

УДК 623.618 623.4.11

МЕТОД ОЦЕНКИ ТРУДОЁМКОСТИ ПРОЦЕССОВ ПРОГРАММИРОВАНИЯ

Старусев Андрей Викторович, аспирант, филиал Астраханского государственного университета в г. Знаменске, 416540, Российская Федерация, Астраханская область, г. Знаменск, ул. Островского, 16, e-mail: starusev-av@yandex.ru

Опыт построения моделирующих алгоритмов для агрегатных систем ещё недостаточен. Сформулированы лишь основные принципы построения моделирующих агрегатов для систем, которые нашли применение и при построении математических моделей моделирующих установок. Выбор принципов построения математических моделей зависит от решаемых задач, ожидаемых результатов моделирования, имеющихся вычислительных средств и времени для решения задачи. При подготовке задач к решению на электронной вычислительной машине используются различные средства программирования. Выбор их определяется, в частности, наличием трансляторов или других средств математического обеспечения, а также длительностью и стоимостью разработки, отладки и эксплуатации программ. Все это предопределяет необходимость тщательного изучения структуры трудовых затрат при программировании задач. Тем не менее, на практике отсутствуют расчетно-обоснованные показатели для оценки трудоёмкости разработки программ. Предлагаемый метод позволяет оценить сложность разрабатываемых программ, что очень важно при планировании работ по программированию. Реализация предложенного метода на практике позволяет более качественно спланировать работы по программированию решаемых задач.

Ключевые слова: сложность программ, число элементарных операций, время реализации программ, программирование, трудовые затраты при программировании задач, рациональное использование ресурсов

THE METHOD FOR VALUATING PROGRAMMING-BASED LABOUR INPUT PROCESSES

Starusev Andrey V., post-graduate student, Branch of Astrakhan State University in Znamensk, 16 Ostrovsky St., Znamensk, Astrakhan region, 416540, Russian Federation, e-mail: starusev-av@yandex.ru

The article relates that experience in creating modeling algorithms is still insufficient to enable their wider application in modular systems. It adds that thus far only the main principles for creating modeling aggregates have been formulated for systems which could be applied to create mathematical models of modeling setting. In the critique's view, the choice of principles for creating mathematical models depends on the resolved tasks expected for the modeling results, the available computing methods and the time allocated for task decision. It would appear that electronic computers use various programming aids to prepare their decision-oriented tasks. Their choice is defined, in particular, by the presence of translators or other means of software, and also by the program's duration, and the cost of developing, debugging and maintaining it. Prior to undertaking programming tasks, the structure of labor expenses needs to be carefully studied and understood. One means of accomplishing this task would be to use rated-justified indexes that estimate the development of labor input for programs in practice. The suggested method would enable the researcher to estimate the complexity of the developed programs' scheduling and programming tasks. However, practical implementation of the proposed method would actually enable the researcher to undertake better programming efforts aimed at resolving the computing-related tasks.

Keywords: complexity of programs, number of elementary operations, time of implementation of programs, programming, labor expenses for programming tasks, rational usage of resources

Процесс подготовки задач к машинному решению обычно подразделяется на следующие этапы [1, 2]:

- анализ проблемы и содержательная постановка задачи;
- алгоритмизация задачи и составление блок-схем программ;
- программирование (кодирование) задачи;
- отладка (автономная и комплексная) программы;
- оформление рабочей (эксплуатационной) документации.

Результаты этапов постановки и алгоритмизации задачи объединяют в документ, называемый «Задание на программирование задачи». В нем содержится детальное описание последовательности решения конкретной задачи и описание данных, необходимых при составлении программы.

Сложность программ определяется конкретными условиями постановки и решения задачи на электронной вычислительной машине (ЭВМ), персональной электронной вычислительной машине (ПЭВМ), но всегда связана с пространственно-временными характеристиками алгоритма, в качестве которых выступают объем памяти, требуемой для хранения перерабатываемой информации, время реализации алгоритма и т.д. Точно так же мера сложности программ нужна для оценки объективных трудностей, присущих различным этапам процесса подготовки задач к машинному решению, в первую очередь для оценки трудоёмкости процессов программирования.

Очевидно, что затраты труда на разработку программы определяются сложностью последней [3]. Под сложностью программы понимается её длина. В общем случае длиной программы называется числовая характеристика, которая определяет длину записи программы в принятом наборе операторов. В качестве такой характеристики принимаем число машинных команд программы и соответственно объём памяти, занимаемой программой.

Кроме длины сложность программы определяем связностью, которую для решаемых задач оцениваем максимальным числом исходных, промежуточных и результатных показателей, удерживаемых в памяти при выполнении программы.

Следующей характеристикой сложности считаем число элементарных операций, необходимых для выполнения программы (объём вычислений).

Время реализации $T(P)$ программы на ЭВМ (ПЭВМ) определяем через объём вычислений:

$$T(P) = a_0 + \sum_{i=1}^K N_i t_i, \quad (1)$$

где N_i – количество операций i -го типа; t_i – время выполнения i -й операции; K – количество типов операций.

Точная оценка количества операций обычно затруднительна, поэтому всё множество операций разделяем на «короткие» N_K и «длинные» $N_{дл}$ операции. Тогда время реализации программы составит:

$$T^*(P) = N_K t_K + N_{дл} t_{дл}, \quad (2)$$

где $t_K, t_{дл}$ – время реализации соответственно «короткой» и «длинной» операции.

