УДК 004.023, 004.272

Ю.А. Иванов, аспирант
Донецкий Национальный технический университет, кафедра КИ
yuriy.o.ivanov@gmail.com

Оптимизация вычислительного процесса на многоядерной системе с использованием алгоритма искусственной пчелиной колонии

В статье рассмотрена задача разработки расписаний для эффективного управления процессом моделирования разночастотных динамических объектов в режиме реального времени на многоядерной платформе. Методами математического программирования данная задача классифицирована как смешанная нелинейная целочисленная. В работе было предложено решение с использованием метавэвристического алгоритма искусственной пчелиной колони, адаптированного для поставленной дискретной задачи оптимизации. Предложен подход для инициализации начального состояния роя, как решение задачи о камнях с использованием жадного алгоритма. Для рассмотренных наборов входных данных определен оптимальный состав популяции в алгоритме. Выполненная оценка работы и надежности алгоритма показала возможность применимости и высокий уровень его эффективности при решении задачи разработки расписания.

Ключевые слова: расписание, вычислительный процесс, метаэвристика, дискретная оптимизация, алгоритм искусственной пчелиной колонии.

Введение

Повышение эффективности исследования различных производственных процессов в значительной мере достигается внедрением средств их модельной поддержки. Для эффективного управления моделями объектов, необходимо решать задачи оптимизации характеристик процесса. Известно достаточно много работ оптимизирующих вычислительный процесс в моделях динамических систем при ограничении систем жесткого реального времени [1,2]. Одними из основных применяемых методов являются методы математического программирования. В зависимости от характера целевого множества переменных рассматриваемой модели это могут быть методы линейного, нелинейного или дискретного программирования.

1. Модель вычислительного процесса. Особенности оптимизации

В работе [3] предложен принцип организации вычислительного процесса для цифровой части системы полунатурного моделирования. Вычисляющая часть системы моделирования может характеризоваться следующими особенностями: на цифровую часть поступают циклически n заявок на исполнение, каждой из которых необходимо процессорное время $\tau_i(i=1,2,...,M)$ с директивным периодом $T_i(i=1,2,...,M)$, где M – количество потоков вычисления фазовых переменных. Обработка потоков вычисления происходит циклически, согласованно по периоду полного цикла $T_ц = НОК(T_i)$. Процесс выполнения всех потоков образует цикл реального времени (РВ-цикл) с периодом L. Для модификации предложенной модели для вычислительного процесса обработки потоков на многоядерной системе в терминах работы[3] предлагается ввести параметр отображающий разбиение потоков по группам. Пусть

$$x_{i,j} = \begin{cases} 0, \text{если } i \text{ поток не обрабатывается на } j-\text{м ядре,} \\ 1, \text{если обрабатывается.} \end{cases}$$

Тогда верно, что нагрузка на каждое ядро не может превышать 100%

$$(\forall j)\sum_{i=1}^{M}\frac{\tau_{i,j}}{T_i}x_{i,j} \le 1 \tag{1}$$

Суммарную нагрузочную характеристику оценивать не имеет смысла, поскольку если расписание невозможно построить для хотя бы одного ядра, то вся система не будет функционировать.

Каждый поток может выполняться только на одном ядре

$$(\forall i)\sum_{j=1}^{n}x_{i,j} = 1 \tag{2}$$

при этом в расписании должны быть учтены все M потоков, из которых состоит задача моделирования (ЗМ)

$$\sum_{j=1}^{n}\sum_{i=1}^{M}x_{i,j} = M \tag{3}$$

Каждый РВ-цикл в такой модели имеет собственное значение периода L_j, который является натуральным числом и не может превышать значение минимального периода обрабатываемых потоков $T_j^{\min}$ на ядре.

Эффективность всей системы, как решателя задачи моделирования, является суммой ее РВ-циклов и определяется функцией:

$$F = \sum_{j=1}^{n} \left[\sum_{i=1}^{M} \frac{\tau_{i,j}}{T_i^{'}} x_{i,j} + \frac{\sum_{k=1}^{M} x_{k,j} p}{L_j} \right] \qquad (4)$$

где $(\forall j,i) T_i^{'} = T_i - \left] \frac{T_i}{L_j} \right[$,

p – время переключения контекста.

