УДК 004.051

В.В. СКЛЯР[1], С.А. МАЛОХАТЬКО[2]

[1]*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина*
[2]*ЗАО «НПП «Радий», Украина*

ОПТИМИЗАЦИЯ ПРОГРАММНОГО КОДА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Сформулированы и решены математические задачи нахождения оптимальных характеристик программного обеспечения, которые применимы при выполнении инженерной задачи оптимизации работы программ. Задачи представляют собой разновидность задачи о ранце и решены методом динамического программирования, что позволяет повысить полноту оценки и улучшить характеристики программного обеспечения и систем в целом. Проанализированы принципы применения инструментальных средств профилирования программ, применяемых для оптимизации программного обеспечения.

Ключевые слова: *оптимизация программы, профилирование программы, динамическое программирование, задача о ранце.*

Введение

Среди практических задач программной инженерии особое место занимает задача оптимизации работы программного обеспечения (ПО) [1, 2]. При этом под оптимизацией понимается модификация ПО для улучшения его эффективности. В данной статье мы будем рассматривать временные характеристики функционирования ПО, которые являются наиболее критичными для систем реального времени [3].

Следует отметить, что термин «оптимизация ПО» (или «оптимизация программного кода») не предполагает строгого решения задачи оптимизации (т.е. нахождения экстремума) в строгой математической постановке [4,5]. Оптимизированное ПО обычно лишь соответствует заданным временным или ресурсным ограничениям, при этом, как правило, невозможно доказать, что полученные характеристики соответствуют экстремуму. Такая ситуация объясняется двумя следующими причинами. Во-первых, функционирование ПО зависит от входных данных, т.е. оптимальное функционирование при одних входных данных не означает оптимального функционирования при всех возможных входных данных. Во-вторых, в большинстве случаев невозможно задать веса для компонентов программного кода, поскольку все они должны быть выполнены. В современной литературе по программной инженерии отсутствует какая-либо математическая формализация постановки задачи оптимизации ПО. Следует также отметить, что публикации, посвященные теоретическим аспектам оптимизации ПО, достаточно редки, и как правило, сводятся к приемам работы с инструментальными средствами оптимизации.

Целью данной статьи является выявление случаев, когда задачи оптимизации ПО могут быть сформулированы как задачи нахождения экстремума, и решение таких задач в общем виде.

1. Формализация задачи оптимизации ПО

Задача оптимизации ПО может быть представлена как набор итераций, включающих изменение структуры кода и проверку того, соответствуют ли после модификации временные параметры ПО заданным (рис. 1).

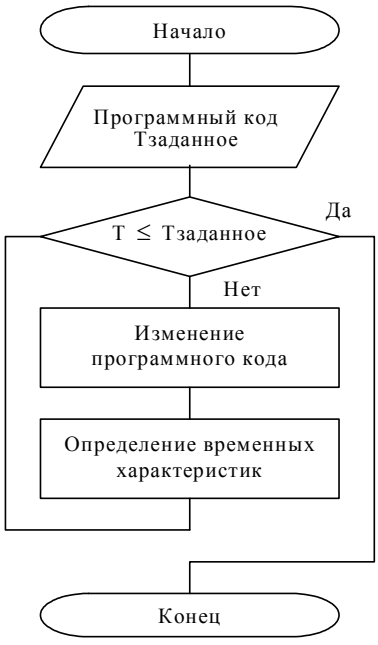

Рис. 1. Общий вид алгоритма для процедуры оптимизации ПО

2. Решение задачи оптимизации при ограничениях по времени

Рассмотрим случаи, когда оптимизация ПО может быть сформулирована и решена как оптимизационная задача. Такая задача может быть сформулирована, например, при разработке прототипа ПО, когда некоторые выполняемые функции могут быть удалены из технического задания в том случае, когда обладая низким приоритетом, функция требует значительных временных ресурсов, которые могут нарушить требуемое время выполнения $T_{заданное}$.

Тогда, если имеется множество функций, реализуемых ПО, для каждой из которых задан вес (приоритет) w_i и время выполнения t_i, в общем виде оптимизационную задачу можно сформулировать следующим образом. Необходимо выбрать функции ПО так, чтобы получить максимальный суммарный вес этих функций при одновременном соблюдении ограничения по времени. В такой постановке оптимизационная задача представляет собой классическую задачу о ранце (с возможностью единичного выбора) [4,5].

