

УДК 004.658

С.В. Лаздынь, С.Ю. Землянская

Донецкий национальный технический университет, г. Донецк,
кафедра автоматизированных систем управления
E-mail: slazd@ukr.net, zsaa@ya.ru

ОПТИМИЗАЦИЯ РАСПРЕДЕЛЁННЫХ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ И ОБЪЕКТНОГО МОДЕЛИРОВАНИЯ

Abstract

Lazdyn S.V., Zemlianskaya S.Y. Distributed corporative information systems optimization based on genetic algorithm and object modeling. The distributed corporative information systems (DCIS) optimization problem is formulated. The maximal system response time is chosen as an optimality criterion. The new approach to DCIS optimal parameters determination, based on a joint use of DCIS object model and modified genetic algorithm, is proposed.

Keywords: distributed corporate informative system, optimization, criterion, object, model, genetic algorithm, multichromosome, fitness-function, population.

Анотація

Лаздынь С.В., Землянська С.Ю. Оптимізація розподілених корпоративних інформаційних систем з використанням генетичних алгоритмів і об'єктного моделювання. Сформульовано проблему оптимізації розподілених корпоративних інформаційних систем (РКИС). Критерієм оптимізації обрано максимальний час відповіді системи на запит користувача. Для визначення оптимальних параметрів і конфігурації РКИС запропоновано новий підхід, який базується на взаємодії об'єктної моделі РКИС і генетичного алгоритму.

Ключові слова: розподілена корпоративна інформаційна система, оптимізація, критерій, об'єкт, модель, генетичний алгоритм, мультихромосома, фітнес-функція, популяція.

Аннотация

Лаздынь С.В., Землянская С.Ю. Оптимизация распределённых корпоративных информационных систем с использованием генетических алгоритмов и объектного моделирования. Сформулирована проблема оптимизации распределённых корпоративных информационных систем (РКИС). В качестве критерия оптимизации выбрано максимальное время ответа системы на пользовательский запрос. Для определения оптимальных параметров и конфигурации РКИС предложен новый подход, основанный на взаимодействии объектной модели РКИС и генетического алгоритма.

Ключевые слова: распределённая корпоративная информационная система, оптимизация, критерий, объект, модель, генетический алгоритм, мульти хромосома, фитнес-функция, популяция.

Общая формулировка проблемы оптимизации РКИС

Распределённые корпоративные информационные системы — это один из сложных видов компьютерных систем, которые создаются для крупных промышленных предприятий и различных корпораций для организации информационного обмена о результатах деятельности и, самое главное, — для оперативного принятия управляющих решений. Крупные современные корпорации, как правило, объединяют несколько предприятий (организаций), которые территориально могут размещаться в разных городах или областях страны, то есть являются распреде-

лєнными. Связь между компьютерами в таких системах обеспечивается при помощи локальных и глобальных вычислительных сетей, а данные хранятся в распределєнных базах данных.

Проектирование и внедрение (или модернизация) РКИС требует значительных капиталовложений, временных и трудовых затрат коллективов квалифицированных специалистов. При этом возникают проблемы, связанные, с одной стороны с ограниченным финансированием создаваемой (или модифицируемой) РКИС, а с другой стороны — с необходимостью обеспечения эффективного функционирования РКИС.

Таким образом, задача определения оптимальных параметров РКИС, необходимых для достижения максимальной производительности системы при ограниченных финансовых затратах, требует проведения дальнейших исследований. Указанную задачу можно решить путєм использования современных методов моделирования и оптимизации [5],[6].

Краткий анализ проведенных исследований по оптимизации РКИС.

В проведенных ранее исследованиях, в основном создавались модели для несложных компьютерных систем, построенных на локальных сетях, или же для построения моделей более сложных систем применялся ряд допущений, Для оптимизации таких систем использовались традиционные методы исследования операций: линейное, динамическое и дискретное программирование, различные модификации этих методов [3–5]. Применение таких методов для оптимизации РКИС ограничено большой размерностью, неоднородностью структуры РКИС, а также вычислительной сложностью расчєтов. Построенные ранее аналитические или имитационные модели не могут быть использованы для моделирования РКИС в связи с их сложностью, громоздкостью или наличием ряда допущений, снижающих точность результатов, полученных при использовании таких моделей.

Выбор оптимальной или «близкой» к оптимальной структуры такой системы можно осуществить с помощью одного из подходов научного направления Natural Computing, основанного на принципах природных механизмов принятия решений и включающего нейросетевые вычисления, жадные алгоритмы, муравьиные алгоритмы, табуированный поиск, генетические алгоритмы и другие [5].

Муравьиные и жадные алгоритмы хорошо себя зарекомендовали при оптимизации систем, характеризуемых набором однородных параметров (РКИС описывается большим количеством несвязанных параметров) или же если необходимо выбрать оптимальную последовательность действий, в то время как в РКИС все события имеют случайный характер.

Табуированный поиск (Tabu Search) представляет собой вариацию известного метода градиентного спуска с памятью. В процессе поиска ведєтся список табуированных (запрещєнных для перехода) позиций из числа уже рассчитанных. Критическими параметрами алгоритма является диапазон запретов. В процессе поиска осуществляются операции включения в запрещєнный список состояний вокруг текущего состояния, что добавляет фактор случайности в процесс поиска. Для оптимизации РКИС этот метод труднореализуем в связи с необходимостью определения множества состояний вокруг текущего состояния, которое характеризуется большим количеством не связанных параметров

Генетические алгоритмы, используя аналогию между естественным отбором и процессом выбора наилучшего решения из множества возможных, являются одним из самых распространєнных вариантов реализации эволюционных вычислений. Моделируя отбор лучших вариантов как процесс эволюции в популяции особей, можно получить решение задачи оптимизации, задав начальные условия эволюционного процесса. Преимуществом генетических алгоритмов перед другими является простота их реализации, относительно высокая скорость работы, параллельный поиск решения сразу несколькими особями, позволяющий избежать попадания в «ловушку» локальных оптимумов (нахождения первого попавшегося, но не самого удачного оптимума). Недостаток — сложность выбора схемы кодирования, то есть выбора параметров и вида их кодирования в «хромосомах», возможность вырождения популяции, сложность описания ограничєний планирования.

