

ХИБРИДНА КОМБИНАЦИЯ МЕЖДУ ЛАКОМ И ГЕНЕТИЧЕН АЛГОРИТЪМ ЗА РЕШАВАНЕ НА ЗАДАЧАТА ЗА ЕДНОМЕРЕН РАЗКРОЙ

Тодор Балабанов¹, Иван Благоев¹, Зорница Атанасова¹

¹Институт по информационни и комуникационни технологии

Българска академия на науките

ул. „акад. Георги Бончев“, блок 2, кабинет 514, град София 1113, България

todorb@iinf.bas.bg

<http://www.iict.bas.bg/>

GREEDY-GENETIC ALGORITHM HYBRID SOLUTION OF 1D STOCK CUTTING PROBLEM

Todor Balabanov¹, Ivan Blagoev¹, Zornitsa Atanassova¹

¹Institute of Information and Communication Technologies

Bulgarian Academy of Sciences

acad. Georgi Bonchev Str, block 2, office 514, 1113 Sofia, Bulgaria

todorb@iinf.bas.bg

<http://www.iict.bas.bg/>

Abstract

One-dimensional stocking cut problem is a very well know industrial problem. The main idea behind it is that there are stock size beams provided by the steel factory for example. Beams should be cut in smaller pieces, according requirements given by the designers and the engineers. This is a pure combinatorial problem and the goal is minimization of the total stock size beams wastage. In this research stock size beams can be with different length, according steel factory specifications. The article proposes hybrid solution as combination between genetic and greedy algorithms. All experiments are done with custom made software solution based on JavaScript which was required by an industrial partners.

Keywords: One-dimensional cutting stock problem, genetic algorithms, greedy algorithms.

ВЪВЕДЕНИЕ

Задачата за едномерен разкрой е добре известна в различни индустриални производства. Това е класическа комбинаторна задача, при която материал оформен като пръти (например стоманени греди) трябва да бъде нарязан на предварително зададени парчета [1]. В общия случай прътите имат една и съща дължина, но в това изследване се разглежда вариация при която прътите имат няколко стандартизирани дължини. Задачата е оптимизационна и целта на оптимизацията е рязането на парчетата така да бъде подредено, че остатъците неизпол-

зван материал да се минимизира [2]. Задачата е важна за множество индустрии, като металургия, стъklarство, текстил, гледане на пчели [3,4] и други. Дори при оформянето на рекламните блокове по масовите медии задачата се решава на принципа на едномерния разкрой, така че рекламни клипове с различна дължина да бъдат поместени в определен времеви отрязък. Настоящото изследване е провокирано от нуждите на строителната архитектура, при която стоманени пръти биват нарязвани по такъв начин, че да послужат за скелет на стоманени конструкции. Поради тази причина изложението се базира на терминологията изпол-

звана в металургията и архитектурата. В настоящото изследване, използването на лакком алгоритъм от [5] е надградено с комбинация от генетичен алгоритъм. Експериментите са извършени с JavaScript програма и са използвани примери от реалната индустриална практика. Предложеният хибриден алгоритъм е стохастичен (основно заради генетичния алгоритъм) и представлява интерес за производителите на софтуер, под формата на заместител за алгоритми, като „пълно изчерпване“ или „динамично оптимизиране“.

ФОРМУЛИРАНЕ НА ЗАДАЧАТА

Добре известно е, че задачата за едноразмерен разкрой (one-dimensional cutting stock или 1D-CSP) е комбинаторна и попада в класа NP-Hard задачи [6]. По своята същина задачата е от целочисленото линейно програмиране. Когато броят на парчетата, които трябва да се нарежат е малък, то са приложими общоизвестните точни числени методи. Сложността идва при голям брой и разнообразни по дължина парчета за нарязване.

Класическият вариант на 1D-CSP представлява нарязване на стандартизиран размер пръти, на парчета с размер зададен от проектанта. Основната цел при рязането е да се минимизира общия остатък от всеки прът, след нарязването.

