

ОЦЕНКА НА ПАРЕТО ФРОНТА С ЕВРИСТИЧЕН ЕДНОЦЕЛЕВИ РЕШАТЕЛ

Тодор Балабанов

*Българска академия на науките
Институт по информационни и комуникационни технологии
ул. "акад. Георги Бончев", блок 2, кабинет 514
гр. София 1113, България
todor.balabanov@iict.bas.bg
<http://www.iict.bas.bg/>*

PARETO FRONT ESTIMATION WITH A HEURISTIC SINGLE-OBJECTIVE SOLVER

Todor Balabanov

*Bulgarian Academy of Sciences
Institute of Information and Communication Technologies
acad. Georgi Bonchev Str., block 2, office 514
1113 Sofia, Bulgaria
todor.balabanov@iict.bas.bg
<http://www.iict.bas.bg/>*

Abstract

A basic difference between single-objective and multi-objective optimization is that in single-objective optimization there is only one function for which an optimum is sought, while in multi-objective optimization there are more functions for which optimums must be achieved simultaneously. In single-objective optimization, the result of the calculations is a single value. Unlike multi-objective optimization, the result of the calculations is multiple values known as Pareto optimal solutions. Due to these features, the most common approach for solving a multi-objective problem is by reducing it to a single-objective problem. The simplest way to scale is through an additive model, in which each objective receives a weighting factor. This study proposes the use of the single-objective LibreOffice Calc NLP Solver for Pareto optimal solutions generation.

Keywords: metaheuristics; multi-objective optimization; single-objective solver.

ВЪВЕДЕНИЕ

Многокритериалната оптимизация е под- дял на общата оптимизация. Тя има значи- телна важност в практиката, тъй като мно- жество индустриални оптимизационни зада- чи се описват с набор от критерии, които критерии най-често са противоречиви [1]. Най-използваният подход за решаването на многокритериална задача е чрез свеждането ѝ до еднокритериална.

В реалната практика, най-често много- критериалните задачи са и нелинейни [2].

Поради липсата на подходяща техника за директното решаване на многокритериални- те задачи, то те биват свеждани до еднокри- териални и решавани с помощта на инстру- ментите от еднокритериалната оптимиза- ция. Сложността при многокритериалните задачи идва от факта, че решението им пред- ставлява множество от компромисни точки (множество на Парето [3]). Като паралел, ре- зултата от еднокритериалната оптимизация най-често е едно единствено решение. Из- ключение правят многомодалните задачи, където са налични повече от един глобални

оптимуми с еднакви стойности. Основна задача в многокритериалната оптимизация е да се намерят колкото се може повече Парето оптимални точки/решения. Намирането на тези решения е от съществено значение, тъй като сравнени две по две, дават съотношението на компромис между отделните целеви функции. По този начин, лицата вземащи решения ще имат по-голям набор от възможни компромиси, избирайки окончателното решение [4].

При многокритериалните задачи на входа на две или повече функции $f = \{f_1, \dots, f_m\}$ [5] постъпва един и същи входен вектор $x = \{x_1, \dots, x_n\}$. Наличието на повече от една функция за оптимизация води до появата на две пространства на търсенето – пространство на променливите и пространство на критериите [6]. По отношение на променливите, често се задава диапазон на валидните стойности, под формата на долна и горна граница. Освен диапазона за стойности на променливите, често в практиката се задават ограничения, като равенства или неравенства, които също трябва да бъдат удовлетворени. Ако дори само една целева функция, равенство или неравенство има нелинеен характер, то цялата задача се превръща в нелинейна и решенията трябва да се търсят с инструменти за нелинейна оптимизация.