Численные значения весов можно получить исходя из приоритетов функций ПО. В простейшем случае задача «взвешивания» функций может быть решена следующим образом. Пусть имеется i = 1..M функций ПО, для каждой из которых необходимо определить приоритет r_i, задаваемый число от 1 до N (N – наивысший приоритет). Сумма всех приоритетов функций составляет $R = \sum_{i=1}^{M} r_i$. Тогда базовая единица веса составляет $w_r = 1 / R$. Вес каждой из функций определяется как $w_i = r_i \cdot w_r = r_i / \sum_{i=1}^{M} r_i$. Поскольку суммарный вес функций ПО W будет соответствовать суммарному приоритету при ограничениях на ресурсы (время), W можно назвать показателем эффективности.

В формальном виде задача оптимизации ПО может быть сформулирована в следующем виде: найти функции ПО, для которых $W \to \max$ при $T \le T_{заданное}$. Задача нахождения максимальной эффективности при заданном времени выполнения является задачей динамического программирования.

Целевой функцией является

$$f(W)=\sum_{i=1}^{A} w_i \cdot \frac{1}{t_i} \to \max$$

при ограничениях $\sum_{i=1}^{A} t_i \le T_{ЗАДАННОЕ}$.

Решение такой задачи включает следующую последовательность действий (рис. 2).

1. Присвоение времени выполнения ПО значения T = 0 и счетчику итераций значения s = 0.

2. Формирование кортежа $W^{(A-s)} / T^{(A-s)} = \left\langle \frac{w_i}{t_i} \right\rangle$.

3. Выбор максимального элемента кортежа $w_i / t_i = \max$.

4. Проверка условия существования нескольких элементов кортежа с одинаковым максимальным значением.

5. Если условие, указанное в действии 4, выполняется, то следует выбрать ту функцию, для которой $t_i = t_{i\,min}$.

6. Присвоение суммарному времени выполнения ПО значения $T=T+t\left(w_i/t_i=\max\right)$ и счетчику значения s = s + 1.

7 Проверка условия $T \ge T_{заданное}$. Если данное условие выполняется, то выполнение алгоритма заканчивается.

8. Если условие, указанное в действии 6 не выполняется, то выбранная посредством шагов 3-5 функция f_i включается в состав множества функций ПО F, а из кортежа $W^{(A-s)}/T^{(A-s)}$ удаляется элемент, соответствующий функции.

После этого снова выполняются шаги 2–8.

Решением задачи является множество функций ПО $F(W \to \max) = \{f_i\}$, для которого имеем значение показателя эффективности $W=\sum_i w\left(\frac{w_i}{t_i}=\max\right)$ и значение времени выполнения $T=\sum_i t\left(\frac{w_i}{t_i}=\max\right)$.

Аналогичная задача может решаться в процессе эксплуатации ПО, когда на протяжении одного вычислительного такта необходимо решить ограниченное число задач с различными приоритетами. Постановка и решение такой задачи совпадают с рассмотренной в данном разделе оптимизационной задачей о включении функций прототипа в релиз. При решении такой задачи веса могут задаваться динамически, т.е. меняться от такта к такту.

3. Решение задачи оптимизации при ограничениях по эффективности

Оптимизационная задача может быть сформулирована в другой постановке (хотя такой подход не является типичным): найти функции ПО, для которых $T \to \min$ при $W \ge W_{заданное}$.. Целевой функцией в данном случае является $f(T)=\sum_{i=1}^{A} t_i \cdot \frac{1}{w_i} \to \min$ при ограничениях $\sum_{i=1}^{A} w_i \ge W_{ЗАДАННОЕ}$.