За стандартната дефиниция на проблема се използват следните означения:

- c - фабрични дължини на прътите;
- w_i - подредба на дължините за елемент i ;
- v_i - желано количество от елемент i ;
- x_{ij} - брой елементи i отрязани от прът j ;
- y_j - индикация дали j прът се ползва;
- t_j - остатък от прът j ;

$$\min \sum_{j=1}^m t_j \quad (1)$$

$$\sum_{i=1}^n x_{ij} w_j + t_j = c \cdot y_j, \quad j = 1, \dots, m. \quad (2)$$

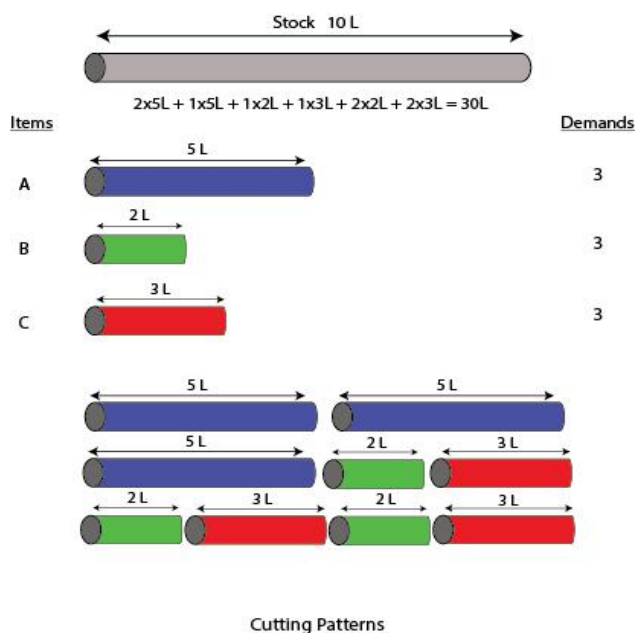
$$\sum_{i=1}^n x_{ij} = v_i, \quad j = 1, \dots, m. \quad (3)$$

$$x_{ij} \geq 0 \text{ integer}, \quad i = 1, \dots, n; \quad j = 1, \dots, m.$$

$$t_j \geq 0 \text{ integer}, \quad j = 1, \dots, m.$$

$$y_j \in \{0, 1\}.$$

Целевата функция е минимизирана на сумата от остатъците на използваните пръти (1). Ограниченията в модела са, че всички парчета трябва да бъдат нарязани и всички остатъци трябва да бъдат отчетени.



Фиг. 1. Визуално представяне на задачата [6]

На Фиг. 1 е визуално представен пример за нарязване на три вида елементи (items), от които трябва да има по три броя (demands), при фабрична дължина на пръта от 10 единици. В този пример е възможно нарязване без остатъци, като за целта се използват три пръта. В първия прът две парчета с дължина 5. Във втория едно парче с дължина 5, едно парче с дължина 2 и едно парче с дължина 3. В третия прът две парчета с дължина 2 и две парчета с дължина 3.

Приема се, че задачата е коректно поставена и всички елементи са по-малки или равни на прътите с фабрична дължина. Също така, се приема, че доставката на пръти е неограничена.

КОМБИНАЦИЯ МЕЖДУ ЛАКОМ И ГЕНЕТИЧЕН АЛГОРИТЪМ

Настоящото научно изследване предлага хибридна комбинация между лакоми и генетични алгоритми. Такова хибридно решение е полезно в ситуации, където броят на парчетата е толкова голям, че класическите точни методи работят с неприемливо за практиката изчислително време.

При разглеждания случай са възможни пръти с няколко различни производствени дължини (използваните примери са с 4 различни производствени дължини). Така формулирана задачата се разделя на две отделни комбинаторни задачи – за избор на комбинация от възможните пръти с производствена дължина и за избор на реда, в който ще бъдат рязани отделните парчета.

За първата подзадача предложението е да се използва генетичен алгоритъм, който да избере реда на прътите, които могат да бъдат с различна производствена дължина. За втората подзадача предложението е елементите за нарязване да бъдат подреждани според стъпките на лаком алгоритъм (greedy algorithm).

Генетични алгоритми

Популацията на генетичния алгоритъм се състои от индивиди (решения) представляващи масив от цели числа. Всеки индивид кодира (на случаен принцип при инициализацията) поредица от стойности, избрани от дължините на прътите с производствена дължина. За да се гарантира валидност на първоначалните хромозоми, за всеки елемент се заделя отделен прът, като се следи елементът да е по-къс или равен от елемента за който е отделен. Теоретичната обосновка за генетичните алгоритми не предлага универсални стойности за размера на популацията и поради тази причина този параметър се определя емпирично, чрез експерименти. В настоящото изследване е избрана стойността 37, на база няколкократни опити.