Таким образом, модель вычислительно процесса выполняющего обработку потоков задачи моделирования на n ядрах, может быть записана так

$$\arg\min F(L_j, x_{i,j})$$

$$(\forall j) L_j \in N, L_j \le T_j^{\min} \; i = \overline{1,M} \;\; j = \overline{1,n}$$

$$(\forall j,i) x_{i,j} \in Z, 0 \le x_{i,j} \le 1$$

$$F = \sum_{j=1}^{n} \left[\sum_{i=1}^{M} \frac{\tau_{i,j}}{T_i^{'}} x_{i,j} + \frac{\sum_{k=1}^{M} x_{k,j} p}{L_j} \right] \qquad (5)$$

$$(\forall j) \sum_{i=1}^{M} \frac{\tau_{i,j}}{T_i} x_{i,j} \le 1, \; (\forall i) \sum_{j=1}^{n} x_{i,j} = 1$$

$$\sum_{j=1}^{n} \sum_{i=1}^{M} x_{i,j} = M \, , \; (\forall j,i) T_i^{'} = T_i - \left] \frac{T_i}{L_j} \right[$$

Данная модель, определяется функцией F, зависящей от переменных $L_j, x_{i,j}$. Время доступа к памяти системы в модели не рассматривается в связи с тем, что может быть заменено константой, не зависит от переменных целевой функции и не влияет на определение параметров расписания.

Основными характеризующими особенностями данной задачи являются:

1.Многокритериальность. Управляющие переменные модели представлены двумя множествами определяющими разбиение задач по ядрам системы и периоды переключения задач для каждого из ядер.

2.Многоэкстремальность. График функции модели является нелинейным и функция не является выпуклой. При этом она имеет несколько экстремумов, количество, которых зависит от набора входных данных.

В связи с этими особенностями применение детерминированных методов решения целочисленных задач в данном случае не применимо, так как подходы, основанные на применении локального поиска, не дадут необходимого решения. В связи, с чем в работе были рассмотрены недетерминированные методы: симплекс метод, метод имитации отжига, генетический алгоритм, муравьиный алгоритм и алгоритм искусственной пчелиной колонии (artificial bee colony (ABC). Во многих исследованиях[4], в том числе для решения дискретных задач было выполнено сравнение реализаций алгоритмов для разных типов задач[5], [6]. В результате применения к оптимизационным многокритериальным задачам популяционные алгоритмы показали лучший уровень сходимости. Имея множество сходных черт в общей идее своей работы, данные алгоритмы все же имеют существенные отличия. В связи с тем, что алгоритм ABC не уступает остальным из своей группы ни по параметру сходимости, ни по количеству оценок, но при этом обладает существенными преимуществами, среди которых простота реализации и трактовки окончательного результата в работе для нахождения оптимального решения выбран метаэвристический алгоритм искусственной пчелиной колонии.

2. Содержательное описание алгоритма искусственной пчелиной колонии

Алгоритм ABC предназначен для нахождения глобальных экстремумов сложных многомерных функций и в качестве идеи для его создания послужила модель поведения пчел улья при поиске источников питания. В общем виде алгоритм предполагает такие шаги: 1) инициализация, перенос занятых пчел на их участки сбора, 2) выбор участков для принятия решения, 3) посылка пчел-разведчиков искать новые участки, 4) сохранение лучших участков найденных на данный момент, 5) если условие окончания не достигнуто, переход на п.2.

Однако учитывая множество вариантов представления, как самой функции, так и ее параметров данные шаги существенно отличаются для каждой конкретной задачи. Кроме того для повышения эффективности работы алгоритма, на некоторых этапах его выполнения могут возникнуть шаблонные ситуации, когда возможно применение общеизвестного детерминированного метода решения на одном из шагов. При этом заменяется случайная составляющая и тем самым повышается точность решения общего алгоритма и сам алгоритм становится уже гибридным.