Выбор критерия эффективности РКИС

Основная цель создания РКИС — обеспечение возможности оперативного принятия управляющих решений, на основании данных о состоянии дел в корпорации (с учётом текущей финансовой и производственной ситуации), доступа пользователей к необходимым данным и предоставление возможности быстрого и безопасного обмена данными в рамках некоторой корпоративной организации. Таким образом, одно из главных требований к РКИС — обеспечение наиболее быстрой реакции на запрос пользователя. Следовательно, критерий эффективности функционирования РКИС можно представить как время реакции системы на запросы пользователей.

Рассмотрим, как определяется время реакции системы. Запросы, как правило, формируются пользователями на рабочих станциях и затем передаются по каналам связи узлам обработки, которые могут включать сервер баз данных и сервер приложений для обработки сложных запросов, после чего ответ по каналам связи передаётся пользователю. Тогда время выполнения запроса пользователя T_{resp} определяется следующим образом:

$$T_{resp} = T_t + T_p + T_w + \max_{i=1, Nsq} \{Tsq_i\} \tag{1}$$

$$Tsq_i = Tt_i + Tp_i + Tw_i$$

$$Tt_i = \sum_{j=1}^{Ncq_i} (Vq_i + Va_i) / S_j$$

$$Tp_i = Tpa_i + Tpdb_i$$

$$Tpa_i = Ca_i / Psa_{qi} + \max_{j=1, Nsq_i} \{Tsq_j\}$$

$$Tpdb_i = Cdb_i / Psdb_{qi} + \max_{j=1, Nsq_i} \{Tsq_j\}$$

$$Tw_i = Twt_i + Twp_i$$

где Tsq_i — время выполнения i -го подзапроса, T_t — время передачи запроса, Tt_i — время передачи

i -го подзапроса, T_p — время обработки запроса, Tp_i — время обработки i -го подзапроса, Tpa — время выполнения запроса (подзапроса) сервером приложений, Tpa_i — время выполнения i -го подзапроса сервером приложений, $Tpdb$ — время выполнения запроса сервером баз данных, $Tpdb_i$ — время выполнения i -го подзапроса сервером баз данных, Ncq_i — количество каналов связи между источником и обработчиком i -го подзапроса, S_j — пропускная способность j -го канала связи (кбит/с), Vq_i — объём i -го подзапроса (байт), Va_i — объём ответа на i -й подзапрос (байт), Ca_i — сложность i -го подзапроса при обработке подзапроса сервером приложений (JOPS), Cdb_i — сложность i -го подзапроса при выполнении подзапроса сервером баз данных (TPC), Nsq — количество подзапросов данного запроса, Nsq_i — количество подзапросов i -го подзапроса, Psa_{qi} — производительность сервера приложений, обрабатывающего i -й подзапрос (JOPS/с), $Psdb_{qi}$ — производительность сервера баз данных, обрабатывающего i -й подзапрос (TPC/с), T_w — время задержки при передаче и обработке запроса, Tw_i — время задержки при передаче и обработке i -го подзапроса, Twt — время задержки при передаче запроса по каналам связи (с), Twt_i — время задержки при передаче по каналам связи i -го подзапроса (с), Twp — время ожидания в очереди запросов к серверу, Twp_i — время ожидания в очереди запросов к серверу.

При оценке производительности компьютерной системы по скорости реакции используются критерии двух типов: средневзвешенные и пороговые. Средневзвешенный критерий представляет собой сумму времен реакции на все пользовательские запросы корпоративной системы, которые требуют реакции системы (запросы на чтение), то есть сумму вида

$$Tr_A = \sum_{i=1}^{Nq} Tresp_i / Nq, \tag{2}$$

где $Tresp_i$ — время реакции системы на i -ый запрос пользователя, Nq — число различных пользовательских запросов в системе.

Пороговый критерий отражает наихудшее время реакции системы на запрос пользователя:

$$Tr_L = \max_{i=1, Nq} \{Tresp_i\} \quad (3)$$

Пороговый критерий оптимизации (Tr_L) гарантирует всем пользователям некоторый удовлетворительный уровень реакции системы на их запросы. Средневзвешенный критерий (Tr_A) может дискриминировать некоторых пользователей, для которых время реакции слишком велико при том, что при усреднении получен вполне приемлемый результат.

Таким образом, в качестве критерия оптимальности РКИС выбираем пороговый критерий (3).

Постановка задачи оптимизации РКИС

Пусть задана распределённая корпоративная информационная система со следующими параметрами:

$\{MN_n\}$ — множество узлов РКИС; $n = \overline{1, Nn}$, где Nn — количество узлов РКИС;

$\{MSdb_i\}$ — множество серверов баз данных; $i = \overline{1, Ndb}$, где Ndb — количество серверов БД;

$\{MSa_j\}$ — множество серверов приложений, размещённых в узлах РКИС; $j = \overline{1, Nsa}$, где Nsa — количество серверов приложений;

$\{MWS_k\}$ — множество рабочих станций, размещённых в узлах РКИС; $k = \overline{1, Nws}$, где Nws — количество серверов БД;

$\{MC_l\}$ — множество каналов связи, соединяющих между собой узлы РКИС; $l = \overline{1, Nc}$, где Nc — количество каналов;

$\{MNd_n\}$ — множество сетевых устройств, размещённых на узлах РКИС, $n = \overline{1, Nnd}$, где Nnd — количество сетевых устройств;

$\{MDS_m\}$ — множество наборов данных; $m = \overline{1, Nds}$, где Nds — количество наборов данных БД;

$\{MA_p\}$ — множество приложений; $p = \overline{1, Na}$, где Na — количество приложений, используемых в РКИС;

$\{MQ_q\}$ — множество запросов, $q = \overline{1, Nq}$, где Nq — количество запросов, используемых в РКИС.

