

Испит из Основних оптимизационих алгоритама у инжењерству

Напомене: Испит траје до 150 минута. На усмену одбрану испитног задатка долази се са попуњеним овим документом и одговарајућим прилозима (фајловима). Студент попуњава: своје податке у табели, датум испита, као и садржаје одељака од 1 до 5, уносећи одговарајуће податке уместо објашњења за попуњавање. Задатак се ради коришћењем рачунара, а коришћено развојно окружење или одговарајући софтверски пакет морају бити доступни за време испита. Студент може понети и свој рачунар. Задатак носи до 30 поена.

Индекс (година/број)	Презиме и име	Датум	Поени	Испит	Оцена
2021/0381	Обрадовић Божић				

1. Опис проблема оптимизације

Приликом склапања тимова циљ спортских директора је балансирање опречних захтева по питању плате тима и његовог квалитета. Квалитетнији играчи који чине костур тима захтевају веће плате и уносније уговоре (у складу са њиховим могућностима и значају за успех тима) док са друге стране власници посматрају спортски тим као бизнис и желе да остваре што већи профит за шта је један од начина ангажовање играча са мањим платама. Критеријуми максимизације квалитета екипе и минимизације њене плате су у овом случају супростављени па не постоји једно оптимално решење.

Инжењерски посматрано ово је проблем вишеструке оптимизације и његово решење може да се представи парето фронтом, што је управо циљ ове оптимизације. У овом пројекту коришћени су јавно доступне плате играча у НБА лиги из сезоне 2020/21 док је мера њиховог квалитета увезена из видео игре NBA 2K22 (њихов *overall rating* тј. оцена).

Овај проблем припада класи SAT проблема; наиме, екипа може да се посматра као низ бита дужине једнаке укупном броју расположивих играча и у таквом запису i -ти бит има вредности 1 ако је играч i члан екипе. С обзиром да је велики број играча на располагању (преко 400) коришћен је мало другачији запис, конкретно, екипа је представљена као низ (различитих!) играча x унапред задате дужине (x_1, x_2, \dots, x_N) .

Играч x_i је дефинисан као структура која садржи идентификациони број, име, презиме, позицију (бек, крило, високи), квалитет (цео број између 65 до 100) и плату (у доларима). Коришћен формат екипе је такав да она садржи 12 играча (4 бека, 5 крила и 3 висока, тим редом). Један играч може да игра за више екипа али у једној екипи исти играч не може да се нађе више од једном.

2. Дефиниција оптимизационе функције и оптимизационог простора

Овај оптимизациони проблем има два критеријума: максимизацију квалитета тима и минимизацију плате тима, самим тим оптимизационе променљиве су плата тима и његов квалитет.

2.1. Плата тима

Правила НБА лиге налажу различите границе у виду плате тимова (*salary cap*¹, *luxury cap threshold*²). У овом пројекту постојећи систем је значајно поједностављен; уведено је ограничење да збир плата свих играча у екипи мора да буде мањи од фиксне вредности *sal_max* која износи *luxury cap threshold* за сезону 2021/22

¹Горња граница новца коју екипе смеју да исплате играчима.

²НБА лига има "мекан" *salary cap*, тј. постоје правила и изузеци којима он може да се пређе. Све док је плата тима мања од *luxury cap threshold* власник тима не плаћа казне због пробијања *salary cap*-а. У случају да је плата тима изнад *luxury cap threshold* власник тима из свог џепа плаћа казну за сваки долар потрошен изнад *salary cap*.

тј. 136.606 милиона долара.

Коришћена оптимизациона функција sal је збир плата свих играча тима:

$$sal = \sum_{i=1}^N x_i.salary \quad (1)$$

Такође, проблем може да се дефинише тако да се тражи максимум оптимизационе функције дефинисане као

$$sal' = sal_max - sal = sal_max - \sum_{i=1}^N x_i.salary \quad (2)$$

Приликом сваке промене садржаја тима (тј. замена играча) мора да се провери да ли је $sal < sal_max$ (или $sal' > 0$).