Для инициализации начального положения пчел из улья выбираются случайные позиции согласно логике алгоритма. Однако эффективность его работы может быть улучшена в случае принятия оптимального плана решения на этапе инициализации. Поиск такого плана может быть осуществлен на основании каких-то особенностей конкретной решаемой задачи. В данном случае это размещение потоков модели динамической системы на вычислительных ядрах системы. В худшем случае для конкретной задачи случайная инициализация может вообще не дать варианта решения, который бы удовлетворял поставленным ограничениям. Кроме того дальнейший перенос пчел на случайные позиции также может не принести успеха. Поэтому в качестве начального значения в работе было предложено инициализировать задачу значением равномерного базового распределения потоков по ядрам. При такой формулировке ставится цель достижения минимального значения нагрузки на каждом из ядер и таким образом формирование равномерного распределения базовых значения нагрузки $\frac{\tau_i}{T_i}$, без учета РВ-цикла. Таким образом, может быть гарантировано сбалансированное распределение, которое предоставит максимальный запас свободной процессорной мощности при модификации периодов выполнения. Такой план решения будет удовлетворять всем условиям системы ограничений и при дальнейшей его разработке уже рабочими пчелами будет уточнен. Таким образом, на начальных шагах алгоритма будет получен один из перспективных экстремумов.

Формулировка данной задачи равносильна общеизвестной «задаче о камнях»(bin packing), которая принадлежит классу NP- полных задач[7]. Задача заключается в распределений n различных объектов (камней) на m групп так, чтобы объемы объектов во всех группах были минимальны. Сложность задачи определяется дискретным характером. Для ее решения предложено использовать эвристический жадный алгоритм, заключающийся в последовательной загрузке ядер потоками по мере убывания их сложности. Формальное описание алгоритма, позволяющего быстро находить приемлемое для искомых оценок решение, адаптированное к решаемой проблеме заключается в следующем:

1. Имеется множество нагрузок каждого потока (камней) $\frac{\tau_i}{T_i}, i=\overline{1,M}$ которые отсортированы по убыванию: $\frac{\tau_1}{T_1} > \frac{\tau_2}{T_2} > ... > \frac{\tau_M}{T_M}$.
2. Имеется множество ядер (куч) с текущей нагрузкой $F_j, j=\overline{1,n}$. Первоначально множества определяются как пустые, т.е. нагрузка каждого ядра $F_j = 0$.
3. Очередная нагрузка добавляется в то множество из ядер, для которого меньше величина F_j. Т.е. берется самая большая загрузка и помещается на самое мало загруженное в рассматриваемый момент ядро $\left(\frac{\tau_i}{T_i} = \max\right) \to (F_j = \min)$.
4. Если еще остались нераспределенные потоки, то переход на п.3.

Для переноса пчел на участки принята система координат, позволяющая однозначно представить все возможные планы решения. В эту систему включены: представление распределения потоков по ядрам, значение периода L_j на каждом из ядер.

Для удобства, распределение потоков по ядрам рассматривается как десятичное число *x*, ограниченное 0 слева и значением числа Белла для комбинации потоков и ядер: $0 \le x \le S'(M,n)$. Такое число кодирует содержимое матричного представления распределения XTAB, в каждом столбце может быть только одна единица, которая отражает, что i-й поток назначен j-е ядро.

n	**0**	**0**	**…**	**1**	**1**
			…	**0**	**0**
2	**0**	**1**	**…**	**0**	**0**
1	**1**	**0**	**…**	**0**	**0**
n/M	*1*	*2*	…	…	*M*

Если представить номера ядер как цифры числа, начиная с нуля, XTAB будет описывать число в системе счисления с основанием n. Таким образом, для перевода XTAB в x используется перевод из системы счисления с основанием n в десятичную и обратно.

Представление значений периода РВ-цикла будет представляться множеством значений L_j с ограничениями:

$$1 \le L_j \le \begin{cases} T_i, (x_{i,j}=1) \cap (i=\min), i=\overline{1,M} \\ 1, (x_{i,j} \ne 1), i=\overline{1,M} \end{cases}$$

Таким образом, координата участка - $\lfloor x; L_j \rfloor$. Окрестность данной координаты – все точки, охватывающие заданное значение РВ-цикла, а также остальные значения в диапазоне $\pm range$, где $range \in N$, является настраиваемым параметром алгоритма, т.е. $L_j \pm (1:range)$. Под отправкой пчелы в окрестность точки *x* в работе понимается установка по координате

$$pos_{i+1} = \begin{pmatrix} x_i \\ L_{i1} \\ \vdots \\ L_{in} \end{pmatrix} + \begin{pmatrix} 0 \\ rnd_1 * rL_1 \\ \vdots \\ rnd_n * rL_n \end{pmatrix}, \qquad (6)$$

где rnd – случайное целое число, принимающее значение в диапазоне $\left[-range;+range\right]$;

rL – случайное булево число, принимающее значение 0 либо 1.

