассортиментных производств // Прикладная информатика. 2013. Т. 45, № 3. С. 56–68.
Borisenko, A.B., Karpushkin, S.V. Parallel algorithm for optimal process equipment selection of multiproduct batch plants // Applied Informatics. 2013. Vol. 45, No. 3. P. 56–68. (in Russ.)
- [14] **Message Passing Interface Forum.** Official MPI (Message Passing Interface) standards documents, errata. Available at: <http://www.mpi-forum.org/>
- [15] **Тимошевская Н.Е.** Параллельные методы обхода дерева // Матем. моделирование. 2004. Т. 16. № 1. С. 105–114.
Timoshevskaya, N.E. Parallel methods for tree traverse // Matematicheskoe Modelirovanie. 2004. Vol. 16, No. 1. P. 105–114. (in Russ.)
- [16] **OpenMP Architecture Review Board.** The OpenMP API Specification for Parallel Programming. Available at: <http://www.openmp.org/>
- [17] **Малыгин Е.Н., Карпушкин С.В., Борисенко А.Б.** Математическая модель функционирования многопродуктовых химико-технологических систем // Теорет. основы хим. технологии. 2005. Т. 39, № 4. С. 455–465.
Malygin, E.N., Karpushkin, S.V., Borisenko, A.B. A Mathematical Model of the Functioning of Multiproduct Chemical Engineering Systems // Theoretical Foundations of Chemical Engineering. 2005. Vol. 39, No. 4. P. 429–439.
- [18] **Кнут Д.Э.** Искусство программирования. Т. 1. Основные алгоритмы. М.: Вильямс, 2010. 720 с.
Knuth, D.E. The Art of Computer Programming. Vol. 1: Fundamental Algorithms. Massachusetts: Addison-Wesley, 1997. 650 p.
- [19] Intel Developer Zone. Использование задач вместо потоков. Адрес доступа: <https://software.intel.com/ru-ru/articles/using-tasks-instead-of-threads>
Intel Developer Zone. Using Tasks Instead of Threads. Available at: https://software.intel.com/sites/default/files/m/d/4/1/d/8/1-6-AppThr_Using_Tasks_Instead_of_Threads.pdf

*Поступила в редакцию 6 марта 2015 г.,
с доработки — 29 мая 2015 г.*

Using of OpenMP for optimal choice of process equipment for multiproduct batch plants

BORISENKO, ANDREY B. *, KARPUSHKIN, SERGEY V.

Tambov State Technical University, Tambov, 392000, Russia

*Corresponding author: Borisenko, Andrey B., e-mail: b0ris2004@yandex.ru

In this work, we parallelize the Branch-and-Bound (B&B) algorithm and illustrate it with a practical application — the optimal selection of equipment for multi-product batch plants. We develop and evaluate an implementation of the parallel B&B method on a multicore system with shared memory using C++ programming language and OpenMP version 3.0 (Open Multi-Processing) for the task-based approach.

To limit the degree of parallelism to a certain level of the tree we introduce a *granularity* parameter G : subtrees below the granularity level are not split into tasks but rather processed sequentially. We conducted our runtime experiments on a computer consisting of 4 CPUs with the total of 32 cores. We analyzed the impact of the degree of parallelism controlled by the granularity parameter on the runtime and show the analysis for speedup and efficiency of the processors' usage.

Selecting a suitable granularity value is crucial for optimal performance, but with a large number of tasks we may not have enough memory. It happens when $G > 10$ in our case. Implementation of our algorithm has near-linear speedup and provide high parallel scalability. Efficiency of processors' usage is shown to be 0.9–0.95.

Keywords: discrete optimization, parallel branch-and-bound method, OpenMP, multi-product batch plants, choice of process equipment.

Acknowledgements. This work was generously supported by the German Academic Exchange Service (DAAD) and by the Ministry of Education and Science of the Russian Federation under “Mikhail Lomonosov II”-Programm.

Received 6 March 2015

Received in revised form 29 May 2015