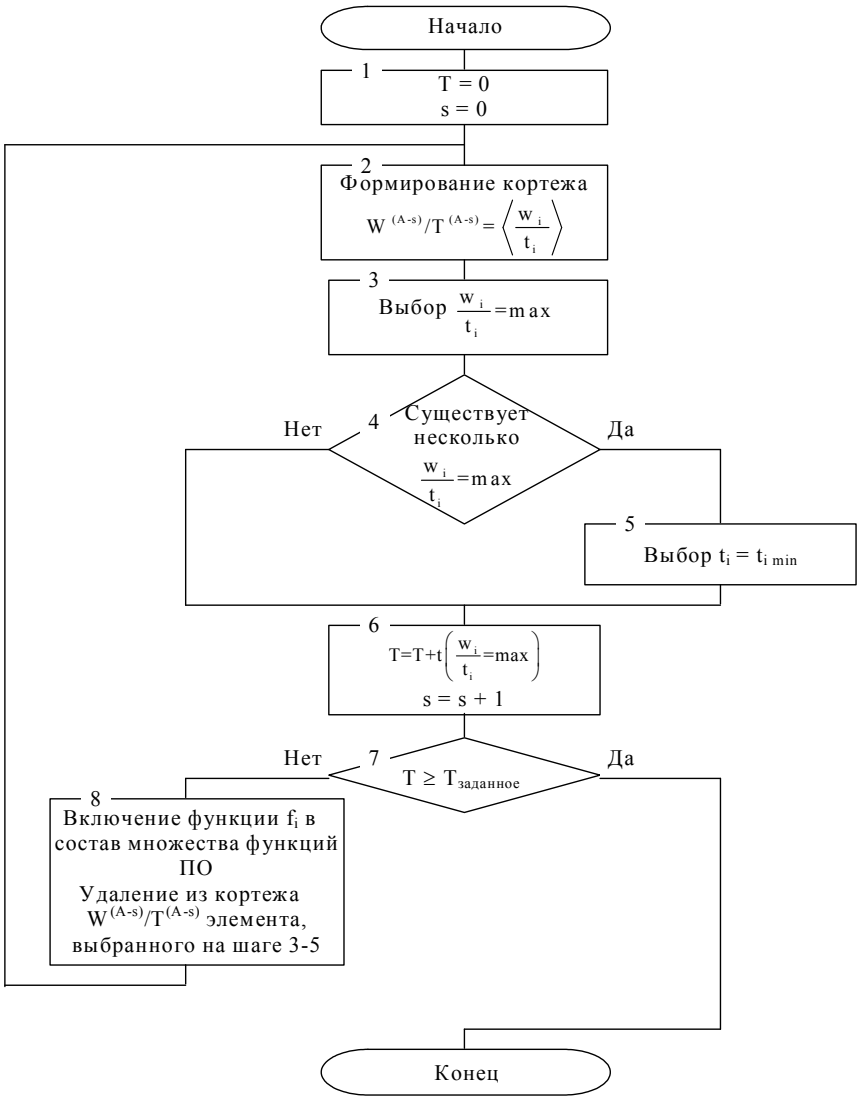

Рис. 2. Алгоритм решения оптимизационной задачи нахождения максимальной эффективности ПО при заданном времени выполнения

Решение такой задачи включает следующую последовательность действий (рис. 3).

1. Присвоение суммарной эффективности (весу) ПО значения W = 0 и счетчику итераций значения s = 0.

2. Формирование кортежа $T^{(A-s)} / W^{(A-s)} = \langle t_i / w_i \rangle$.

3. Выбор максимального элемента кортежа $t_i / w_i = \min$.

4. Проверка условия существования нескольких элементов кортежа с одинаковым максимальным значением.

5. Если условие, указанное в действии 4, выполняется, то следует выбрать ту функцию, для которой $w_i = w_{i\,max}$.

6. Присвоение суммарной эффективности выполнения ПО значения $W=W+w\left(t_i / w_i = \max\right)$ и счетчику значения s = s + 1.

7 Проверка условия $W \le W_{заданное}$. Если данное условие выполняется, то выполнение алгоритма заканчивается.

8. Если условие, указанное в действии 6 не выполняется, то выбранная посредством шагов 3-5 функция f_i включается в состав множества функций ПО F, а из кортежа $T^{(A-s)}/W^{(A-s)}$ удаляется элемент, соответствующий функции.

После этого снова выполняются шаги 2–8.