Структура РКИС (территориальное расположение серверов, рабочих станций в узлах РКИС и каналы связи между ними) задана следующим образом:

а) соответствие между узлами и расположенными на них рабочими станциями, серверами и сетевыми устройствами, то есть между множествами $\{MSa_j\}$, $\{MSdb_i\}$, $\{MWS_k\}$, $\{MNd_n\}$ и $\{MN_n\}$ такое, что каждому элементу множеств $\{MSa_j\}$, $\{MSdb_i\}$ и $\{MWS_k\}$ поставлен в соответствие один из элементов множества $\{MN_n\}$ и $\{MNd_n\}$, а каждому элементу множества $\{MNd_n\}$ поставлен в соответствие один из элементов множества $\{MC_l\}$:

$$\forall MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MN_{nsa} \in \{MN_n \mid n = 1, Nn\}$$

$$\forall MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MNd_{dsa} \in \{MNd_d \mid d = 1, Nnd\}$$

$$\forall MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \exists MN_{ndb} \in \{MN_n \mid n = 1, Nn\}$$

$$\forall MSdb_i \in \{MSdb_i \mid i = 1, Ndb\} \exists MNd_{ddb} \in \{MNd_d \mid d = 1, Nnd\}$$

$$\forall MWS_k \in \{MWS_k \mid k = 1, Nws\} \exists MN_{nws} \in \{MN_n \mid n = 1, Nn\}$$

$$\forall MWS_k \in \{MWS_k \mid k = 1, Nws\} \exists MNd_{dws} \in \{MNd_d \mid d = 1, Nnd\}$$

$$\forall MNd_d \in \{MNd_d \mid d = 1, Nnd\} \exists MN_{ndd} \in \{MN_n \mid n = 1, Nn\}$$

$$\forall MNd_d \in \{MNd_d \mid d = 1, Nnd\} \exists MC_{ld} \in \{MC_l \mid l = 1, Nc\}$$

б) соответствие между рабочими станциями РКИС и запросами, порождаемыми этими рабочими станциями), то есть между множествами $\{MQ_q\}$ и $\{MWS_k\}$ такое, что каждому элементу множеств $\{MQ_q\}$ поставлен в соответствие один из элементов множества $\{MWS_k\}$:

$$\forall MQ_q \in \{MQ_q\} \exists MWS_{kq} \in \{MWS_k \mid k = 1, Nws\}$$

Пусть заданы:

$\{MTbd_r\}$ — множество типов серверов БД; $r = \overline{1, Ntsd}$, где $Ntsd$ — количество типов серверов БД;

$\{MTA_s\}$ — множество типов серверов приложений; $s = \overline{1, Ntsa}$, где $Ntsa$ — количество типов серверов приложений;

$\{MTC_t\}$ — множество типов каналов связи; $t = \overline{1, Ntc}$, где Ntc — количество типов каналов связи.

$\{MTN_u\}$ — множество типов сетевых устройств; $u = \overline{1, Ntn}$, где Ntn — количество типов каналов связи.

Конфигурация РКИС определяется соотношениями (4) и (5):

1) для каждого из узлов заданы типы серверов баз данных, серверов приложений, сетевых устройств и каналов связи такие, что обеспечиваются соответствия между множествами:

$$\{MSdb_i\} \text{ и } \{MTbd_r\}, \text{ в котором } \forall MSdb_i \in \{MSa_i \mid i = 1, Ndb\} \exists MTbd_{ri} \in \{MTbd_r \mid r = 1, Ntdb\} \quad (4)$$

$$\{MSa_j\} \text{ и } \{MTA_s\}, \text{ в котором } \forall MSa_j \in \{MSa_j \mid j = 1, Nsa\} \exists MTA_{sj} \in \{MTA_j \mid j = 1, Nta\}$$

$$\{MC_l\} \text{ и } \{MTC_t\}, \text{ в котором } \forall MC_l \in \{MC_l \mid l = 1, Nc\} \exists MTC_{tl} \in \{MTC_t \mid t = 1, Ntc\}$$

$$\{MNd_n\} \text{ и } \{MTN_u\}, \text{ в котором } \forall MNd_n \in \{MC_n \mid n = 1, Nnd\} \exists MTN_{un} \in \{MTC_u \mid u = 1, Ntn\}$$

где каждому элементу первого множества ставится в соответствие только один из элементов второго множества соответствующей пары.

2) задано размещение наборов данных на серверах баз данных и приложений на серверах приложений, то есть определено подмножество серверов баз данных системы $\{MLD_d \mid d=1, mi\}$ и подмножество серверов приложений $\{MLA_a \mid a=1, pj\}$ такие, что обеспечиваются соответствия между множествами:

$$\{MDS_m\} \text{ и } \{MSdb_i\}, \text{ в котором} \quad (5)$$

$$\forall MDS_m \in \{MDS_m \mid m = 1, Nds\} \exists \{MLD_d \mid d = 1, mi\}, MLD_d \in \{MSdb_i \mid i = 1, Ndb\}, 1 \leq mi \leq Ndb$$

$$\{MA_p\} \text{ и } \{MSa_j\}, \text{ в котором}$$

$$\forall MA_p \in \{MA_p \mid p = 1, Na\} \exists \{MLA_a \mid a = 1, pj\}, MLA_a \in \{MSa_j \mid j = 1, Nsa\}, 1 \leq pj \leq Nsa$$

где, каждому элементу первого множества ставится в соответствие хотя бы один элемент второго множества, но не больше Nsa и Ndb соответственно.

Пусть $\{DCIS_p \mid p=1, NP\}$ — множество всех возможных вариантов конфигураций РКИС, где NP — количество вариантов построения РКИС. Необходимо найти такой вариант построения РКИС $DCIS_x$ среди всех возможных вариантов, для которого критерий эффективности (3) имеет минимальное значение:

$$Tr_{opt} = \min_{\{DCIS_p \mid p=1, NP\}} \{ \max_{i=1, Nq} \{Tresp_i\} \} \quad (6)$$

где Nq — количество пользовательских запросов на чтение, обработанных в системе.