Отправка пчел на разведку предполагает переход на позицию со случайными координатами

$$pos_{i+1} = \begin{pmatrix} rndX \\ rndT_1 \\ \vdots \\ rndT_n \end{pmatrix}, \qquad (7)$$

где $rndX$ – случайное целое число, принимающее значение в диапазоне $[1; S^{'}(M,n)]$,

$rndT$ – случайное целое число, принимающее значение в диапазоне $[1:T_i]$.

Для каждого участка может быть определен фитнесс, который является качественной оценкой данного участка и используется для дальнейшего сравнения их между собой. Определяется фитнесс участка путем расчета значения целевой функции (4). Для каждого участка известны все данные, поэтому значение *F* может быть рассчитано однозначно.

Принятие решения для переноса пчел на одну из позиций для сбора выполняется на основании данных уже исследованных роем. Каждая пчела выбирает для полета участок по значению вероятности

$$p_i = \frac{fitness_i}{\sum\limits_{k=1}^{Всеп челы} fitness_k} \qquad (8)$$

отражающей значимость позиции пропорционально ко всем остальным позициям найденным роем. Помимо пчел для выбранных участков в улье работают пчелы, выделенные для сбора на лучших участках, фитнесс которых был лучшим на предыдущем шаге. Это позволяет форсировать уточнение экстремума для перспективного решения, за счет рассмотрения большего количества комбинаций периодов РВ-цикла на найденном участке сбора. Также в улье работают пчелы разведчики, которые перемещаются по новым позициям со случайными координатами и таким образом обнаруживают новые планы решения.

3. Вычислительные аспекты реализации применительно к задаче составления расписания на многоядерной системе

В работе алгоритм ABC реализован с использованием объектно-ориентированного подхода, для этого созданы классы пчелы (bee) и улья (hive). Каждая пчела описывается набором своих характеристик, среди которых: координата $\left[x; L_j\right]$, параметры исходной задачи $\left(M, n, \bar{\tau}, \bar{T}, p\right)$, значение фитнеса.

Каждая пчела может: перелететь на случайный участок или перелететь в окрестность участка. Обе функции выполняются с проверкой выхода за пределы диапазона как координаты *x*, так и L_j, что является необходимым требованием модели вычислительного процесса.

Класс улья содержит все объекты класса пчел и алгоритмы управления ими. Улей описывается таким набором данных: количество пчел разведчиков, количество лучших участков, количество пчел работающих на лучших участках, количество выбранных участков, количество пчел работающих на выбранных участках, диапазон окрестности. Параметр, который определяет максимально допустимое изменение периодов Δ при отправке пчелы в окрестность точки x, то есть $\left[x; L_j \pm \Delta\right]$.

Управление процессом работы алгоритма выполняется из главной программы. Для этого необходимо создать экземпляр класса улей и далее требуемое число раз запустить выполнение функции следующего шага (рис.1) развития популяции улья. Количество итераций ограничено условием прекращения вычислений.

Рисунок 1 – Блок-схема описывающая организацию работы улья на одном шаге алгоритма ABC

4. Настройка параметров и исследование результатов работы алгоритма

Исследование работы алгоритма показало высокую эффективность принимаемого в начале решения, которое получается при инициализации задачи. Это было подтверждено экспериментально, для 50 тестовых последовательностей, начальное приближение, находило экстремум отличающейся от результата не более чем на 15%.

При этом качество рассмотрения остальных планов функции не страдает, дальнейшее рассмотрения различных вариантов координаты x позволяет эффективно определить другие экстремумы функции, что было подтверждено исследованием распределения посещений пчел для различных наборов тестовых последовательностей.

Для эффективной работы алгоритма ABC необходимо выполнение оптимальной настройки его параметров, основными из которых являются: количество итераций, количественные показатели пчел. Существует несколько подходов для выбора условия прекращения работы алгоритма. Однако данный параметр не может рассматриваться отдельно в отрыве от количества активных пчел, которые влияют на разные показатели алгоритма:

1. Количество итераций + количество пчел разведчиков определяет общий диапазон охвата планов решений и таким образом повышают вероятность попадания алгоритмом в область глобального экстремума.