Время реализации программы зачастую превышает расчетное, так как выполняются операции обращения к внешней памяти и увеличивается объём вычислений из-за ухудшения качества программы при использовании средств автоматизации программирования, а также по другим причинам. Поэтому время выполнения программы в общем случае составит:

$$T(P) = K_T K_O T^*(P), \quad (3)$$

где K_T – коэффициент трансляции; K_O – коэффициент обращения к внешней памяти ($K_O = 1$, если длина программы укладывается в оперативной памяти, $K_O > 1$, в противном случае).

Время реализации программы можно оценить в так называемых приведённых (эталонных) операциях [4]:

$$T(P) = N_{\text{Э}} t_{\text{Э}}, \quad (4)$$

где $N_{\text{Э}}$ – число приведённых (эталонных) операций; $t_{\text{Э}}$ – длительность выполнения эталонной операции.

За эталонную принимаем короткую операцию, например логическую или сложения.

Для того чтобы определить общее число эталонных операций, нужно знать α_i – отношения длительности i -й операции (t_i) к длительности короткой операции (τ), принятой за эталонную:

$$\alpha_i = \frac{t_i}{\tau}. \quad (5)$$

В расчетах в качестве эталонной удобно оперировать операцией, в которой оценивается паспортное быстродействие ЭВМ (ПЭВМ).

Если известно α_i , то

$$N_{\text{Э}} = \sum_{i=1}^K \alpha_i N_i. \quad (6)$$

Если известны среднее число операций для реализации данной программы и типичный состав операций для решаемого класса задач в относительных величинах β_i или в процентах, то общее время выполнения программы равно:

$$T(P) = N_{\text{ср}} t_{\text{ср}} = N_{\text{ср}} \sum_{i=1}^K \beta_i t_i; \quad \sum_{i=1}^K \beta_i = 1. \quad (7)$$

Тогда среднее время выполнения одной операции составит:

$$t_{\text{ср}} = \sum_{i=1}^K \beta_i t_i, \quad (8)$$

а среднее быстродействие ЭВМ (ПЭВМ) для решаемого класса задач будет равно:

$$V(P) = \frac{1}{t_{\text{ср}}} = \frac{1}{\sum_{i=1}^K \beta_i t_i}. \quad (9)$$

Совокупность рассмотренных показателей достаточно точно характеризует сложность программ. Реализация рассмотренного метода на практике позволяет изучить структуру трудовых затрат при программировании задач, что весьма важно для рационального использования ресурсов.

Вывод. Предложенный метод позволяет изучить структуру трудовых затрат при программировании задач. Реализация предложенного метода на практике позволяет более качественно спланировать работы по программированию решаемых задач.

Список литературы

1. Гутер Р. С. Программирование и вычислительная математика / Р. С. Гутер, П. Т. Резниковский, С. М. Резник. – Москва : Наука, 1971. – 696 с.
2. Джермейн К. Программирование на IBM/360 / К. Джермейн. – Москва : Мир, 1973. – 870 с.
3. Карманов В. Г. Математическое программирование / В. Г. Карманов. – Москва : ФИЗМАТЛИТ, 2011. – 264 с.
4. Моисеев Н. Н. Математические задачи системного анализа / Н. Н. Моисеев. – Москва : Наука, 1981. – 487 с.

References

1. Guter R. S., Reznikovsiy P. T., Reznik S. M. *Programmirovaniye i vychislitel'naya matematika* [Programming and calculus mathematics]. Moscow, Nauka, 1971. 696 p.
2. Dzhermeyn K. *Programmirovaniye na IBM/360* [Programming by IBM/360]. Moscow, Mir, 1973. 870 p.
3. Karmanov V. G. *Matematicheskoye programmirovaniye* [Mathematical programming]. Moscow, 2011. 264 p.
4. Moiseev N. N. *Matematicheskiye zadachi sistemnogo analiza* [Mathematical problems of system analysis]. Moscow, Nauka, 1981. 487 p.

УДК 519.863

**РАЗРАБОТКА МОДЕЛИ ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ
ОБЪЕКТОВ СЕТИ РОЗНИЧНОЙ ТОРГОВЛИ
НА ОСНОВЕ ЗАКОНА ОБРАТНОГО КВАДРАТА**

Шиккульская Ольга Михайловна, доктор технических наук, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а, e-mail: shikul@mail.ru

Набережная Алена Владимировна, аспирант, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а, e-mail: naberezhnaya_av@mail.ru

Создание новых и развитие имеющихся сетей торговых объектов – актуальная проблема современного рынка. Но необходимо учитывать, что создание розничной торговой сети требует решения ряда важных вопросов, среди которых первоочередным является оптимальное месторасположение новых торговых точек, поскольку убытки от нерационального расположения могут оказаться колоссальными.

На сегодняшний день существует большое число разнообразных методов и моделей размещения торговых объектов, но ни один из указанных методов не является универсальным, отвечающим всем требованиям и запросам современного рынка.

Наиболее эффективные решения проблем достигаются посредством переноса законов, открытых для одной области знаний, в другую. Одним из таких универсальных законов является закон обратного квадрата, который описывает взаимодействие гравитационных полей (закон Ньютона), взаимодействие заряженных частиц (закон Кулона), закон Био-Савара-Лапласа – магнитный аналог закона Кулона.

Авторами была рассмотрена возможность применения закона обратного квадрата для исследуемой предметной области. При этом наиболее подходящим оказался закон Кулона, учитывающий направление сил взаимодействия в зависимости от знаков зарядов взаимодействующих частиц. Также авторами были проанализированы все факторы, влияющие на размер получаемой прибыли / убытков от функционирования объектов торговой сети.