Решением задачи является множество функций ПО $F(T \to \min) = \{f_i\}$, для которого имеем значение показателя эффективности

$$W=\sum_i w\left(t_i / w_i = \min\right)$$

и значение времени выполнения

$$T=\sum_i t\left(t_i / w_i = \min\right).$$

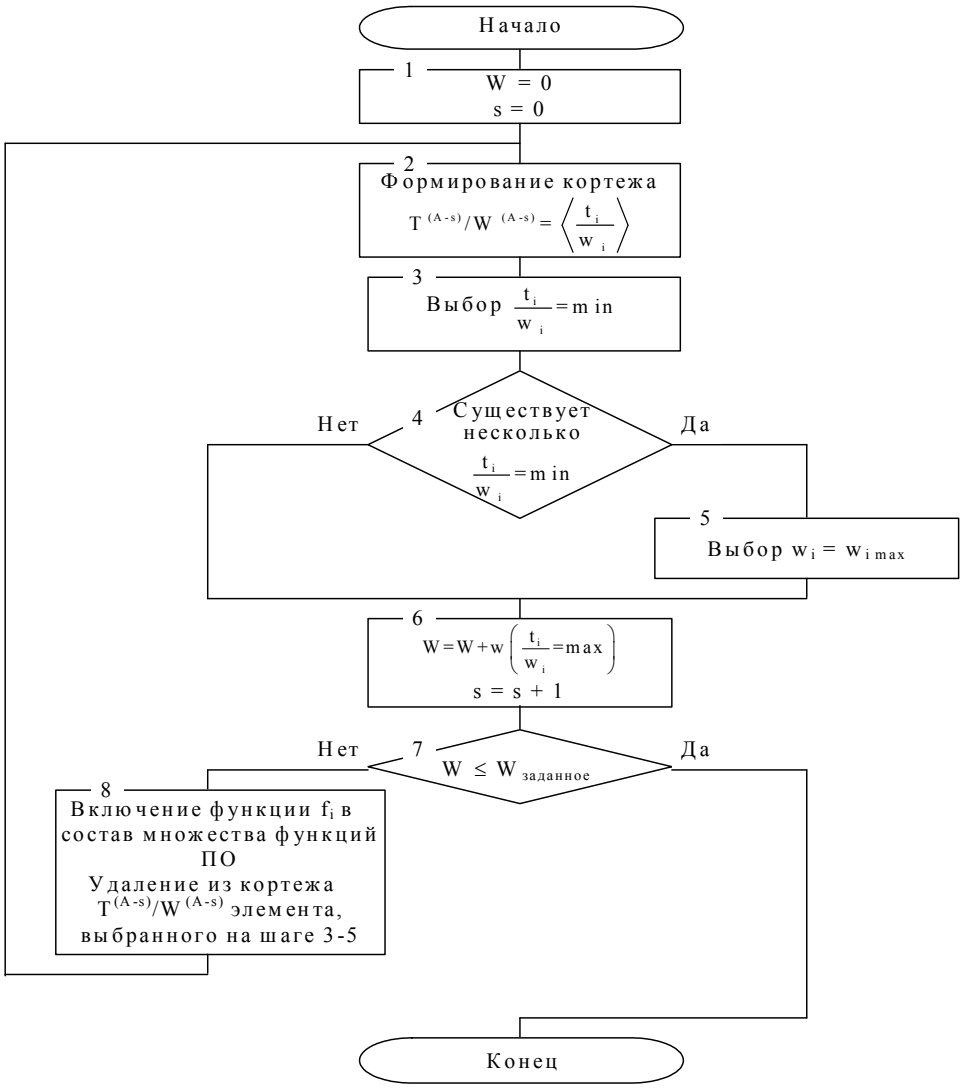

Рис. 3. Алгоритм решения оптимизационной задачи нахождения максимального времени выполнения ПО при заданной эффективности

4. Инструментальные средства для выполнения профилирования ПО

Основным методом, применяемым для оценки результатов оптимизации ПО, является профайлинг.

В программной инженерии под профайлингом (профилированием) ПО подразумевается сбор характеристик работы программы, таких как время и количество выполнений отдельных фрагментов.

Инструментальные средства, используемые для выполнения профайлинга, называют профайлерами (профилировщиками).