При этом необходимо учитывать ряд ограничений, накладываемых используемыми в системе данными и приложениями, а также финансовыми возможностями организации, в рамках которой функционирует или проектируется РКИС. Рассмотрим эти ограничения:

1. Ограничение на размещение данных. Объём дискового пространства помещённого в данном узле сервера БД должен быть достаточным для размещения необходимых наборов данных:

$$\sum_{j=1}^{Nds} Vds_j \leq Vd_i \quad \forall MSdb_i \in \{MSdb_i \mid i = \overline{1, Ndb}\} \quad (7)$$

где Ndb — количество серверов БД в системе, Vd_i — объём дискового пространства i -го сервера,

Nds_i — количество наборов данных, размещённых на i -м сервере, Vds_j — объём j -го набора данных.

2. Каждый из наборов данных должен быть размещён хотя бы на одном сервере баз данных

$$\forall MDs_j \in \{MDs_j \mid j = \overline{1, Nds}\} \exists MSdb_i \in \{MSdb_i \mid i = \overline{1, Ndb}\} \quad (8)$$

3. Объём оперативной памяти серверов приложений должен быть достаточным для выполнения размещённых на нём приложений:

$$\sum_{j=1}^{Nai} Va_j + Vos_i \leq Vram_i \quad \forall MSa_j \in \{MSa_j \mid j = \overline{1, Nsa}\} \quad (9)$$

где Nsa — количество серверов приложений, $Vram_i$ — объём оперативной памяти i -го сервера, Nai — количество приложений, запускаемых на i -м сервере, Va_j — размер j -го приложения, Vos_i — объём оперативной памяти, занимаемый операционной системой i -го сервера.

4. Стоимость аппаратно-технического обеспечения системы не должна превысить сумму, выделенную организацией на закупку аппаратных средств:

$$\sum_{i=1}^{Nsa} Ssa_i + \sum_{j=1}^{Ndb} Ssdb_j + \sum_{k=1}^{Nws} Sws_k + \sum_{l=1}^{Nnd} Snd_l + \sum_{m=1}^{Nc} Sc_m \leq Sts, \quad (10)$$

где Nsa — количество серверов приложений, Ndb — количество серверов БД, Nws — количество рабочих станций, Nnd — количество сетевых устройств, Nc — количество каналов передачи, Ssa_i — стоимость i -го сервера приложений, $Ssdb_i$ — стоимость i -го сервера БД, Sws_i — стоимость i -й рабочей станции, Snd_i — стоимость i -го сетевого устройства, Sc_i — стоимость i -го канала связи, Sts — сумма, выделенная на закупку технических средств.

Предлагаемый подход к оптимизации РКИС

Сформулированная выше задача построения оптимальной конфигурации РКИС относится к NP-полным задачам перебора [2]. Данная задача не может быть решена классическими аналитическими методами в связи с большим количеством и разнородностью входных параметров, а также невозможностью точного определения значения целевой функции в связи со случайностью процессов, происходящих в системе. Поэтому для оптимизации параметров РКИС предлагается использовать новый подход, основанный на совместном использовании объектной модели РКИС [8] и аппарата генетических алгоритмов (ГА) [9], суть которого состоит в следующем.

Начальный вариант РКИС формируется на основе информации о её структуре. Параметры отдельных компонентов РКИС (типы серверов, рабочих станций, каналов связи, а также размещение наборов данных на серверах баз данных и приложений на серверах приложений) кодируются в виде хромосом ГА. Популяция ГА представляет собой набор некоторых точек пространства поиска.

В процессе оптимизации с помощью операторов ГА генерируются хромосомы, то есть различные варианты построения системы. Полученные схемы являются исходной информацией для объектной модели. Имитационное моделирование работы РКИС на основе объектной модели позволяет получить оценки критериев эффективности РКИС (6). Эти оценки, в свою очередь, являются значениями функции приспособленности ГА (фитнесс-функции) для данного варианта решения. Этот процесс повторяется до тех пор, пока не будет достигнут критерий останова генетического алгоритма.

Схема взаимодействия объектной модели РКИС с ГА представлена на рис. 1.

