A Genetic Algorithm for the Two–level Location Area Planning in Telecommunication Networks

Ons Abdelkhalek
LARODEC Laboratory,
Institut Supérieur de Gestion,
University of Tunis,
2000 Le Bardo, Tunisia,
Email: abdelkhalek.ons@gmail.com

Saoussen Krichen
LARODEC Laboratory,
Faculty of Law,
Economics and Management,
University of Jendouba,
Jendouba, Tunisia,
Email: saoussen.krichen@isg.rnu.tn

Adel Guitouni
Gustavson School of Business,
University of Victoria,
Victoria British-Columbia, Canada
Email: adelg@uvic.ca

Abstract—With the growing number of mobile users, the management of wireless networks to expand network capacity is of increasing importance. The Location Areas Planning (LAP) problem examines the redistribution of network resources to prevent any degradation in the quality of service. The optimization of these resources minimizes the cost of the registration signaling generated by the procedures of mobility. This paper introduces a new formulation of the “Bi-objective Location Area-Planning” (BOLAP) problem. Modeled as a Two–level assignment problem, we minimize two objectives successively: the location Update (LU) then the cost of the BTS–BSC links connections. The model is then iterated until no more improvement is performed in the set of optimal solutions.

We propose to adopt a Genetic Algorithm (GA) to solve the Two–level BOLAP model. Applied on a sample of real instances with different sizes for a big Tunisian telephony operator, our method generates a set of potentially efficient solutions of a good quality in a practicable CPU time. A comparison with the vector evaluated particle swarm optimization (VEPSO) is also reported.

Index Terms—Bi-Objective problem, Location Area Planning problem, Bi–level programming problems, Telecommunication network, GA, VEPSO

I. INTRODUCTION

A mobile terminal is informed by an incoming call with a paging message sent to every cell in the network for each call, which is obviously a waste of radio bandwidth. On the other hand, the mobile could notify the system, via a location update (LU) messages, of its current location at the individual cell level. This would require paging messages to be sent to only one cell, but would be very wasteful due to the large number of location updating messages that are required only when moving between location areas (LAs).

A compromise solution is to group cells into location areas as to minimize the inter-LA border by reducing the Location update between LAs. The problem of grouping cells into location areas, called the “Location Area Planning” (LAP) problem, is an optimization problem that reflects the quality of network management and the search cost that tends to minimize the LU.

LAP problem has an important impact in cellular networks because of the trade-off caused by paging and LU signaling. The management of areas is due to the increasing number of cells in the LAs. The purpose of the LAP problem is then to reduce the traffic on the paging channel, freeing them for reception or emission of calls, in order to redistribute the resources before any quality of service degradations.

The LAP problem was first introduced by Markoulidakis and Sykas [10] as a single objective problem of assigning cells into LAs, trying to minimize LU with fixing a paging boundary.

As the LAP problem is \( NP \)-complete [1], heuristics are promising for approaching the problem. Several heuristics were proposed to solve the LAP: Tabu Search (TS) algorithms [3], Greedy Search (GS) algorithms [13], Simulated Annealing (SA) [4] [5], Genetic Algorithms (GA) [15] [2] and Gondim (1996) [6], and Grouping GAs (GGA) [15] [14]. We can notice that most researches reported in the literature focused only on the uni-objective LAP.

Nevertheless, a more realistic modeling of the problem should involve two objectives. We can clearly notice that, if we increase the cost of assigning a base transceiver station (BTS) to a base station controller (BSC), paging loads in the LAs increase and thus reduces the LU. A high LU traffic is due to small location area zones (The visitor location registers (VLR) service areas are small which leads to small location zones) because of the high concentration of mobile subscribers or to high mobility between the VLR service areas. Consequently, adding a second objective function yield to a better exploration of the solution space.