2. Количество итераций + количество пчел на участках определяет нахождение локального экстремума при определении одной из точек его области как перспективной.

Цель нахождения удовлетворительного решения алгоритмом не ставится, так как предполагается, что она всегда будет достигнута уже при инициализации решением задачи о камнях. Таким образом, настройки улья выполняются с целью повышения вероятности нахождения абсолютного решения и при этом сохранении минимального эффективного размера популяции.

Экспериментальным путем была выполнена настройка минимальной популяции пчелиной колонии. При этом последовательно изменялись параметры алгоритма, таким образом, чтобы для решаемых входных последовательностей результирующее значение экстремума отличалось не более чем на 0,1%. В результате были получены следующие рекомендуемые параметры: необходимо выбрать 100 пчел разведчиков и 200 итераций без изменения результата. Такие настройки позволяют охватить удобоваримое количество экстремумов. Необходим всего один лучший участок со значительным количеством пчел (выбрано 100). Выбранный диапазон изменения периода РВ-цикла при работе на участке – 10, что позволяет охватить достаточное количество решения, при этом, не пропуская локальных экстремумов. Количество выбранных участков – 10, количество пчел на каждом из участков – 25. Такие экспериментальные данные можно объяснить особенностью конкретной задачи и именно тем, что качественно на значение целевой функции влияет в большей степени выбор периода РВ-цикла среди множества возможных значений, чем их распределение по ядрам. Естественно в том случае, если выполняется условие существования расписания.

Для сконфигурированного алгоритма были проведены эксперименты для синтетичных наборов входных данных. Для 5 запусков на одной и той же последовательности алгоритмом был осуществлен переход в экстремумы с ошибкой, не превышающей 0,1%, что видно на графике зависимости номера итерации и значения наилучшего фитнесса (рис. 2).

С целью уточнения этих данных для 10 входных последовательностей рассчитана оценка среднеквадратического отклонения полученных результатов минимума функции (рис.3), значение которой свидетельствует о приемлемости разработанного алгоритма и устойчивости его работы.

Для оценки увеличения погрешности в работе алгоритма были проведены эксперименты, во время которых на вход подавались входные данные с увеличением сложности. Набор входных данных Mxn, начиная с $10x2$ увеличивался: для количества потоков с дискретностью 5, для количества ядер – 1. Для каждой входной последовательности было проведено 10 экспериментов и на основании полученных данных рассчитаны значения относительной ошибки при определении экстремума (рис.4). Из полученного графика видно, что ошибка растет в большей степени при увеличении количества ядер. Алгоритм менее чувствителен к увеличению количества потоков ЗМ. При необходимости составления расписания для специализированных вычислительных систем с количеством ядер, превышающих стандартное, необходима коррекция параметров алгоритма для увеличения показателей его точности.

Заключение

В статье рассмотрена задача оптимизации характеристик циклического вычислительного процесса для многоядерной системы. Проанализированы условия, особенности целевой функции и ограничений математической постановки задачи. Рассмотрены возможные подходы к решению задачи оптимизации. Предложен метаэвристический дискретный алгоритм с улучшенной инициализацией, разработанный на основе алгоритма искусственной пчелиной колонии. Выбраны оптимальные настройки популяции для решения поставленной задачи разработки расписания. Исследованы результаты работы алгоритмы для составленных тестовых последовательностей.