Операцията за селекция е случайна, с елемент на състезателен принцип и допълнителни мерки за опазване на елита в популацията. Дадена е възможност на потребителя да избира размера на елита, като подразбиране стойността е 1% от общата популация.

Като операция за кръстосване е предложено равномерно случайно кръстосване (uniform crossover).

Операцията за мутация променя случайно избран прът в хромозомата, като го замества с дължина от списъка на възможните производствени дължини на прътите.

Тъй като е възможно кръстосването и мутацията да доведат до индивид, представляващ невалидно решение, то по време на оценката (evaluation) са взети допълнителни мерки за коригиране на индивида, така че той да представлява валидно решение на задачата.

Жизнеността (fitness) на новополучените индивиди се изчислява чрез прилагане на процедурата за рязане на пръти с лаком алгоритъм. Общата сума от остатъците на всички използвани пръти се присвоява като жизнена стойност на индивида. Колкото по-малка е тази сума, толкова по-добър е индивида, по-добро е решението, респективно.

Лакоми алгоритми

Жизнеността на индивидите в популацията на генетичния алгоритъм бива оценена с помощта на лаком алгоритъм за нарязване на прътите с производствена дължина.

Лакомите алгоритми са клас точни числени алгоритми, за търсене на оптимални решения при комбинаторни проблеми. За разлика от алгоритми като пълното изчерпване и динамичното програмиране, лакомите алгоритми не гарантират откриването на глобален оптимум, но гарантират субоптимални решения, близки до глобален оптимум.

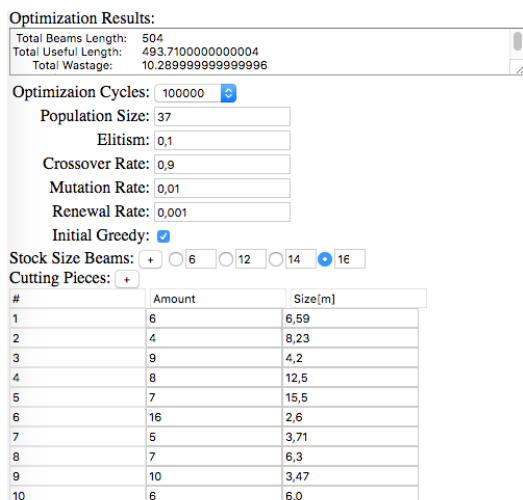
Виртуалното нарязване на прътите става по реда подаден от генетичния алгоритъм. Започва се с първия прът в подредбата на съответната хромозома. Всички елементи, които трябва да бъдат нарязани, се сортират в низходяща последователност, като на първо място се поставя най-дългият елемент. От все още ненарязаните елементи се избира най-дългото парче, което се вества в текущо рязания прът. Парчето се вади от множеството на ненарязаните елементи, а текущо рязаният прът се намалява с дължината на парчето. Тази процедура се повтаря с първия прът, подаден от генетичния алгоритъм, докато в множеството на ненарязаните парчета има такива елементи, които да се вестват в пръта. Когато цялата дължина на пръта бъде изчерпана и вече няма парчета, от ненарязаните елементи, това е остатъкът от първия прът и той се записва в общата сума на остатъците.

След нарязване на първия прът от списъка в хромозомата се пристъпва към втория и така докато бъдат нарязани всички планирани парчета и бъдат използвани всички предвидени пръти.

В процеса на виртуалното рязане с лаконичния алгоритъм са възможни две нежелани ситуации, в следствие на кръстосването и мутацията - наличие на ненужни пръти с производствена дължина и недостиг на пръти с производствена дължина. За първата ситуация, при наличие на излишни пръти, те просто се отстраняват от опашката на хромозомата и това служи за корекция на самото решение. При втората ситуация, на случаен принцип се добавят пръти с производствена дължина, докато не бъде нарязан и последният елемент. Добавянето на пръти в опашката на хромозомата модифицира решението, но това позволява то да бъде трансформирано във валидно решение.