In this paper, we address the Bi-objective model for the Location Area-Planning (BOLAP) problem minimizing both the LU and the distance cost between BTS-BSC. The main purpose is to cluster the cells into disjoint LAs such that the paging load represented by our constraints in all LAs does not exceed the thresholds while generation a compromise between LU load in the different LAs and the distance cost between BTS to BSC and BTS to LA.

To the two-dimensional nature of our model, horizontally
by grouping cells into LAs and vertically by assigning BTS to BSC, we propose to formulate the BOLAP problem, not as a classic bi-objective model but as a bi–level optimization problem [11]. The BOLAP is thus decomposed into two single objective problems in a way that the first problem is embedded within another one. An iterative process is then performed until it reaches an equilibrium assignment process.

We, then, develop a Genetic Algorithm (GA) approach [15] due to its ability to tackle the high complexity of the problems and to generate a promising approximation of the efficient set. A comparison with the vector evaluated particle swarm optimization (VEPSO) [9] is also reported using a set of instances with different sizes from a real case study for a GSM network.

The rest of the paper is structured as follows. In section II, we provide a brief description of our problem modeling. Section III, presents the GA algorithm to solve the BOLAP problem. In section IV, the performance of our GA algorithm is then presented and applied on a bench of realistic problem set for a GSM network then compared with a VEPSO.

II. A Bi–level process for the BOLAP problem

The bi–level BOLAP problem is a combination between the basic LAP problem [5] and a variant of the cell-to-switch assignment (CSA) problem [4], applied to the bi-objective framework. It includes a set of cells \(C\), a set of BTS, a set of BSC, and one Mobile Switching Center (MSC) with a Two-tier structure hierarchy in the network. Note that in our network, each BTS contains a set of cells \(\{C_1, C_2, ..., C_n\}\), each BSC contains a set of BTS and each MSC manage a set of BSC.

The main purpose is to find the appropriate border of an area by redistributing cells into LAs while minimizing the total registration signaling and the distance cost linking BTSs to BSCs and BTSs to LAs. This bi-dimensional optimization problem affects the entire base station subsystem in the GSM networks architecture as it minimizes the inter-LA border and the inter-BSC border. All these choices must satisfy a set of network constraints.

A. Notation

\[N\] The set of cells \(\{C_1, C_2, ..., C_N\}\)
\[M\] The set of LAs \(\{l_1, l_2, ..., l_M\}\)
\[Q\] The set of BTS \(\{j_1, j_2, ..., j_Q\}\)
\[W\] The set of BSC \(\{i_1, i_2, ..., i_W\}\)
\(c_i\) Call traffic of cell \(i\) (in Erlang)
\(r_i\) Number of TRXs for each cell \(i\)
\(h_{is}\) HO rate from cell \(i\) to cell \(s\), \(i \neq s\)
\(w_{is}\) = 1 if cell \(i\) and cell \(s\) in the same LAs
\(v_{ir}\) = 0 otherwise
\(x_{ir}\) = 1 if cell \(i\) is assigned to the LA \(r\)
\(= 0\) otherwise
\(y_{jkr}\) = 1 if BTS \(j\) is assigned to the BSC \(k\) and to LA \(r\)
\(= 0\) otherwise

\(a_{ij}\) = 1 if cell \(i\) is assigned to BTS \(j\)
\(= 0\) otherwise
\(v_{ir}\) Presets: forcing the assignment of cell \(i\) to LA \(r\)
\(\lambda_i\) the average number of incoming calls to cell \(i\) per unit time
\(P_{B_r}\) The paging bound per LA \(r\).
\(d_{jk}\) Distance cost between BTS \(j\) and BSC \(k\)
\(N_{BSC}^k\) Maximum number of links of BSC \(k\) to
\(C_{BSC}^k\) Call traffic threshold of BSC \(k\) (in Erlang)
\(R_{BSC}^k\) TRX maximum capacity of BSC \(k\)

B. Bi–level optimization problem for the BOLAP

The bi–dimensional nature of the problem leads us to naturally decompose the BOLAP model into two hierarchical optimization problem based on the bi–level optimization process, as follows:

- **Upper level** that corresponds to the basic LAP problem. It aims to minimize the LU costs while fixing a threshold of paging load for new LAs. The decision variables \(x_{ir}\) are the variables of the leader in the bi–level optimization process.