2.2. Квалитет тима

Квалитет тима је рачунат по следећем алгоритму изворно преузетом из видео игре FIFA 22 (пошто не постоји јавно доступно како се он рачуна у NBA 2K22) и благо модификованом (подебљани делови):

1. Израчунају се збир оцена свих играча t_rate и просечна вредност индивидуалних оцена t_ovr :

$$t_rate = \sum_{i=1}^N x_i.ovr, \quad t_ovr = \frac{t_rate}{N} \quad (3)$$

2. За све играче који имају оцену већу од t_ovr разлика њихове оцено и t_ovr **помножена њиховим количником** се додаје на t_rate :

$$t_rate += \sum_i \sigma_i \frac{x_i.ovr}{t_ovr} (x_i.ovr - t_ovr) \quad (4)$$

где је

$$\sigma_i = \begin{cases} 1, & x_i.ovr - t_ovr > 0 \\ 0, & \text{иначе} \end{cases} \quad (5)$$

3. Коначна просечна оцена ovr се добија као

$$ovr = \frac{t_rate}{N} \quad (6)$$

Додатна вредност додата t_rate представља утицај квалитетнијих играча, тј. њихову могућност да својим учинком додатно допринесу. Подебљана корекција је уведена услед тога што је описани утицај већи у кошарци него у фудбалу (у кошарци је мање играча на терену па један играч више утиче на саму игру тј. њен исход).

Оптимизациони простор је стриктно дефинисан условом да плата тима мора да буде мања од sal_max па се све плате налазе у интервалу $(0, sal_max)$. Што се квалитета тиче не постоје стриктне границе нити процес оптимизације захтева да се уведу; с обзиром да највећи број играча има оцено у опсегу $(70, 80)$ може се очекивати да ће квалитет екипа бити у опсегу $(70, 90)$.

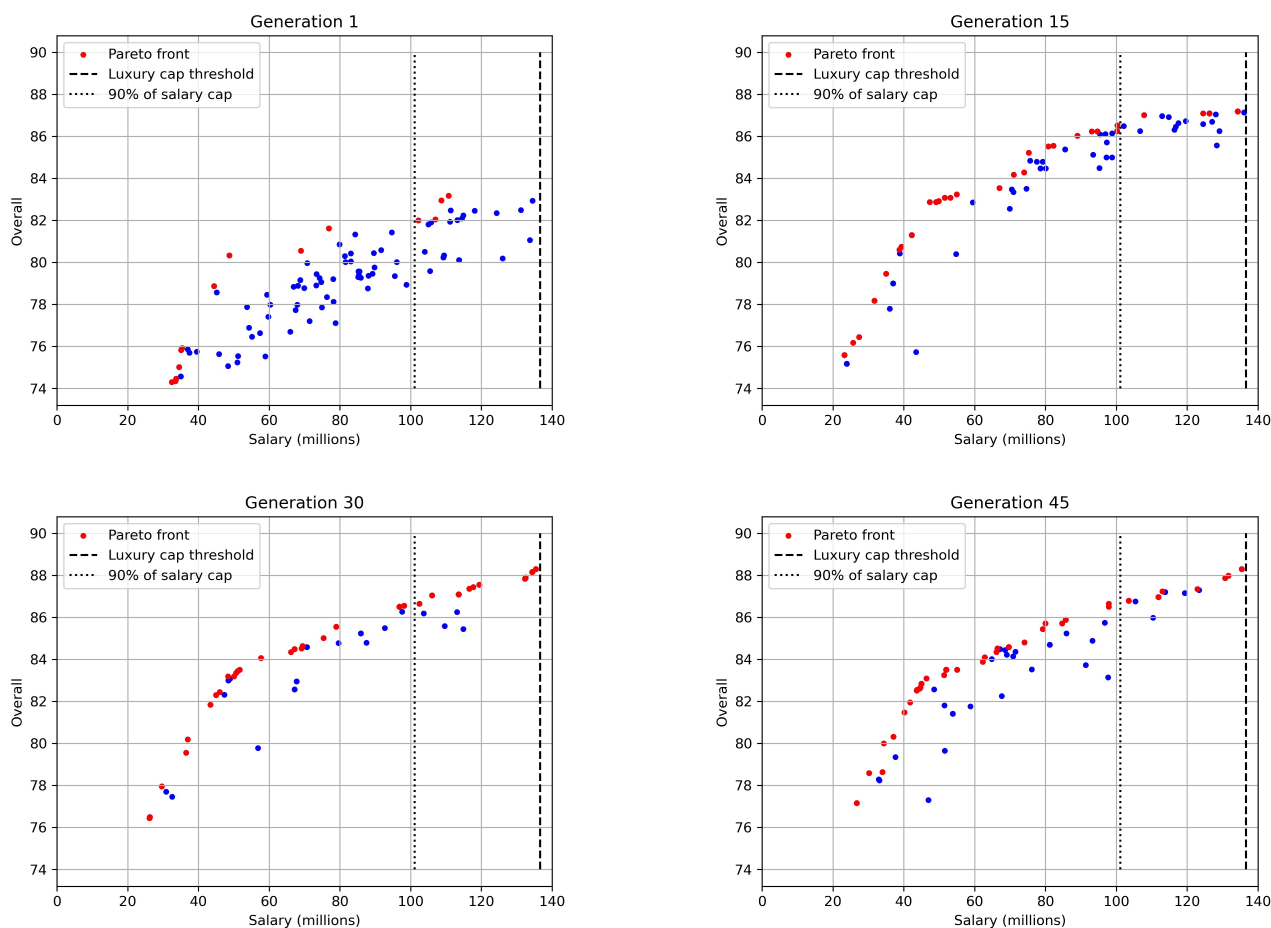
3. Оптимизациони алгоритам или алгоритми

