

Лабораторная работа №4

Эволюционные алгоритмы оценки стоимости проектов в программной инженерии

Цель работы: разработка эволюционного алгоритма оценки стоимости программных проектов. Графическое отображение результатов.

При выполнении лабораторной работы можно использовать следующие источники из прилагаемого списка литературы [1,2]

8.1 Методика COCOMO

Одной из самых популярных моделей, используемых для оценки сложности проектируемого программного обеспечения (ПО), является модель COCOMO (COConstructive COst Model), предложенная Boehm [1,2]. Эта модель разработана на основе фактически статистики 63 проектов ПО (НАСА). Модель позволяет определить математическую зависимость между сложностью ПО, выраженную в килостроках кода, и затратами на его разработку, которые оцениваются в человеко-месяцах. Ядром модели является следующая формула $Ef = aL^b$, где L - длина кода ПО в килостроках; Ef - оценка сложности проекта в человеко-месяцах; a и b - коэффициенты (параметры) модели, которые для различных типов ПО имеют различные значения. Основная проблема модели COCOMO заключается в том, что она не обеспечивает реальных оценок на затраты при проектировании ПО в современных условиях. Т.е. оценка программного обеспечения на основе существующих параметров не всегда дает точный результат; из-за этого часто требуется настройка параметров для получения более точных результатов.

Поэтому в настоящее время идет активный поиск новых моделей (или развития и модификаций существующих). Это ограничение модели СОСОМО можно преодолеть путем применения методов искусственного интеллекта, таких как искусственные нейронные сети, генетические алгоритмы и другие метаэвристики.

В данной лабораторной работе для определения значений коэффициентов a и b используются генетический или роевой алгоритм в соответствии с заданным вариантом. Фактически задача сводится к машинному обучению на заданной обучающей выборке. В этом случае обучающая выборка строится на основе следующей таблицы, которая дает реальные данные для 18 проектов НАСА, на основе которых мы ищем зависимость между L и E_f .

Таблица 8.1. Экспериментальные данные проектов НАСА

Номер проекта	L	Me	E_f	E_{fm}	E_{fm2}
1	90,2000	30,0000	115,8000	124,8585	134,0202
2	46,2000	20,0000	96,0000	74,8467	84,1616
3	46,5000	19,0000	79,0000	75,4852	85,0112
4	54,5000	20,0000	909,8000	85,4349	94,9828
5	31,1000	35,0000	39,6000	50,5815	56,6580
6	67,5000	29,000	98,4000	99,0504	107,2609
7	12,8000	26,000	18,9000	24,1480	32,6461
8	10,5000	34,0000	10,3000	18,0105	25,0755
9	21,5000	31,0000	28,5000	37,2724	44,3086

10	3,1000	26,000	7,0000	4,5849	14,4563
11	4,2000	19,0000	9,0000	8,9384	19,9759
12	7,8000	31,0000	7,3000	13,5926	21,5763
13	2,1000	28,0000	5,0000	1,5100	11,2703
14	5,0000	29,0000	8,4000	8,2544	17,0887
15	78,6000	35,0000	98,7000	110,5249	118,0378
16	9,7000	27,0000	15,6000	18,2559	26,8312
17	12,5000	27,0000	23,9000	23,3690	31,6864
18	100,8000	34,0000	138,3000	135,4825	144,4587

При этом данные по 13 проектам (из 18) можно использовать в качестве обучающего множества, а данные остальных 5 проектов – в качестве тестового множества (для проверки - тестирования) полученных значений коэффициентов a и b . Иногда для обучения используется расширенное множество проектов НАСА, представленное в табл.8.2.

Таблица 8.2 Расширенное множество проектов НАСА

№	L — в килостроках	Ef — реальная стоимость в человеко-месяцах	COCOMO с ошибкой MRE	ИНС с MRE	ГА с MRE	Гибрид с MRE
1	2.2	8.4	24.15	13.65	8.95	6.32
2	3.5	10.8	3.95	5.26	4.69	1.13
3	5.5	18	7.36	5.21	6.75	4.35
4	6	24	58.88	34.10	27.63	28.02
5	9.7	25.2	20.05	11.50	13.49	7.61
6	7.7	31.2	23.91	12.35	7.54	12.42
7	11/3	36	30.83	17.45	12.45	13.35
8	8.2	36	29.55	16.68	14.23	11.21
9	6.5	42	28.32	18.52	11.64	13.42