Най-използваната и най-интуитивната скаларизация [7] от многокритериална задача към еднокритериална задача е чрез адитивен модел. При този модел всяка целева функция се асоциира с тегло, а така претеглените стойности на функциите се сумират (1).

$$f = w_1 f_1(x) + \dots + w_m f_m(x) \quad (1)$$

Точно този модел е приложен и в настоящото изследване. За всички разглеждани задачи се търсят минимални стойности, както на агрегираната функция, така и на отделните целеви функции. При ситуации в които за някой от критериите се търси максимум, то неговото тегло участва в скаларизиращата функция с отрицателна стойност.

В изследването се предлага използването на LibreOffice Calc NLP Solver за решаване на задачи от многокритериалната оптимизация. След въвеждащата секция, представяща задачата за търсене на Парето оптимални решения, докладът е организиран по следния начин: Секция 2 – разяснява по ка-

къв начин еднокритериалният решател в LibreOffice Calc може да се приложи за многокритериални задачи; Секция 3 – излага информация за проведените експерименти и резултатите получени от тях; Секция 4 – съставлява заключение на извършеното изследване и дава насоки за бъдещи разработки.

LIBREOFFICE CALC NLP SOLVER

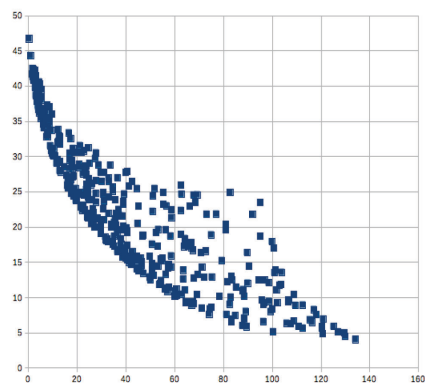
LibreOffice е софтуерен продукт с отворен код, който предлага решение за нуждите на малкия и средния бизнес. Софтуерът се състои от следните основни модули: Writer (текстообработка), Calc (електронни таблици), Impress (презентации), Draw (векторна графика), Math (математически формули) и Base (бази от данни). В модула за електронни таблици може да се използват линеен и нелинеен решател за оптимизационни задачи (Tools->Solver). Нелинейната оптимизация е значително по-съществена, тъй като е значително по-сложна от линейната оптимизация. Най-добре развитият под-модул за нелинейна оптимизация в LibreOffice Calc е хибридна комбинация между еволюция на разликите (Differential Evolution) и рояк от частици (Particle Swarm Optimization). Този под-модул е реализиран на програмния език Java, към текущата стабилна версия на софтуера. Програмистите, поддържащи под-модула, планират той да бъде пренаписан на C++, но този план е средно-към дълго-срочен. В текущото му състояние, под-модулът е предназначен единствено за решаване на еднокритериални задачи. От практическа гледна точка, това означава, че на потребителя се предлага само една единствена стойност, като крайно решение. Това е така дори за задачи в които съществуват повече от един, равностойни глобални оптимуми. Евристичната природа на двата алгоритъма в решателя, предопределят факта, че пресметнатият резултат не гарантира глобална оптималност. Най-често, полученото решение е близко до глобалния оптимум.

Ако някой има желание решателят в LibreOffice Calc да се използва за многокритериални задачи, то би било възможно Java кодът да се адаптира за тази цел. Макар и такава възможност да съществува, обемът работа би бил твърде голям и би отнел твърде много време, дори за опитни софтуерни специалисти. В същото време, наличието на

скриптов езици в LibreOffice Calc (примерно Python), дават възможност за много по-лесно адаптиране на решателя, така че да се търсят решения и на многокритериални задачи. Първата стъпка в този процес е на всеки от критериите да се приложи тегловен коефициент, спрямо описаното в уравнение (1). Втората стъпка е свързана с това, че решателят генерира единичен резултат, а в случая на много критерии се търси цяло множество от решения. Логично е решателят да се стартира многократно, така че да предложи множество от решения, близки до Парето фронта. Точно поради евристичната природа на двата алгоритъма в решателя, генерираните решения биха били близо до Парето фронта, но няма гаранция, че биха били точно на самия фронт. За всяко стартиране на решателя се подготвят различни стойности за тегловните коефициенти на критериите. Тези различни стойности на коефициентите символизират възможността на лицата вземащи решения да определят различна важност за отделните критерии. След като бъде генерирано първоначално множество от точки, лицата вземащи решения могат ръчно да коригират тегловните коефициенти и с ръчно стартиране на решателя да получават нови точки, близки до Парето фронта.