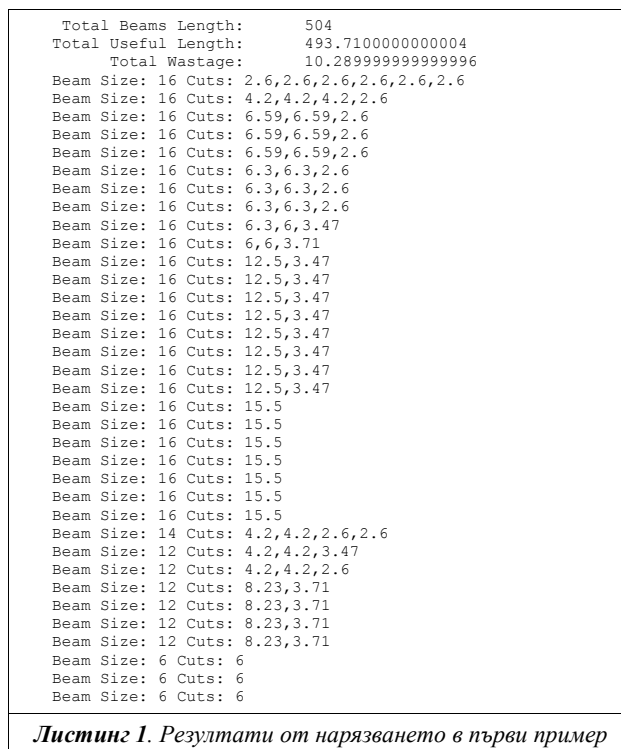
ЕКСПЕРИМЕНТИ И РЕЗУЛТАТИ

Предложената комбинация от лаконичен и генетичен алгоритъм е приложена над серия индустриални примери, като практическата реализация на алгоритмите е на езика JavaScript, по задание.



Фиг. 2. Първи пример

При първия пример (Фиг. 2) са налични десет различни размера елементи и четири различни производствени дължини на прътите.



Листинг 1. Резултати от нарязването в първи пример

Листинг 1 демонстрира по какъв начин хибридният алгоритъм решава поставената задача.

Фиг. 3. Втори пример

Във втория пример (Фиг. 3) са използвани пръти с две различни производствени дължини.

```

Total Beams Length: 520
Total Useful Length: 493.7100000000004
Total Wastage: 26.289999999999997
Beam Size: 40 Cuts:
2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6
Beam Size: 40 Cuts:
4.2,4.2,4.2,4.2,4.2,3.71,3.71,3.71,3.47,3.47,3.47
Beam Size: 40 Cuts: 8.23,8.23,6.59,6.59,6.59,3.71
Beam Size: 40 Cuts: 6.59,6.3,6.3,6.3,6.3,6.3,6.3,6.3
Beam Size: 40 Cuts: 15.5,15.5,8.23
Beam Size: 40 Cuts: 15.5,15.5,8.23
Beam Size: 40 Cuts: 12.5,12.5,12.5
Beam Size: 40 Cuts: 12.5,12.5,12.5
Beam Size: 20 Cuts: 3.47,3.47,3.47,3.47,3.47,2.6
Beam Size: 20 Cuts: 3.47,3.47,3.47,2.6,2.6,2.6
Beam Size: 20 Cuts: 6,6,4.2,3.71
Beam Size: 20 Cuts: 6.3,6.3,6
Beam Size: 20 Cuts: 6,6,6
Beam Size: 20 Cuts: 15.5,4.2
Beam Size: 20 Cuts: 15.5,4.2
Beam Size: 20 Cuts: 15.5,4.2
Beam Size: 20 Cuts: 12.5,6.59
Beam Size: 20 Cuts: 12.5,6.59

```

Листинг 2. Резултати от нарязването във втори пример

Листинг 2 показва разпределението на парчетата и как влияе наличието на пръти със значително по-голяма производствена дължина в сравнение с дължината на отделните елементи.

При третия пример (Фиг. 4) е зададен прът само с една производствена дължина. При ситуация, в която има само един прът със стандартизирана производствена дължина, популацията на генетичния алгоритъм се инициализира и с поне един индивид, който би дал оптимално решение след прилагане на лаком алгоритъм, дори и ако процедурата по оптимизация с генетичен

Optimization Results:

Total Beams Length: 540
Total Useful Length: 493.7100000000004
Total Wastage: 46.2899999999999985

Optimization Cycles: 100000

Population Size: 37
Elitism: 0,1
Crossover Rate: 0,9
Mutation Rate: 0,01
Renewal Rate: 0,001
Initial Greedy:

Stock Size Beams: + 30

Cutting Pieces: +

#	Amount	Size[m]
1	6	6,59
2	4	8,23
3	9	4,2
4	8	12,5
5	7	15,5
6	16	2,6
7	5	3,71
8	7	6,3
9	10	3,47
10	6	6,0

алгоритъм не бъде изпълнена.