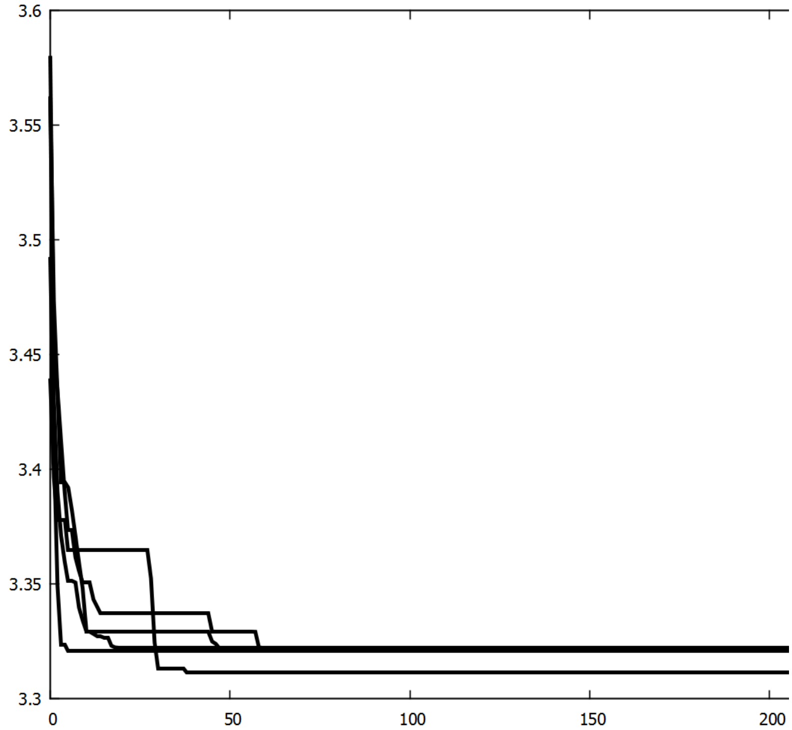

Рисунок 2 - Тестовые проходы алгоритма на одной последовательности входных данных

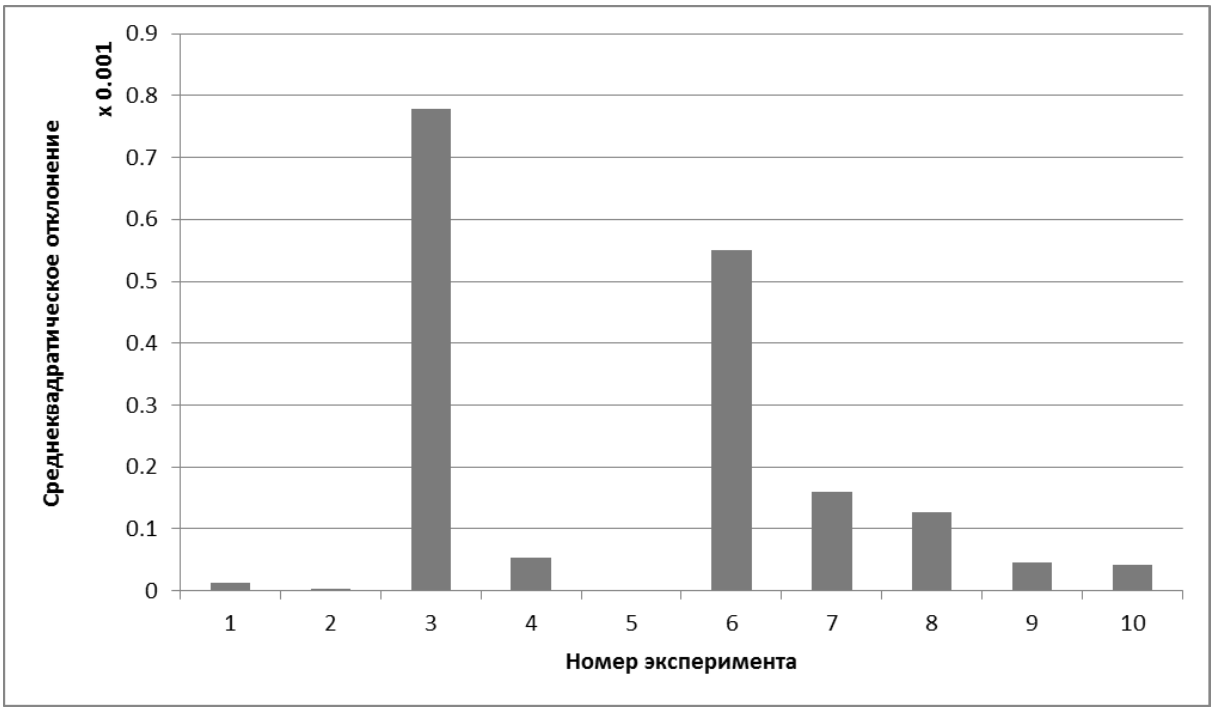

Рисунок 3 - Среднеквадратическое отклонение фитнесса для тестовых последовательностей