![Fig. 1. The bi-level BOLAP Problem](image)

\[
\text{Min}\ Z_1(x, y) = \sum_{i=1}^{N} \sum_{s=1}^{N} h_{is} w_{is} (1.1)
\]
\[
\text{S.t.}\ \sum_{i=1}^{N} x_{ir} = 1 \forall i = 1, ..., N (1.2)
\]
\[
w_{is} = 1 - \sum_{j=1}^{M} x_{ir} x_{js} \forall i, s = 1, ..., N \neq s (1.3)
\]
\[
x_{ir} = v_{ir} \forall i, r = 0 \text{ or } 1 (1.4)
\]
\[
P_{B_r} = \sum_{i=1}^{N} x_{ir} \lambda_i \geq 0 \forall r = 1, ..., M (1.5)
\]
\[
x_{ir}, w_{is} \in \{0, 1\} \forall i, s = 1, ..., N; r = 1, ..., M (1.6)
\]

\[
\text{Min}\ Z_2(x, y) = \sum_{k=1}^{W} \sum_{j=1}^{Q} d_{jk} \sum_{r=1}^{M} y_{jkr} (2.1)
\]
\[
\text{S.t.}\ \sum_{k=1}^{W} \sum_{j=1}^{Q} y_{jkr} = 1 \forall j = 1, ..., Z (2.2)
\]
\[
\sum_{j=1}^{Q} y_{jkr} = N_k \forall k = 1, ..., W (2.3)
\]
\[
\sum_{k=1}^{W} \sum_{j=1}^{Q} a_{ij} y_{jkr} \forall j = 1, ..., Z, r = 1, ..., M (2.4)
\]
\[
\sum_{k=1}^{W} \sum_{j=1}^{Q} y_{jkr} \leq C_{BSC}^k \forall k = 1, ..., W (2.5)
\]
\[
\sum_{r=1}^{M} \sum_{j=1}^{Q} y_{jkr} \leq R_{BSC}^k \forall k = 1, ..., W (2.6)
\]
\[
y_{jkr} \in \{0, 1\} \forall j, k, r (2.7)
\]

- **Lower level** which is a variant of the basic CSA problem. The main objective is to minimize the distance cost linking BTSs to BSCs and BTSs to LAs while satisfying both proximity and capacity constraints of the network. The decision variables \(y_{jkr}\) are the variables of the
follower as it interacts with the upper level when they make decisions.

In the upper level problem, we create a border in the LA just as it reaches an appropriate paging load in order to optimize the traffic on radio bandwidth and for releasing channels for incoming and outgoing calls. This decision is then communicated to the lower level model through the leader variable \( x_{ir} \).

The second optimization problem takes into consideration the leader variable to find an optimal solution \( Z_2(x, y) \). Both proxi mity constraints of cells (e.g. cells of the same BTS must be in the same LA) and the capacity constraints (e.g. TRX, Traffic and the number of links) are considered, because when all cells of a site are connected to a BSC or assigned to a LA, they create that amount of load limit.

For the next iterations, the optimal solution of the cost function is reported to the upper level as a constraint \( Z_2(x, y) \). Both equations (1.4) and (1.5) ensure that if \( w_{ir} = 0 \) means that cells \( i \) and \( s \) are assigned to the same LA and 1 otherwise. In some situations, the network designer or the decision maker would like to pre-assign a cell \( i \) and \( s \) to LA \( r \) which is captured by constraints (1.1) and (1.2). The paging bound is enforced by equations (1.5). The binary constraints of the decision variable \( x_{ir} \) are ensured in equation (1.6).