10	8	42	22.22	13.21	15.47	9.34
11	20	48	27.21	14.65	16.32	12.16
12	10	48	41.66	23.98	19.84	19.84
13	15	48	46.19	28.04	23.11	26.74
14	10.4	50	34.90	25.47	17.02	21.95
15	13	60	9.36	6.53	5.31	7.15
16	14	60	25.88	15.41	17.54	8.46
17	19.7	60	6,10	7.21	4.21	2.54
18	32.5	60	93.91	47.35	56.47	36.10
19	31.5	60	3.81	6.52	5.46	1.07
20	12.5	62	27.96	13.11	10.84	4.31
21	15.4	70	22.51	10.13	12.76	7.02
22	20	72	60.76	45.68	33.82	27.11
23	7.5	72	41.75	32.61	24.15	15.04
24	16.3	82	29.79	23.40	17.37	7.46
25	15	90	39.54	27.68	21.51	19.01
26	11.4	98.8	42.04	25.10	19.07	21.74
27	21	107	36.75	24.55	16.53	9.02
28	16	114	34.48	24.55	16.53	9.92
29	25.9	117.6	27.85	19.36	11.57	17.09
30	24.6	117.6	31.65	21.87	16.34	14.82
31	29.5	120	18.94	11.15	7.13	6.44
32	19.3	155	35.78	17.30	21.06	16.72
33	32.6	170	29.88	19.54	15.19	5.68
34	35.5	192	32.10	16.35	8.37	13.06
35	38	210	28.46	13.19	19.50	15.43
36	48.5	239	24.31	8.43	12.07	7.94
37	47.5	252	37.81	21.36	18.64	11.83
38	70	278	21.28	9.42	11.46	6.24

39	66.6	300	23.76	11.30	16.79	9.22
40	66.6	352.8	35.17	19.25	11.20	13.62
41	50	370	36.90	23.54	13.48	7.42
42	79	400	45.74	31.29	22.97	18.06
43	90	450	38.29	20.11	31.73	15.94
44	78	571.4	24.50	13.64	8.03	5.21
45	100	215	120.66	86.14	61.42	51.04
46	150	324	49.50	26.80	13.09	23.83
47	100	360	44.97	17.67	25.07	12.62
48	100	360	15.85	6.23	8.62	9.84
49	190	420	1.89	4.87	3.84	2.65
50	115.8	480	11.37	16.49	5.32	5.42
51	101	750	19.87	10.67	6.46	12.71
52	161.1	815	4.76	10.25	8.41	5.95
53	284.7	973	38.36	21.43	17.09	10.14
54	227	1181	3.93	2.36	6.31	4.62
55	177.9	1228	3.64	9.84	5.08	2.06
56	282.1	1368	17.21	9.46	11.36	7.92
57	219	2120	29.00	21.03	15.81	8.31
58	423	2300	25.78	16.07	7.44	9.02
59	302	2400	0.46	3.24	5.64	2.54
50	370	3240	25.21	8.62	3.21	6.87

Напомним, что для того, чтобы применить генетический алгоритм для решения некоторой проблемы необходимо, прежде всего, определить:

1. Кодирование (представление потенциального решения);
2. Для определенного кодирования выбрать или разработать генетические операторы кроссовера, мутации и репродукции.

3. Фитнесс-функцию из условия задачи.

4. Определить параметры ГА: число особей в популяции, значения вероятностей кроссовера P_c и P_m .

В данном случае потенциальное решение представляется вектором значений параметров (a,b) . Значения каждого параметра лежат в некотором диапазоне $a \in [L_a, U_a]$, $b \in [L_b, U_b]$.

Для кодирования значений вектора (a,b) можно использовать как двоичное кодирование, так и непосредственное представление потенциального решения в виде вектора вещественных чисел (a,b) . Кодирование решения определяется вариантом курсовой работы согласно приведенной далее таблице. В случае двоичного кодирования можно использовать стандартный 1-точечный, 2-точечный (или однородный) кроссовер и стандартный оператор мутации. В случае вещественного кодирования следует использовать какой-либо вещественный кроссовер (например, в виде линейной комбинации родительских векторов) и вещественную мутацию. Значения параметров ГА следует подобрать экспериментально в ходе эксперимента.

Фитнесс-функция

В качестве фитнес-функции в данном случае следует взять различие между реальными значениями стоимостей проектов и модельными значениями (оценками) стоимостей этих же проектов, которые вычислены согласно приведенной формуле с найденными с помощью ГА коэффициентами a и b . Это различие (расстояние между оценками) можно оценить по-разному - в различной метрике. Можно взять, например, метрику

абсолютных значений (Манхэттен – метрика городских кварталов), где это различие определяется с помощью следующей формулы

$$MD = \sum_{i=1}^n |Ef_i - Efm_i|.$$

Здесь Ef_i – реальная (измеренная) стоимость i -го проекта в человеко-месяцах и Efm_i – модельная оценка того же проекта, вычисленная с помощью приведенной формулы с найденными путем применения ГА коэффициентами a и b . Для оценки различия можно использовать и другие метрики [18], например:

- среднее значение относительной погрешности (MMRE)

$$MMRE = \frac{1}{n} \sum_{i=1}^n \frac{|Ef_i - Efm_i|}{Ef_i},$$

- корень квадратный среднеквадратичной ошибки (RMS)

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n |Ef_i - Efm_i|},$$

- отклонение (дисперсия) (VAF) $VAF = \left[1 - \frac{\text{var}(Ef - Efm)}{\text{var}(Ef)} \right] \times 100\%,$