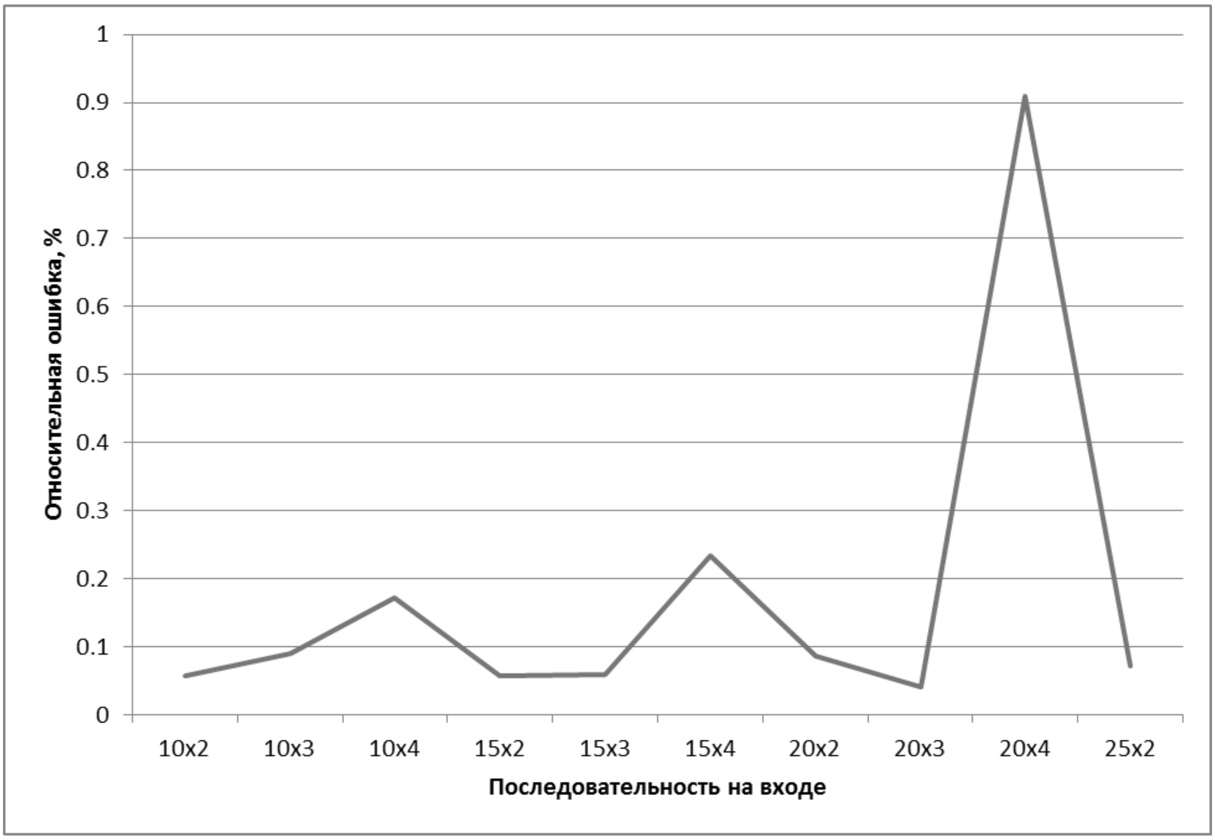

Рисунок 4 - Зависимость относительной ошибки в работе алгоритма от роста сложности набора входных данных

Список литературы

1. Layland James Scheduling Algorithms for Multiprocessing in a Hard Real-Time Environment / James W Layland, Chang L Liu. – Journal of the ACM, v.20, n.1 – New York: ACM, 1973 – P. 46–61.

2. Grossmann Ignacio E. Discrete Optimization Methods and their Role in the Integration of Planning and Scheduling / Ignacio E. Grossmann. – AICHE SYMPSIUM SERIES, v.326 – 2002 – P. 150-168

3. Иванов Ю.А. Анализ выполнения программ при моделировании динамических систем / Ю.А. Иванов // Наукові праці Донецького національного технічного університету. Серія «Проблеми моделювання та автоматизації проектування динамічних систем» (МАП-2012). Випуск 1. – Донецьк: ДВНЗ «ДонНТУ». – 2012.