D. Lower level

The second objective function (2.1) minimizes the total distance cost linking BTSs to BSs and BTSs to LAs in order to improve the quality signal transmission when BTS is assigned to a particular LA.

In equations (2.2), we assume that each BTS must be assigned to only one BSC and one LA. We assume in equations (2.3) that the total number of BTS assigned to LA \( r \) is equal to the total number of BTS present in the LA for our current network. Equations (2.4) are the proximity constraints which ensure that if cell \( i \) and \( s \) belongs to the same BTS \( j \), they must be assigned to the same LA \( r \) as BTS \( j \). Equations (2.5), (2.6) and (2.7) represent the traffic capacity, the TRX capacity and links limitations for each BSC. The decision variables \( y_{jkr} \) are binary as captured in constraints (2.8).

III. GENETIC ALGORITHM TO SOLVE BOLAP PROBLEM

As the LAP problem is \( NP \)-complete [1] and the CSA is an \( NP \)-Hard problem [4], we can clearly conclude that our two-level BOLAP problem is also \( NP \)-Hard. The use of heuristics is unavoidable to solve this problem. We propose to apply a meta-heuristic algorithm due to its ability to give near optimal solutions and to be adapted to complex environment.

This may lead in adopting one of the model-based heuristics. Thus, we propose to adopt the Genetic Algorithm as solution approach since it gave good results for the LAP problems [7][14][15].

A. Encoding strategy

A chromosome consists of a solution represented by \( N \) Cells to be assigned. Each cell is defined by a substring constructed in two stages with \( W + M \) genes. The first \( M \) genes indicate the location area in which the cell belongs where the last \( W \) genes indicate the assignment of the BTS, to which the cell belongs, to a certain BSC. In the first stage, we assign the cell \( i \), \( i = \{1, 2, ..., N\} \) to the LA \( r \), \( r = \{1, 2, ..., M\} \) in a way that a cell is assigned to only one LA at the time. Based on the first allocation, as a second stage, we assign BTS \( j \), \( j = \{1, 2, ..., W\} \) (belonging to cell \( i \)) to BSC \( k \), \( k = \{1, 2, ..., Z\} \) with respect to the paging, proximity and capacity constraints. Figure 2 illustrates this procedure of encoding.

B. Initial population

In each population, a chromosome has an \( N \)-length equal to the number of cells to be assigned in our network. The initial population is randomly generating the assignment of substrings in a chromosome, where each cell \( i \) is randomly assigned to a LA \( j \). Then Based on that, a BSC \( k \) is assigned to a BTS \( r \) with respect to constraints (1.1) and (1.2).

Each chromosome is then adjusted in order to fit to our constraints and thus come-up with a set of feasible solutions that will represent our initial solution.

C. Fitness Evaluation

The dominance relationship is used for fitness evaluation. The proposed fitness evaluation is based on NSGA II [8] with our two objective functions \( Z_1 \) and \( Z_2 \).

D. Sorting process

In each iteration a sorting process is activated in order to generate the list of non-dominated set \( ND_i \). The non-dominated sorting approach (NDS) is used only to rank the best \( P_i \) individuals and thus assign a rank to each Pareto-fronts.
E. Crossover

For each iteration, a one-point crossover process is performed based on these steps:

1) two parents chromosome \(c h_1\) and \(c h_2\) are randomly chosen from the current population \(P_0\)
2) a probability \(prop\) is randomly generated and a crossing point \(I_3\) is randomly generated from \(c h_1\) and \(c h_2\)
3) if \(prop < (1 - p_c)\) then \(substring_1\) and \(substring_2\) are built as shown in figure 3
4) else substrings are equal to chromosomes parents

F. Mutation

For each iteration, a probability \(p\) is randomly generated where the probability of mutation is set to \(p_m = 0.02\) [15]. If \(p > p_m\) then the mutation process is activated on a random chromosome by making small changes on the assignment of cells to the existing LA. These changes are presented either by randomly removing an existing LA and assigning a new one or by assigning a new BSC to a certain BTS.