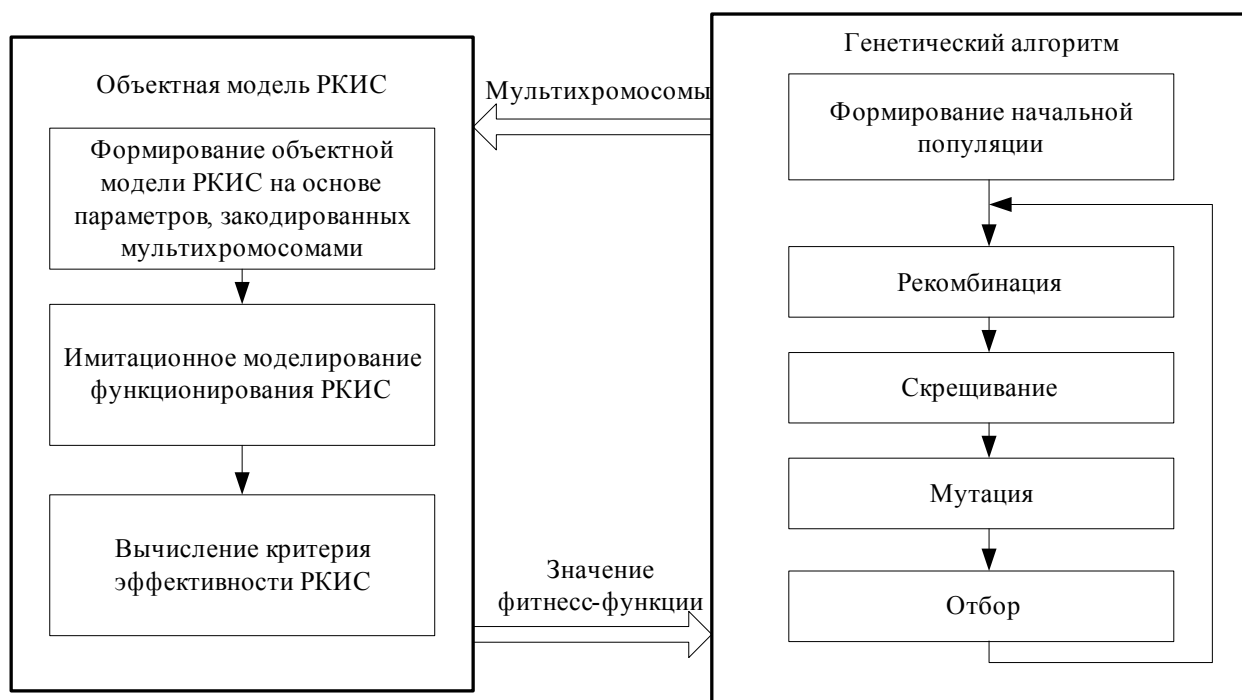


Рисунок 1 — Схема взаимодействия генетического алгоритма и объектной модели при оптимизации РКИС

Объектная модель РКИС

Для построения имитационной модели РКИС был выбран объектно-ориентированный подход в связи с тем, что он позволяет представить РКИС в виде совокупности объектов заранее созданных классов, включающих свойства описываемых объектов методы, реализующие взаимодействие объектов системы. Такой подход позволяет строить и модифицировать модели РКИС произвольной сложности.

Проведенный анализ показал, что в составе РКИС можно выделить следующие группы типовых компонентов, параметры которых влияют на производительность системы: компоненты технического обеспечения (компьютерное и сетевое оборудование), программное обеспечение, данные. Для этих типовых компонентов разработаны классы объектов.

Компоненты технического обеспечения:

- Рабочая станция (класс TWorkstation) — место формирования пользовательских запросов;
- Сервер приложений (класс TAppserver) — сервер, хранящий и выполняющий сложные приложения, производительность сервера оценивается в условных единицах –JOP — количество тестовых операций, выполняемых в секунду;
- Сервер БД (класс TDBServer) — используется для обработки и чтения/записи информации, производительность такого сервера измеряется в условных единицах TPC — количество транзакций в секунду;
- Узел компьютерной системы (класс TNode) — совокупность серверов и /или рабочих станций и сетевых устройств, территориально расположенных в одном месте;
- Сетевое устройство (класс TNetDevice) — коммутатор, маршрутизатор;
- Канал передачи (класс TChannel) — физическая среда передачи данных, может представлять собой коммутируемый канал связи, выделенную телефонную линию, оптоволоконный кабель, беспроводный канал связи и др.

Компоненты программного обеспечения:

– Приложение (класс TCISApplication) — пользовательское приложение, запускаемое с определённой периодичностью на рабочей станции, характеризуется набором формируемых запросов. Если приложение находится на сервере приложений, то запускается оно по запросу с рабочей станции;

– Запрос (класс TQuery) (на модификацию (класс TModifyQuery), чтение (класс TReadQuery), обновление данных (класс TRefreshQuery)). Запрос формируется к конкретному набору данных; место обработки запроса определяется распределением набора данных между серверами. Каждый запрос может инициировать последовательность других запросов (подзапросов).

Данные:

– Набор данных (класс TDataSet) — логический набор данных, который имеет название и характеризуется набором серверов, на которых он физически размещается.

Для унификации объектов модели введен абстрактный класс TBaseObject, включающий свойства, общие для всех объектов модели: идентификатор объекта (ID) и его наименование (Name). Все остальные классы модели являются производными от класса TBaseObject и наследуют эти свойства. На рис. 2 приведена UML-диаграмма иерархии классов объектной модели РКИС.

Модель РКИС представляет собой совокупность взаимодействующих объектов разработанных классов. В модели реализован алгоритм моделирования, управляемый событиями, при котором в качестве следующего значения часов модельного времени выбирается минимальное время события из списка событий [8].

Модификация генетического алгоритма для оптимизации РКИС

При использовании генетического алгоритма для оптимизации РКИС необходимо кодировать большое количество разнородных параметров РКИС, таких как серверы, сетевые устройства, каналы связи, наборы данных. В связи с этим применение классического генетического алгоритма для определения оптимальных параметров РКИС не представляется возможным. Рассмотрим предлагаемые модификации способов кодирования хромосом и реализации операторов генетического алгоритма, учитывающие специфику решаемой задачи.

Кодирование структуры и параметров РКИС

Информацию о структуре РКИС предлагается кодировать в виде набора из 4 мультихромосом, которые отображают:

1. Параметры технических средств типы (серверов баз данных и приложений, сетевых устройств) на узлах РКИС;
2. Типы каналов связи, соединяющих узлы системы;
3. Размещение приложений в узлах системы;
4. Размещение наборов данных в узлах системы.

Хромосомы, кодирующие типы серверов БД, серверов приложений, каналов связи и сетевых устройств, имеют однородную структуру, поэтому рассмотрим структуру для кодирования в системе условного устройства, различные типы которого используются узлами системы.

Пусть имеем r типов устройства U , которое нужно разместить в n узлах системы. Расположим типы устройства по порядку, тогда каждый тип устройства кодируется двоичной комбинацией (порядковый номер данного типа), длина p которой определяется по формуле:

$$p = \lceil \log_2 r \rceil \quad (11)$$

На узле S может использоваться несколько различных устройств: $U_1 \dots U_k$, каждое из которых может иметь определённый тип, причём количество типов каждого из устройств обозначим $r_1 \dots r_k$ соответственно. Тогда использование различных устройств узлом S системы можно представить в виде битовой строки:

