

## Accepted Manuscript

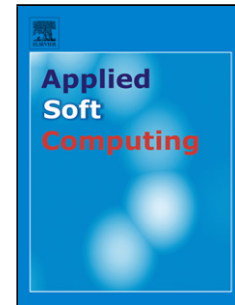
Title: A Local Search based Approach for Solving the Travelling Thief Problem

Author: Mohamed El Yafrani Belaïd Ahiod

PII: S1568-4946(16)30506-3  
DOI: <http://dx.doi.org/doi:10.1016/j.asoc.2016.09.047>  
Reference: ASOC 3846

To appear in: *Applied Soft Computing*

Received date: 4-4-2015  
Revised date: 5-5-2016  
Accepted date: 29-9-2016



Please cite this article as: Mohamed El Yafrani, Belaïd Ahiod, A Local Search based Approach for Solving the Travelling Thief Problem, *Applied Soft Computing Journal* (2016), <http://dx.doi.org/10.1016/j.asoc.2016.09.047>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

---

**Algorithm 1** JNB (first-fit version)

---

```
1  $x \leftarrow \text{initial tour}$  ▷ we always use a Lin-Kernighan tour
2  $z \leftarrow \text{initial picking plan}$ 
3 repeat
4   Evaluate  $G$ ,  $t$ , and  $p$  for  $(x, z)$ 
5   Update  $t^{\text{map}}$ ,  $t^{\text{acc}}$ , and  $w^{\text{acc}}$ 
6   for  $i \leftarrow 2 \dots n - 1$  do
7      $\bar{x} \leftarrow \text{swap}(x, i)$ 
8      $\overline{t^{\text{map}}} \leftarrow \text{tour mapper for } (\bar{x}, z)$ 
9      $\overline{t^{\text{acc}}} \leftarrow \text{recover time accumulator for } (\bar{x}, z)$ 
10     $\overline{w^{\text{acc}}} \leftarrow \text{recover weight accumulator for } (\bar{x}, z)$ 
11    for  $k \leftarrow 1 \dots m$  do
12       $\bar{z} \leftarrow \text{bitflip}(z, k)$ 
13      if no space left then skip iteration end if
14      calculate  $\Delta_w$  and  $\Delta_p$ 
15       $\bar{p} \leftarrow p + \Delta_p$ 
16       $i_{BF} \leftarrow t_{A_k}^{\text{map}}$  ▷ index of bit-flip
17      if  $i_{BF} = 0$  then  $\bar{t} \leftarrow 0$  else  $\bar{t} \leftarrow \overline{t_{i_{BF}-1}^{\text{acc}}}$  end if
18      for  $r \leftarrow i_{BF} \dots n$  do
19         $w_c \leftarrow \overline{w_r^{\text{acc}}} + \Delta_w$ 
20         $\bar{t} \leftarrow \bar{t} + \frac{d_{\bar{x}_r, \bar{x}_{r+1 \bmod n}}}{v_{\text{max}} - w_c * C}$ 
21      end for
22       $\overline{G} \leftarrow \bar{p} - R * \bar{t}$ 
23      if  $\overline{G} > G$  then break loop end if ▷ first fit
24    end for
25    if  $\overline{G} > G$  then break loop end if ▷ first fit
26  end for
27  if  $\overline{G} > G$  then
28     $x \leftarrow \bar{x}$ 
29     $z \leftarrow \bar{z}$ 
30  end if
31 until  $\overline{G} \leq G$ 
```

---

Download English Version:

<https://daneshyari.com/en/article/4963502>

Download Persian Version:

<https://daneshyari.com/article/4963502>

[Daneshyari.com](https://daneshyari.com)