

# Raspoređivanje višeprosesorski zahtevnih taskova na 3 procesora

Seminarski rad u okviru kursa

Računarska inteligencija

Matematički fakultet

[overview verzija](#)

Ivana Jordanov

ivanajordanov47@gmail.com

20. jun 2020.

## Sažetak

Raspoređivanje taskova tako da izvršavani redom zauzmu što manji vremenski interval je NP težak problem koji će biti opisan i rešavan u ovom radu. Pod rednim izvršavanjem podrazumeva se da se u jednom trenutku samo jedan task može izvršavati na jednom procesoru, ali je zato paralelno izvršavanje više taskova na različitim procesorima dozvoljeno. Ovaj NP težak problem rešavan je algoritimima optimizacije. NP-težak, Raspoređivanje, Optimizacija

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Opis problema</b>	<b>2</b>
<b>3</b>	<b>Algoritmi</b>	<b>2</b>
<b>4</b>	<b>Rezultati</b>	<b>2</b>
<b>5</b>	<b>Zaključak</b>	<b>2</b>
	<b>Literatura</b>	<b>2</b>

## 1 Uvod

Rasporedjivanje višeprocesorskih taskova je NP-težak problem [1].

U okviru rada predstavljeni su algoritmi kojima je problem opisan u poglavlju 2. Najpre opis odabrane strukture podataka, a zatim i algoritmi uniformnog interfejsa dati su u poglavlju 3. Konačni rezultati i njihovo poređenje mogu se pročitati u poglavlju 4

## 2 Opis problema

MINIMUM 3-DEDICATED PROCESSOR SCHEDULING

INSTANCE: Set  $T$  of tasks, set  $P$  of 3 processors, and, for each task  $t \in T$ , a length  $l(t) \in \mathbb{Z}^+$  and a required subset of processors  $r(t) \subseteq P$ .

SOLUTION: A schedule for  $T$ , i.e., a starting time function  $s: T \rightarrow \mathbb{Z}^+$  such that, for any two tasks  $t_1$  and  $t_2$  with  $r(t_1) \cap r(t_2) \neq \emptyset$ , either  $s(t_1) + l(t_1) < s(t_2)$  or  $s(t_2) + l(t_2) < s(t_1)$ .

MEASURE: The makespan of the schedule, i.e.,  $\max_{t \in T} (s(t) + l(t))$ .

## 3 Algoritmi

Backtracking, Bnb, simulirano kaljenje, hibridni algoritam od simuliranog kaljenja.

## 4 Rezultati

Backtracking - 20 min za instancu od 10 elemenata

Bnb - malo bolji

SA - gornje navedeno resava u par sekundi, ali cesto gresi na cak i na malim ulazima

Hibridni - resava tacno ulaze od oko 30,40 elemenata, i sa duzinama taska izmedju 1 i 7 otkucaja. Sa insancama oko 100 u oko 9% slucajeva daje tacno resenje

## 5 Zaključak

U radu je pokazano kako se malim promenama algoritama optimizacije mogu dobiti značajne razlike u efikasnosti. Sa nemogućnosti da se problem od preko 15 taskova reši za manje od 20 minuta, dolazi se do toga da problem od 30 taskova biva rešen za manje od 2 sekunde i to sa optimalnim rešenjem. Na uštrb efikasnosti, implementirani algoritam ne garantuje optimalno rešenje. Ono što može da se uradi po tom pitanju implementacija novog traženja suseda za dati algoritam. Neka naredna verzija ovog hibridnog algoritma mogla bi da uvede prilagođeno traženje suseda, u zavisnosti na odmaklost konvergencije. Slično algoritmu optimizacije rojem čestica. Druga opcija je reimplementacija funkcije za traženje suseda tako da "pametno" bira gde će da ugnjezdi selektovani task.

## Literatura

- [1] MINIMUM 3-DEDICATED PROCESSOR SCHEDULING. <https://www.csc.kth.se/~viggo/wwwcompendium/node188.html>. Accessed: 2020-06-10.