У овом пројекту коришћен је NSGA (Non-Sorted Genetic Algorithm). Овај алгоритам се заснива на стандардном генетском алгоритму са изменама у делу за рачунање описне функције. За сваки критеријум израчунавају се описне функције за свако решење у генерацији, након чега се проналазе сва *nondominated solutions* која одговарају процењеном парето фронту у текућој генерацији и она чине ранг 1. Поступак се овако понавља док сва решења нису рангирана (ранг 2, 3 итд...). Описна функција на основу које се бирају решења за укрштање се рачуна на основу ранга (први ранг је бољи од осталих) и такође подстиче распрострањеност решења у свим деловима оптимизационог простора. Укрштање и мутација се раде на стандардан начин примерен генетским алгоритмима; парето фронт се памти и побољшава током генерација (тј. током оптимизације).

Почетно решење, тј. почетна популација је одређена на случајан начин одабиром из одговарајућих група (бекови, крила, високи). Део алгоритма који се бави рачунањем парето фронтова и описне функције је написан по упуштвима из [1]. Укрштање је вршено тако што су два тима мењала по 3 играча на насумично одабраним позицијама. Мутација је представљена променом играча на једној насумично одабраној позицији са насумично одабраним играчем који игра ту позицију. Након обе операције извршена је провера да ли новонастали тимови садрже “дупликат” играче и да ли задовољавају ограничење по питању плате.

4. Резултати

Почетну популацију је чинило 100 екипа са, као што је раније наведено, по 12 играча. Код је покренут за 45 генерација са вероватноћама укрштања и мутације од 0.8 и 0.15, респективно. Различите генерације, њихови парето фронтови и релеватне границе³ су приказане на Слици 1.



Слика 1: Приказ различитих генерација

Са порастом генерација уочава се формирање, обликовање и стабиловање парето фронта (тј. његове процене).

5. Прилози

За писање кода коришћен је *Python* програмски језик и он је реализован у *Spyder* окружењу. Код је сачуван и приложен под називом *oaa_bozidar-obradovic.py*. Током извршавања код прави и попуњава четири пара (тј.

³Јако ретко екипе имају плате испод 90% *salary cap*-а, такође, ако трошкови на плате играча током сезоне не пређу ову границу екипа је у обавези је да новац вишка расподели играчима на крају сезоне. *Salary cap* за сезону 2021/22 износи 112.414 милиона долара па је вредност приказане границе 101.173 милиона долара

8 укупно) текстуалних фајлова (за генерације 1, $N_{gen}/3$, $2N_{gen}/3$ и N_{gen} , где је N_{gen} број генерација): фајл са подацима о целој генерацији и фајл са подацима о њеном парето фронту. Такође, за сваку од горенаведених генерација код избацује и чува графички приказ (Слика 1). Осим кода приложен је и фајл *official 2k22 data.csv* у којем се налазе подаци о играчима (име, презиме, позиција, плата) који се увозе и даље обрађују у коду.

Уз урађени задатак обавезно се прилаже (1) код програма који извршава оптимизацију, или (2) пројекат коришћен за оптимизацију у одговарајућем софтверском пакету. Навести имена приложених фајлова (или пројеката), као и кратак опис садржаја приложених фајлова, у једној до две реченице. Уколико постоји више прилога, пожељно је да буду запаковани у један zip-фајл. Овај попуњени документ, заједно са прилозима, послати на адресу olcan@etf.rs до почетка испитивања.

Литература

- [1] Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: John Wiley & Sons, Inc., 2001. ISBN: 047187339X.