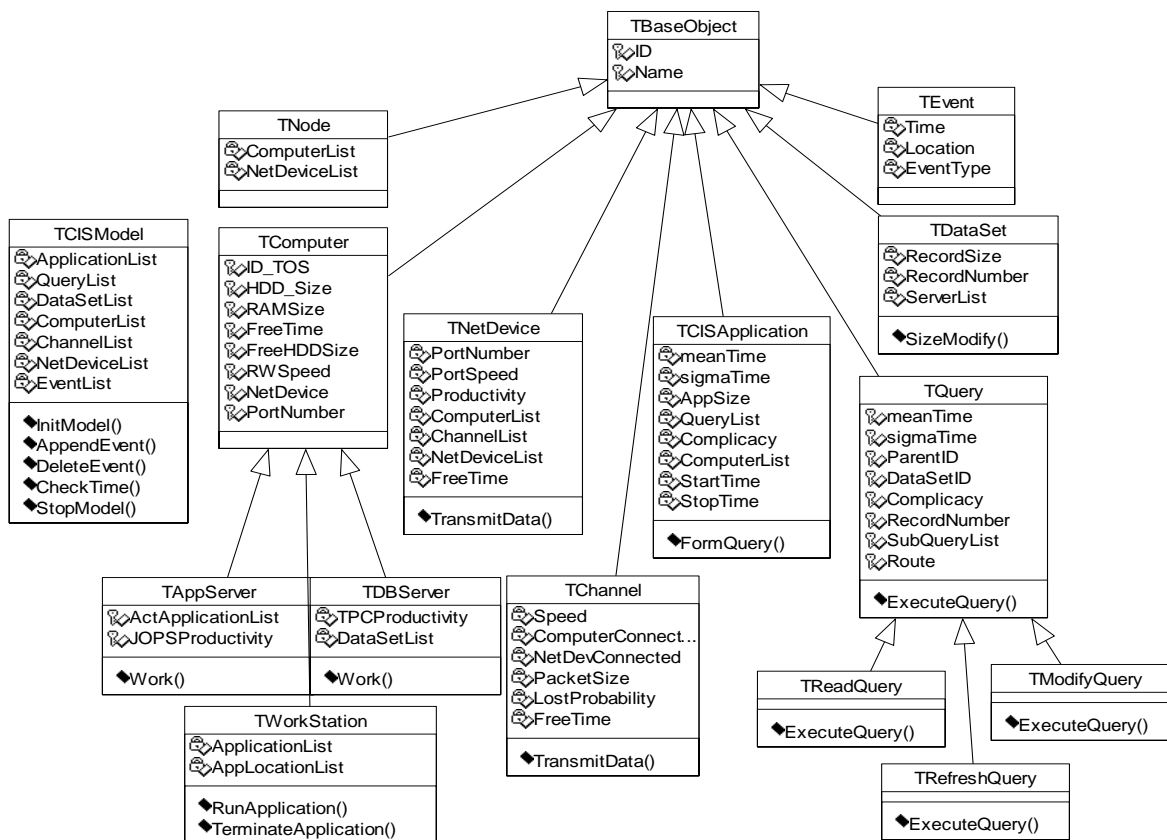


Рисунок 2 — UML-диаграмма иерархии классов объектной модели РКИС

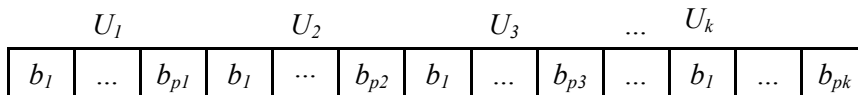


Рисунок 3 — Кодирование использования условного устройства U узлами РКИС

Здесь $(b_1...b_{p_i})$ — двоичное представление порядкового номера типа устройства U_i из общего списка. Если на каком-либо узле отсутствует устройство U_i , то соответствующая этому устройству битовая строка заполняется нулями.

Если РКИС состоит из Nn узлов, то для кодирования структуры РКИС потребуется Nn хромосом, каждая из которых описывает состав технических средств (типы используемого сервера БД, сервера приложений, сетевого устройства) определённого узла системы. Таким образом, получаем мультихромосому Xt структура, которой представлена на рис. 4.

Здесь p_1, p_2 и p_3 вычисляются в соответствии с формулой (11):

$$p_1 = \log_2 Ndb, p_2 = \log_2 Nsa, p_3 = \log_2 Nnd$$

где Ndb — количество типов серверов БД, Nsa — количество типов серверов приложений; Nnd — количество типов сетевых устройств, p_1, p_2 и p_3 — длины битовых строк, используемых для кодирования типов серверов БД, серверов приложений и сетевых устройств соответственно.

| Узел | Тип сервера БД | | | Тип сервера приложений | | | Тип сетевого устройства | | |
|----------|----------------|-----|-----------|------------------------|-----|-----------|-------------------------|-----|-----------|
| | b_{11} | ... | b_{1p1} | b_{11} | ... | b_{1p2} | b_{11} | ... | b_{1p3} |
| S_1 | b_{11} | ... | b_{1p1} | b_{11} | ... | b_{1p2} | b_{11} | ... | b_{1p3} |
| S_2 | b_{21} | ... | b_{2p1} | b_{21} | ... | b_{2p2} | b_{21} | ... | b_{2p3} |
| ... | | ... | | | ... | | | ... | |
| S_{Nn} | b_{n1} | ... | b_{np1} | b_{n1} | ... | b_{np2} | b_{n1} | ... | b_{np3} |

Рисунок 4 — Структура мультихромосомы Xt для кодирования параметров устройств, используемых узлами РКИС

Типы каналов связи, соединяющих между собой узлы РКИС, закодируем в виде матрицы, строками и столбцами в которой выступают узлы системы, а на пересечении строки и столбца зададим тип канала связи (в виде битовой строки), используемого между этими узлами. Получим мультихромосому Xc , структура которой приведена на рис. 5.

| Узлы \ Узлы | S_1 | S_2 | ... | S_{Nn} |
|-------------|-----------|-----------|-----|------------|
| S_1 | C_{11} | C_{12} | ... | C_{1Nn} |
| S_2 | C_{21} | C_{22} | ... | C_{2Nn} |
| ... | ... | ... | ... | ... |
| S_{Nn} | C_{Nn1} | C_{Nn2} | ... | C_{NnNn} |

Рисунок 5 — Структура мультихромосомы Xc для кодирования типов каналов связи между узлами РКИС

Здесь C_{ij} — битовая строка, задающая тип канала связи между узлами S_i и S_j :

$$C_{ij}=(c_1...c_{ct}), C_{ii}=(0...0)$$

где $ct = \log_2 Ntc$, Ntc — количество типов каналов связи, используемых в РКИС.