4. Pham D. T. The bees algorithm – a novel tool for complex optimization problems / [Pham D. T., Ghanbarzadesh A., Koc. E, and oth]. // Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems. — 2006. — pp. 454—459.

5. Sang H. Discrete artificial bee colony algorithm for lot-streaming flowshop with total flowtime optimization / H. Sang, L. Gao, Q. Pan // Chinese journal of mechanical engineering. — 2012. — V. 25, №. 25. — pp. 990—1000.

6. Garcia E. A Discrete artificial bee colony algorithm for the multi-objective redistricting problem / [Garcia E., Gutierrez R., Velazquez P. and oth.]. — Universidad Autónoma Metropolitana, 2012. — pp. 1439—1440.

7. Garey M. R. Computers and Intractability: A guide to the theory of NP-Completeness / M. R. Garey, D. S. Johnson. — New York : W. H. Freeman & Co., 1979. — 338 p.

Надійшла до редколегії 16.05.2013

Ю.О. ІВАНОВ[1]
[1] Донецький Національний технічний університет, кафедра КІ

ОПТИМІЗАЦІЯ ОБЧИСЛЮВАЛЬНОГО ПРОЦЕСУ НА БАГАТОЯДІЕРНІЙІ СИСТЕМІ З ВИКОРИСТАННЯМ АЛГОРИТМУ ШТУЧНОЇ БДЖОЛИНОЇ КОЛОНІЇ

У статті розглянута задача розробки розкладів для ефективного керування процесом моделювання багаточастотних динамічних об'єктів в режимі реального часу на багатоядерній платформі. Методами математичного програмування дана задача класифікована яз змішана нелінійна цілочисельна. В роботі було запропоновано вирішення її з використанням метаевристичного алгоритму штучної бджолиної колонії адаптованим для необхідної дискретної задачі оптимізації. Був запропонований підхід для вирішення ініціалізації начального стану рою, як вирішення задачі про камені з використанням жадібного алгоритму. Для розглянутих наборів вхідних даних був визначений оптимальний склад популяції у алгоритмі. Була виконана оцінка роботи та надійності алгоритму, яка показала можливість застосування та високий рівень його ефективності при вирішення задачі розробку розкладу.

Ключові слова: розклад, обчислювальний процес, метаевристика, дискретна оптимізація, алгоритм штучної бджолиної колонії.

Y.O. IVANOV[1]
[1] Donetsk National Technical University

SCHEDULING OPTIMIZATION IN MULTICORE SYSTEM USING ARTIFICIAL BEE COLONY ALGORITHM

In this paper job-shop model for multicore platform was considered. The aim of the work is determination of model parameters and use them to develop schedule for hard real-time simulation. Relations in the model are set to achieve optimal control for dynamic systems including significantly different object frequencies. The model that describes computational process is complex and has several strict restrictions. In terms of mathematical programming the task is classified as a mixed integer nonlinear problem (MINLP). Given problem is multi-objective and has multiple extremas. That means finding a relative minimum is very complex for this task. Well known methods were considered: simplex method, simulated annealing, genetic algorithm, ant colony optimization and the artificial bee colony. Metaheuristic artificial bee colony algorithm was proposed for solving. This algorithm has been adapted for the solution of discrete optimization problem. In the paper a new approach was proposed to initialize the initial state of the swarm. Its value was determined as a solution of the bin packing problem using a greedy algorithm. In the worst case random initialization of a particular task may not give a feasible solutions that would satisfy the constraints posed. In addition a further bees movement to random positions also may not bring success. Therefore, as an initial value were proposed uniform basic task flow distributions across multiple cores to initialize bees' positions in the work. With this formulation, we achieved the goal of the minimum value of the load on each the core. Also this solution allows getting an optimal value of the objective function on the first step of the algorithm. The optimal bee population for algorithm was defined for the known data sets. Experiment showed that algorithm is less sensitive to increase number of threads in simulation problem. In a case when required a schedule for specialized computer systems with more cores than the standard should be adjusted in the parameters of the algorithm to increase the performance and its accuracy. The reliability and feasibility of the algorithm were estimated for input test sequences. Test runs of algorithm demonstrated its applicability and high efficiency schedules as a result.

Keywords: schedule, computational process, metaheuristic, discrete optimization, artificial bee colony algorithm.