Примерами таких инструментальных средств являются AMD CodeAnalyst, Profiling Tools (встроен в Microsoft Visual Studio), DevPartner Studio и др.

Наиболее продвинутым с точки зрения функциональности является DevPartner Studio, который позволяет реализовать следующие основные функции:

– анализ функционирования (Performance Analysis) – непосредственный анализ производительности кода, при котором определяется время выполнения и количество вызовов программных компонентов различного уровня (строки, функции, модули); данная функция работает по методу инструментирования, т.е. добавляет свои инструкции для каждой строки исходного кода, на основании чего определяется процессорное время выполнения строк кода;

– статический анализ (Static Analysis), заключающийся в поиске нежелательных конструкций программного кода для управляемого кода;

– анализ покрытия (Code Coverage Analysis) — процесс выявления неиспользуемых участков кода;

– обнаружение ошибок (Error Detection) – обнаруживает утечки памяти, выходы за пределы массива, невозвращенные ресурсы и т.д.

С учетом разработанной методики, указанные инструментальные средства могут быть доработаны

таким образом, чтобы обеспечивать математически строгую оптимизацию времени выполнения ПО.

Заключение

Сформулированные и решенные в статье оптимизационные задачи в совокупности образуют метод оценки эффективности программного обеспечения путем оптимизации программного кода по критерию «приоритет выполнения/время выполнения».

Это позволяет повысить полноту оценки и улучшить характеристики программного обеспечения и систем в целом.

Важной прикладной задачей является оценка и выбор инструментальных средств для оптимизации программного кода методом профилирования, т.е. путем сбора информации о временных характеристиках ПО.

Литература

1. Касперски К. Техника оптимизации программ. Эффективное использование памяти / К. Касперсики. – СПб: БХВ-Петербург, 2003. – 464 с.

2. Магда Ю.С. Использование ассемблера для оптимизации программ на С++ / Ю.С. Магда. – СПб: БХВ-Петербург, 2004. – 492 с.

3. Скляр В.В. Оценка качества программного обеспечения верхнего уровня информационно-управляющих систем АЭС / В.В. Скляр, Ю.А. Белый, С.А. Малохатько // Радіоелектронні і комп'ютерні системи. – 2007. – №. 6(25). – С. 153-158.

4. Левитин А.В. Алгоритмы: введение в разработку и анализ / А.В. Левитин. – М.: Вильямс, 2006. – 576 с.

5. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Вильямс, 2006. – 1296 с.

Поступила в редакцию 20.01.2009

Рецензент: д-р техн. наук, проф., декан механико-математического факультета Г.Н. Жолткевич, Харьковский национальный университет им. В.Н. Каразина, Харьков.

ОПТИМІЗАЦІЯ ПРОГРАМНОГО КОДУ З ВИКОРИСТАННЯМ МЕТОДІВ МАТЕМАТИЧНОГО ПРОГРАММУВАННЯ

В.В. Скляр, С.А. Малохатько

Сформульовано та вирішено математичні задачі знаходження оптимальних характеристик програмного забезпечення, які застосовуються при виконанні інженерних задач оптимізації роботи програм. Задачі являють собою різновид задачі про ранець і вирішені методом динамічного програмування. Проаналізовано принципи застосування інструментальних засобів профілювання програм, які використовуються для оптимізації програмного забезпечення.

Ключові слова: оптимізація програми, профілювання програми, динамічне програмування, задача про ранець.

PROGRAM CODE OPTIMIZATION WITH USING MATHEMATICAL PROGRAMMING METHODS

V.V. Sklyar, S.A. Malohatko

Mathematical problems of software optimal characteristics searching which are used for software engineering works during software performance optimization are formulated and solved. These problems are kinds of a knapsack problem, and the problems are resolved by a dynamic programming method. Using principles of tools for software profiling which are used for software optimization are analysed.

Key words: software optimization, software profiling, dynamic programming, knapsack problem.

Скляр Владимир Владимирович – канд. техн. наук, доцент, доцент кафедры технологии компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: vvsklyar@mail.ru.

Малохатько Сергей Анатольевич – начальник отдела разработки программного обеспечения ЗАО «НПП «Радий», Кировоград, Украина, e-mail: malokhatko@yahoo.com.