Размещение приложений на серверах приложений также целесообразно закодировать в виде мультихромосомы. Пусть в системе используется Na приложений. Тогда размещение приложений на сервере приложений Sa_i кодируется битовой строкой длины Na . Использование Na приложений Nsa серверами кодируется Nsa строками длиной Na , т.е. получаем мультихромосому. Вид полученной мультихромосомы представлен на рис. 6.

| Серверы приложений \ Приложения | A_1 | A_2 | ... | A_{Na} |
|---------------------------------|------------|------------|-----|-------------|
| Sa_1 | a_{11} | a_{12} | ... | a_{1Na} |
| Sa_2 | a_{21} | a_{22} | ... | a_{2Na} |
| ... | ... | ... | ... | ... |
| Sa_{Nsa} | a_{Nsa1} | a_{Nsa2} | ... | a_{NsaNa} |

$$a_{ij} = \begin{cases} 1, & \text{если приложение } A_j \text{ используется } i\text{-м сервером приложений} \\ 0, & \text{если приложение } A_j \text{ не используется } i\text{-м сервером приложений} \end{cases}$$

Рисунок 6 — Структура мультихромосомы Xa для кодирования размещения приложений на серверах приложений РКИС

Кодирование размещения наборов данных (НД) на серверах БД представлено мультихромосомой Xd на рис. 7.

| | | | | |
|------------------|------------|------------|-----|--------------|
| Наборы данных | DS_1 | DS_2 | ... | DS_{Nds} |
| Серверы БД | | | | |
| Sdb_1 | d_{11} | d_{12} | ... | d_{1Nds} |
| Sdb_2 | d_{21} | d_{22} | ... | d_{2Nds} |
| ... | ... | ... | ... | ... |
| Sdb_{Ndb} | d_{Ndb1} | d_{Ndb2} | ... | d_{NdbNds} |

$$d_{ij} = \begin{cases} 1, & \text{если } DS_j \text{ размещён на } i\text{-м сервере} \\ 0, & \text{если } DS_j \text{ отсутствует на } i\text{-м сервере} \end{cases}$$

Рисунок 7 — Структура мультихромосомы X_d для кодирования размещения наборов данных на серверах БД

Здесь Nds — количество наборов данных РКИС; Ndb — количество серверов БД в РКИС; Sdb_i — i -й сервер БД; DS_j — j -й набор данных.

Таким образом, конфигурация РКИС кодируется набором из 4 мультихромосом, определяющих типы серверов и сетевых устройств на узлах системы, типы каналов связи между узлами, размещение приложений системы на серверах приложений и размещение наборов данных на серверах БД.

Генетические операторы

Оператор отбора

Стратегия отбора является составной частью ГА и определяет особи для скрещивания. Из популяции, содержащей N особей, выбирается случайным образом t особей и две лучшие из них, для которых значение фитнес-функции максимально, затем используются для скрещивания.

Оператор рекомбинации

Данный оператор реализует обмен соответствующими (гомологичными) хромосомами между парами особей с заданной вероятностью P_{k1} , что соответствует обмену серверами и сетевыми устройствами различных типов на одном узле между двумя модификациями РКИС, а также изменению размещений наборов данных и приложений на серверах баз данных и приложений. Выбор родительских особей для рекомбинации осуществляется с помощью оператора отбора. Из каждой пары родительских особей формируется пара дочерних путём обмена гомологичными хромосомами между родительскими особями. Обмен происходит с вероятностью P_{k1} .

Оператор скрещивания реализует обмен участками хромосом между родительскими особями, что соответствует изменению типов серверов, сетевых устройств, используемых в узле РКИС, типов каналов связи между узлами, а также наборов данных и приложений, размещаемых на серверах в новом варианте РКИС в соответствии с одним из родительских вариантов.

Особенность оператора скрещивания заключается в том, что применяются различные модификации оператора скрещивания к различным мультихромосомам, а именно: к мультихромосоме, кодирующей типы устройств применяется двухточечный оператор скрещивания, а к остальным — однородный. Скрещивание производится с вероятностью P_{k2} ,

Оператор мутации

Оператор мутации применяется ко всем членам промежуточной популяции, сформированной с помощью оператора скрещивания. Оператор мутации с вероятностью P_{mut} инвертирует биты в хромосоме каждой особи. При применении оператора мутации необходимо проверять полученное решение на соответствие условиям (5)–(8).

Оператор селекции

После скрещивания особей необходимо решить, какие из новых особей войдут в следующее поколение. Создается промежуточная популяция, которая включает в себя как родителей, так и их потомков. Члены этой популяции оцениваются, а затем из них выбираются N самых лучших, которые и войдут в следующее поколение.

Алгоритм оптимизации конфигурации РКИС

Каждая мультихромосома в многохромосомном представлении структуры РКИС отвечает за определённый аспект решения (параметры технических средств системы, размещение наборов данных или приложений). Это позволяет комбинировать наборы мультихромосом в одном решении, что является приближением к естественной эволюции. С другой стороны, представление решения набором мультихромосом дает возможность организации поиска решений в различных постановках, то есть по определённому набору параметров, оставляя отдельные виды мультихромосом неизменными в процессе генетического поиска. Фиксация отдельных хромосом приводит к сужению пространства поиска, при этом возможна потеря оптимальных решений. Поэтому представляется целесообразным комбинирование отдельных постановок при поиске оптимального решения. В общем случае возможны три подхода к комбинированию постановок: последовательный, параллельный и параллельно-последовательный [9, 10].