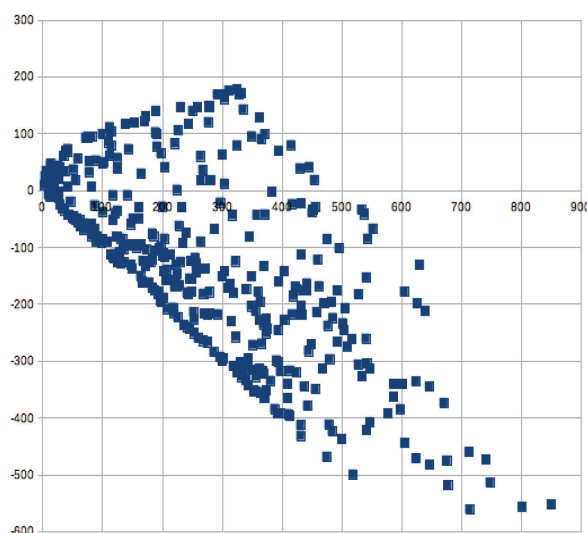
ЕКСПЕРИМЕНТИ И РЕЗУЛТАТИ

Всички експерименти с реализирани с настолна компютърна конфигурация, притежаваща следните параметри: Intel Core i5 2,3 GHz, Single CPU with 2 cores, 8GB RAM, macOS High Sierra 10.13.6, Java SE 11.0.2, LibreOffice 7.0.6.2. Скриптът на Python, стартиращ многократно решателя, е публично достъпен на [8], а числените резултати от експериментите са достъпни на [9].



Фиг. 1. Резултати за функцията Binh and Korn

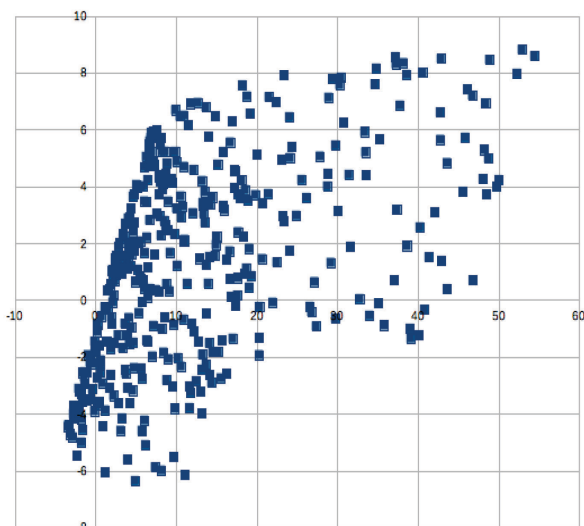
За провеждането на тестовете са избрани четири многокритериални задачи, използвани като еталон при тестовете на алгоритми за многокритериална оптимизация, а именно: Binh and Korn, Chankong and Haimes, Binh-4 и Constr-Ex [10]. И четирите задачи имат по две целеви функции, ограничения за променливите в x вектора и ограничения под формата на нелинейни неравенства.