- Евклидово расстояние $ED = \sqrt{\frac{1}{n} \sum_{i=1}^n (Ef_i - Efm_i)^2}.$

8.2 Практическая часть

Тестовые примеры

Для данного вида задачи существует большое число тестовых примеров – Benchmark-ов. Например, можно использовать сокращенное (табл.?) или расширенное множество проектов НАСА(табл

Порядок выполнения лабораторной работы

1. Разобраться в теоретическом описании математического метода оценки стоимости программного проекта – модели СОСОМО.
2. Из приведенной выше табл.8.1 (или табл.8.2) экспериментальных данных (программных проектов НАСА) отобрать из 18 проектов в качестве обучающего множества 13 (40) проектов.
3. В соответствии с вариантом лабораторной работы, заданного табл.8.3 определить тип используемого эволюционного алгоритма (генетический или роевой алгоритм, генетическое программирование), кодирование потенциального решения, вид ошибки в целевой функции, вид генетических операторов кроссовера, мутации и репродукции

Таблица 8.3 Выбор варианта лабораторной работы

№ варианта	Тип эволюционного алгоритма	Кодирование решения	Фитнесс - функция	Оператор кроссовера	Оператор мутации	Оператор репродукции	
1	ГА	Двоичн.стр .	MD	1-точечный	Классическая	рулетка	
2	ГА	Веществ. вектор	MMRE	арифметич	Арифметич	ранговый	
3	ГА	Двоичн.стр .	RMS	2-точечный	Классическая	Турнир	

4	ГА	Веществ. вектор	ED	арифметич	арифмети ч	рулетка	
5	ГП	Дерево	MD	поддеревьев	Усекающ. Растущ.	ранговый	
6	ГА	Дерево	MMRE	Поддеревьев	Усекающ. Растущ.	Турнир	
7	ГП	Дерево	RMS	поддеревьев	Усекающ. Растущ.	ранговый	
8	РА	Веществ. вектор	ED	-	-	-	
9	РА	Веществ. вектор	MD	-	-	-	
10	РА	Веществ. вектор	MMRE	-	-	-	

4. Отработать алгоритм решения задачи с помощью заданного метода на обучающем множестве.

5. Разработать программу на любом доступном вам языке программирования, включающую в себя реализацию пользовательского интерфейса в виде диалогового меню, реализацию алгоритма решения поставленной задачи заданным методом.

6. Протестировать разработанную программу: вычислить заданный тип ошибки на тестовом множестве – оставшихся 5 (из 18) проектов табл.8.1 (или табл.8.2).

5. Выполнить вывод полученного решения в виде текста и графиков.

В отчете должны быть представлены следующие пункты:

1. Титульный лист
2. Постановка задачи.
3. Теоретические сведения.
4. Входные данные - таблица.

5. Разработанный генетический алгоритм в виде псевдокода или блок-схемы.
6. Используемое кодирование потенциального решения в ГА (двоичное или вещественное).
7. Используемые генетические операторы: отбора родителей, кроссовера и мутации.
8. Метод отбора обучающего и тестового множества из входных данных (лучше случайный).
9. Используемая фитнес-функция (тип ошибки).
10. Используемые значения параметров (мощность популяции, вероятность кроссовера и мутации).
11. График обучения (падение значения ошибки в процессе обучения по итерациям-эпохам).
12. Диаграмма значений ошибки на обучающем и тестовом множестве.
13. Найденные значения коэффициентов для модели СОСОМО и формула в целом.
14. Выводы по полученным результатам.
15. Список использованных источников (литература).
16. Листинг программы (в приложении).

Контрольные вопросы

1. Приведите основную формулу методики СОСОМО.
2. Какие данные необходимы для использования методики СОСОМО?
3. Что позволяет оценивать методика СОСОМО?
4. Какие данные можно использовать при обучении (поиске коэффициентов a , b)?
5. Как можно разбить данные на обучающее и тестовое множество?
6. Какое кодирование потенциальных решений можно использовать в этой задаче?
7. Какие типы генетических операторов можно использовать в этой задаче?

8. Какую фитнес-функцию можно использовать в этой задаче?
9. Какие виды ошибок можно использовать в этой задаче?
10. Приведите основную формулу метрики абсолютных значений.
11. Приведите основную формулу метрики среднего значения относительной погрешности.
12. Приведите основную формулу отклонения.
13. Как можно улучшить методику СОСОМО?
14. Как можно применить роевой алгоритм для поиска коэффициентов a , b ?
15. Как можно применить эволюционную стратегию для поиска коэффициентов a , b ?

Литература

1. Ю.А.Скобцов. Эволюционные методы в программной инженерии. Учебное пособие. СПб:ГУАП, 2020.-130с.
2. Ю.А.Скобцов. Генетические алгоритмы в программной инженерии. Москва Вологда:Инфра-Инженерия, 2024.-144с.