При последовательном подходе на каждом этапе осуществляется генетический поиск путем модификации хромосом, входящих в заданный для этого этапа набор модифицируемых типов хромосом MS_i . Это означает, что в полном объеме используется оператор рекомбинации, при котором осуществляется обмен гомологичными хромосомами между парой наборов (решений), а кроссинговер (оператор скрещивания), при котором осуществляется обмен гомологичными генами между парой гомологичных хромосом, и мутация применяются только к тем хромосомам, которые входят в набор типов модифицируемых хромосом.

При параллельном поиске производится распараллеливание процесса генетического поиска. Для каждой параллельной ветви задается набор типов хромосом MS_i , подвергающихся модификации. После некоторого числа генераций осуществляется случайное или направленное перемещение хромосом между любыми популяциями, сформированными на данный момент на параллельных ветвях. После этого на втором шаге модифицированные популяции вновь подвергаются обработке генетическими процедурами на параллельных ветвях.

При параллельно-последовательном подходе на каждой параллельной ветви реализуется последовательная комбинация постановок.

Поскольку изменение параметров устройств РКИС может существенно повлиять на эффективность размещения данных и приложений на серверах и скорость выполнения запросов, целесообразно применить последовательный подход при реализации генетического алгоритма оптимизации, который заключается в последовательной реализации эволюционного процесса в четыре этапа. При этом на каждом этапе модифицируется одна из мультихромосом: устройств Xt , каналов Xc , наборов данных Xd и приложений Xa . Таким образом, в наборы модифицируемых MS_i входит по одному типу мультихромосом:

$$MS_1 \leftarrow \{Xt\}, MS_2 \leftarrow \{Xc\}, MS_3 \leftarrow \{Xd\}, MS_4 \leftarrow \{Xa\}.$$

На первом этапе последовательного генетического алгоритма в качестве исходной служит популяция G_0 , на втором — популяция G_1 , сформированная после отработки первого этапа, и так далее.

Схема реализации последовательного генетического алгоритма для оптимизации РКИС представлена на рис. 8.

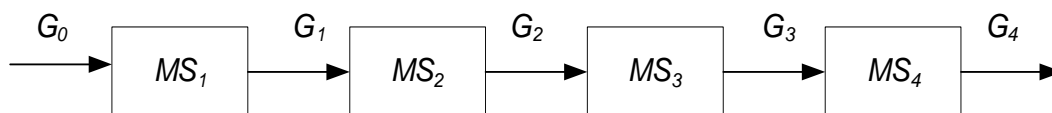


Рисунок 8 — Схема реализации последовательного генетического алгоритма для оптимизации РКИС

Популяция, полученная в результате эволюции на каждом этапе, является входной для следующего поколения (цикла генетического алгоритма). Алгоритм завершает работу после прохожде-

ния всех этапов и из лучших особей выбирается особь с минимальным значением фитнес-функции. Параметры этой особи и представляют собой субоптимальное решение задачи оптимизации.

Выводы

1. Сформулирована задача оптимизации РКИС. В качестве критерия оптимизации предложено использовать максимальное время выполнения пользовательского запроса на выборку данных. Выбраны ограничения при формировании различных вариантов РКИС.

2. Для оптимизации РКИС предложен новый подход, основанный на взаимодействии объектной модели РКИС и генетического алгоритма, в результате работы которого определяются параметры серверов, сетевых устройств, каналов связи РКИС, а также способ размещения наборов данных и приложений на серверах БД и приложений. При этом значение критерия эффективности РКИС будет минимальным с учётом ограничений на размещение данных, приложений и стоимость системы.

3. Разработана динамическая объектная модель РКИС в виде иерархии классов, описывающих поведение основных компонентов РКИС, и управляющего класса, реализующего взаимодействие между объектами РКИС в процессе моделирования.

4. Разработан модифицированный генетический алгоритм для нахождения субоптимального решения задачи оптимизации РКИС, в котором параметры технических средств и размещение наборов данных и приложений системы кодируются при помощи четырёх мультихромосом, а процесс эволюции имитируется адаптированными к специфике задачи генетическими операторами. Предложен последовательный подход к реализации генетического алгоритма.

Литература

1. Ferscha A. Parallel and Distributed Simulation of Discrete Event Systems. *Parallel and Distributed Computing Handbook*, McGraw-Hill. 1996. — PP. 1003–1041.
2. Хаггарти Р. Дискретная математика для программистов. — К.: Техносфера, 2003. — 320 с.
3. Агибалов Г.П., Скутин А.А. Математическая модель и технология разработки безопасных корпоративных информационных систем. Томский государственный университет/ URL: <http://zhurnal.ape.relarn.ru/articles/2001/151.pdf>
4. Гламаздин Е.С., Новиков Д.А., Цветков А.В. «Управление корпоративными программами: информационные системы и математические модели». — М: Компания Спутник+, 2003. — 159 с.
5. Люгер Дж. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем. — 4-е изд. — М.: Вильямс, 2005. — 864 с.
6. Аверилл М.Лоу, Дэвид Кельтон. Имитационное моделирование. — СПб: Питер, 2004. — 846 с.
7. Лаздынь С.В., Землянская С.Ю. Многокритериальная оптимизация корпоративных информационных систем // Наукові праці ДонНТУ. Серія “Обчислювальна техніка та автоматизація” — Випуск 12(118). — Донецьк: ДонНТУ, 2007. — С. 74–82.
8. Лаздынь С.В., Землянская С.Ю. Разработка динамической модели компьютерной информационной системы // Наукові праці ДонНТУ. Серія “Обчислювальна техніка та автоматизація” — Випуск 14(129). — Донецьк: ДонНТУ, 2008. — С. 107–116.
9. *Handbook of Genetic Algorithms*, Edited by Lawrence Davis, Van Nostrand Reinhold, New York, 1991. — 385p.
10. Божич В.И., Лебедев В.Б. Методы генетического поиска для решений, представимых мультихромосомами. // Перспективные информационные технологии и интеллектуальные системы. — Таганрог: Таганрогский государственный радиотехнический университет, 2001. — С.38–44.

Здано в редакцію:
03.03.2009р.

Рекомендовано до друку:
д.т.н, проф. Скобцов Ю.О.