Фиг. 2. Резултати за функцията Chankong and Haimes

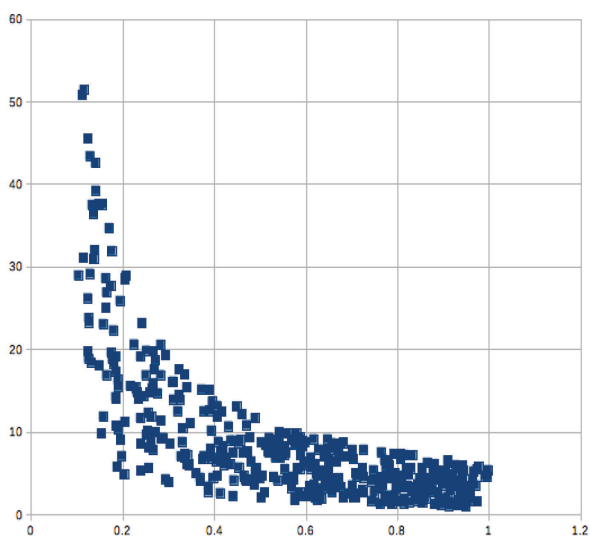
На Фиг. 1 ясно се вижда, че при функцията Binh and Korn, Парето фронтът е ясно оформен. Някои от точките на Парето фронта са по-трудно достижими (дясната част на графиката), защото изискват голяма разлика в тегловните коефициенти на критериите. Освен ясно изразеният Парето фронт, фигурата показва и множество точки в страни от фронта. Тези точки се дължат на факта, че избраният алгоритъм за изреждане на теглата по-често задава равноправни тегла на критериите, отколкото значително различаващи се. Точките са отдалечени от Парето фронта, тъй като решателят генерира субоптимални стойности и при някои стартирания, тези стойности не доближават фронта.

При функцията Chankong and Haimes (Фиг. 2), Парето фронтът също ясно се отличава, но за тази функция много по-голям брой от генерираните точки са на отдалечено разстояние от самия фронт. Това се дължи на факта, че нито решателят, нито скриптът отчитат доминираните решения от недоминираните решения.



Фиг. 3. Резултати за функцията Binh-4

При функцията Binh-4 (Фиг. 3), Парето фронтът е много слабо различим. При тази тестова функция, предложения начин за генериране на решения с нелинейния решател на LibreOffice Calc дава слаби резултати. Преобладаващата част от генерираните точки са на значително разстояние от теоретично известния Парето фронт.

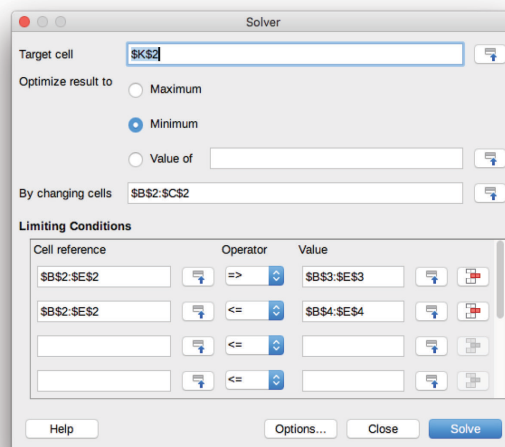


Фиг. 4. Резултати за функцията Constr-Ex

При функцията Constr-Ex (Фиг. 4), Парето фронтът също ясно изразен, но отново за генерирането на някои точки е необходим голям дисбаланс в стойностите на тегловните коефициенти на критериите.

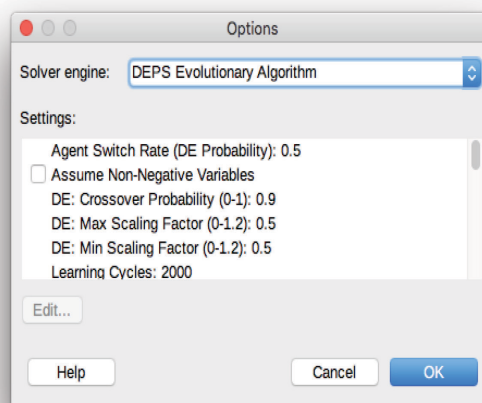
Използването на Python скрипта изисква параметрите на решателя да бъдат зададени предварително и на ръка (Фиг. 5). При текущата стабилна версия на продукта е от съ-

ществено значение да има точно зададени ограничения на свободните променливи.