Фиг. 4. Трети пример

Съществено е да се отбележи, че при всичките примери изпълнението на хибриден

ния алгоритъм отнема само няколко секунди.

```

Total Beams Length: 540
Total Useful Length: 493.7100000000004
Total Wastage: 46.2899999999999985
Beam Size: 30 Cuts:
2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6,2.6
Beam Size: 30 Cuts:
3.47,3.47,3.47,3.47,3.47,3.47,2.6,2.6,2.6,2.6
Beam Size: 30 Cuts: 3.71,3.71,3.71,3.71,3.71,3.71,3.47,3.47,3.47
Beam Size: 30 Cuts: 4.2,4.2,4.2,4.2,4.2,4.2,4.2,4.2,4.2,4.2
Beam Size: 30 Cuts: 6.59,6.59,6.59,6.59,6.59,3.47
Beam Size: 30 Cuts: 6.3,6.3,6.3,6.3,6.3,4.2
Beam Size: 30 Cuts: 6.3,6.3,6.3,6.3,6.3,4.2
Beam Size: 30 Cuts: 6,6,6,6,6,6
Beam Size: 30 Cuts: 8.23,8.23,6.59,6.59
Beam Size: 30 Cuts: 12.5,8.23,8.23
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 15.5,12.5
Beam Size: 30 Cuts: 2.6,2.6

```

Листинг 3. Резултати от нарязването в трети пример

Както е видно от Фиг. 2-4, генетичният алгоритъм се изпълнява със следните параметри:

- Брой генерирани решения: 100000
- Размер на популацията: 37
- Размер на елита: 10%
- Вероятност за кръстосване: 90%
- Вероятност за мутация: 1%
- Добавяне на случайност: 0.1%
- Начално лаком решение: Да

Изброените параметри са подбрани след множество експерименти. Тъй като уеб базираният JavaScript код позволява въвеждането на различни примери, с различен размер на елементите за нарязване и различен по брой или дължини производствени пръти, то вариации в параметрите за оптимизация с генетичния алгоритъм може да доведе до по-бърза сходимост на оптимизационния процес.

ЗАКЛЮЧЕНИЕ

Предложената комбинация между лаком и генетичен алгоритъм е продиктувана от нуждите на индустриално уеб базирано софтуерно приложение, предназначено за извършване на едномерен разкрой. Извършените експерименти потвърждават полезността на предложени хибриден алгоритъм. Евентуално бъдещо развитие би могло да се търси в посока за подмяна на лакомия алгоритъм с динамично оптимизиране. По отношение на генетичния алгоритъм могат да

се направят интересни изследвания, в посока той да бъде подменен с друга глобална оптимизационна евристика, каквато е колонията на мравките [1].

БЛАГОДАРНОСТИ

This work was supported by private funding of Velbazhd Software LLC.

REFERENCE

- [1] Evtimov, G., Fidanova, S.: Ant Colony optimization algorithm for 1D Cutting Stock Problem. Proceedings of 11th Annual Meeting of the Bulgarian Section of SIAM, FASTUMPRINT, Sofia, Bulgaria, 24--25 (2016)
- [2] Evtimov, G.: Project 2: Optimal cutting problem. STOBET Ltd., 120th European Study Group with Industry, Sofia, Bulgaria (2016)

- [3] Dineva, K., Atanasova, T.: Computer System Using Internet of Things for Monitoring of Bee Hives, Vienna, SGEM.GeoConference, vol. 17, Issue 63, 169--176 (2017)
- [4] Dineva, K., Atanasova, T.: Model of Modular IoT-based Bee-Keeping System, ESM'2017, Lisbon, EUROSIS-ETI, 404--406 (2017)
- [5] Avdzhieva, A., Balabanov, T., Evtimov, G., Kirova, D., Kostadinov, H., Tsachev, Ts., Zhelezova, S., Zlateva N.: Optimal Cutting Problem. Problems & final report of 113-th European Study Group with Industry, FASTUMPRINT, Sofia, Bulgaria, 49--61 (2015)
- [6] Tanir, D., Ugurlu, O., Guler, A., Nuriyev, U.: One-dimensional Cutting Stock Problem with Divisible Items, arXiv preprint, arXiv: 1606.01419 (2016)