Фиг. 5. Оптимизационен модел в решателя

Решателят позволява използването на няколко различни алгоритъма. За нелинейна оптимизация най-добре развита е хибридна комбинация от еволюция на разликите и рояк от частици (Фиг. 6). Точно поради тази причина тези алгоритми са приложени в настоящото изследване.



Фиг. 6. Параметри на оптимизационните алгоритми

Двата алгоритъма имат множество параметри, като най-съществен е броят оптимизационни цикли. Подразбиращата му се стойност е 2000, но тя понякога е разумно да бъде увеличена. Разбира се, по-голям брой цикли водят и до по-дълго време за изчисление, което от своя страна води до по-качест-

вени решения. Освен броя цикли, съществен е и прагът за стагнация (дълъг период без подобрене в резултата). Този праг, освен броят цикли, е вторият важен критерий за спиране на оптимизационния процес. Понякога е полезно този праг да бъде силно снижен, така че алгоритмите да работят по-дълго време, дори ако шансовете за подобрене на решението са малки.

ЗАКЛЮЧЕНИЕ

В това изследване е представена възможността за използването на еднокритериален решател в задачи от многокритериалната оптимизация. Като се дават списък от тегла на целевите функции, се генерират различни решения близо до Парето фронта. Стохастичният характер на алгоритмите в решателя не позволява предложените решения да бъдат на самия фронт, като решенията са позиционирани близо до него. От математическа гледна точка, това не е приемливо, но в реалната практика дори приближените решения са за предпочитане при липсата на други.

От проведените експерименти, ясно личи, че някои от генерираните точки са твърде далеч, спрямо Парето фронта. Като бъдеща насока за изследване, би било интересно как да се настроят параметрите на решателя, така че точките максимално да се доближат до фронта.

БЛАГОДАРНОСТИ

This research is funded by Velbazhd Software LLC and it is partially supported by the Bulgarian Ministry of Education and Science (contract D01-205/23.11.2018) under the National Scientific Program „Information and Communication Technologies for a Single Digital Market in Science, Education and

Security (ICTinSES)“, approved by DCM # 577/17.08.2018.

REFERENCE

- [1] Abraham, A., Jain, L.: Evolutionary Multiobjective Optimization. Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing, Springer, London, 1-6 (2005). ISBN 978-1-85233-787-2 DOI 10.1007/1-84628-137-7_1
- [2] Zitzler, E., Laumanns, M., Bleuler, S.: A Tutorial on Evolutionary Multiobjective Optimization. Metaheuristics for Multiobjective Optimisation, vol. 535, 3-37 (2004). ISBN 978-3-540-20637-8 DOI 10.1007/978-3-642-17144-4_1
- [3] Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. Parallel Problem Solving from Nature, vol. 1498, 292-301 (1998). ISBN 978-3-540-49672-4 DOI 10.1007/BFb0056872
- [4] Deb, K.: Multi-objective Optimization. Search Methodologies, 403-449 (2014). ISBN 978-1-4614-6939-1 DOI 10.1007/978-1-4614-6940-7_15
- [5] Emmerich, M., Deutz, A.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural Computing, vol. 17, 585-609 (2018). ISSN 1567-7818 DOI 10.1007/s11047-018-9685-y
- [6] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing, 105-145 (2005). ISBN 978-1-85233-787-2 DOI 10.1007/1-84628-137-7_6
- [7] Miettinen, K., Makela, M.: On scalarizing functions in multiobjective optimization. OR Spectrum, vol. 24, no. 2, 193-213 (2002). ISSN 0171-6468 DOI 10.1007/s00291-001-0092-9
- [8] Balabanov, T.: LibreOffice Single-Objective Solver Used for Multi-Objective Optimization. ResearchGate (2021). DOI 10.13140/RG.2.2.16761.19041
- [9] Balabanov, T.: UniTech 2021 Multi-Objective Benchmark Functions. ResearchGate (2021). DOI 10.13140/RG.2.2.24048.79363.