G. Stopping criteria

The whole process is stopped if a prefixed number of iterations \(I_{max}\) has been accomplished without improving the set of optimal solutions obtained.

The basic outline of the algorithm is summarized as follows:

**Bi-Objective Genetic Algorithm**

\[
\begin{align*}
\text{if } & \text{ } 0 \\
& \text{Randomly generate the first population } P_0 \text{ of size } N \\
& \text{Repair the first population to make it feasible} \\
& \text{Fitness evaluation of } P_0 \\
& \text{Apply the sorting process for each solution based on the fitness evaluation} \\
& \text{Do} \\
& \text{One-point Crossover: generate the set } R_i \text{ of offspring of size } N \\
& \text{If } p > p_m \text{ Then apply the Mutation process} \\
& \text{Sort the set of offspring in } R_i \\
& \text{Fitness evaluation for } P_i \\
& \text{Update the non dominated list } Q_i \\
& \text{Evaluation and select best N solution from } P_i \\
& \text{if } i = i + 1 \\
& \text{While stopping criterion}
\end{align*}
\]

IV. EXPERIMENTATION AND NUMERICAL RESULTS

In this section, we present the results of a set of experiments designed to assess the performance of our proposed GA approach. Each instance is solved with 30 independent runs. The algorithm stops after a maximum number of iterations set to 1000 iterations or when no improvement is performing on the objective functions after 500 iterations. The algorithm has been evaluated on a large and realistic highway LA belonging to a big Tunisian telephony operator.

The GA algorithm parameters are listed in Table I. These parameters were selected after experimentations.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Binary code</td>
</tr>
<tr>
<td>Substring length</td>
<td>(W + M)</td>
</tr>
<tr>
<td>Crossover rate (p_c)</td>
<td>0.1</td>
</tr>
<tr>
<td>Mutation rate (p_m)</td>
<td>0.02</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>1000</td>
</tr>
<tr>
<td>Population size</td>
<td>30</td>
</tr>
</tbody>
</table>

To test the proposed GA, we developed a set of experimentation that varies the number of cells, BTS and BSC in the network. For each instance, we report the average of the following measures: The average CPU time, size of the non dominated set \(|P_{ND}|\) and the best values of our two objective functions \(Z_1\) and \(Z_2\) for the GA.

Based on experimental results reported in table II, we can notice that the CPU time is proportional to the problem size. As the number of cells, BTS and BSC increase in the network, the running of our optimizer becomes longer. Similarly, the amount of LU and connection link cost increases by increasing the size of our network in the area of interest.

In all the problem instances, the paging load didn’t exceed the amount of paging threshold set to (20%, 25%) for respective \(LA_1\) and \(LA_2\). In addition, the percentage of paging varies through instances which can be justified by a random and diversified procedure to build the first population in our adopted algorithm.

Compared to the VEPSO algorithm [9], the GA gives better results as shown in table III. The GA outperforms the VEPSO for almost all our solution sets except for the antennas link cost and the paging load percentage where we can notice a very minor difference. In addition, the CPU running time for the GA is slightly longer comparing to the VEPSO. This can be explained by the time consuming of the repairing process that allows to diversify the solutions and thus to escape from the local optima.

Figures 4 and 5 depict the LA’s border and BTS to BSC assignment of the best solution from the set of efficient sets, for both GA and VEPSO algorithm. We can see in figure 4 that there is a major similarity between LA’s border representation for both GA and VEPSO algorithm. The new LA border
TABLE III
COMPARISON BETWEEN THE GA AND VEPSO ALGORITHM TO SOLVE A BOLAP WITH 138 CELLS

<table>
<thead>
<tr>
<th>Solution sets</th>
<th>GA</th>
<th>VEPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution time (sec)</td>
<td>80.06</td>
<td>75.1</td>
</tr>
<tr>
<td>Number of efficient solutions (scenarios)</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Best objective function $Z_1$ (LU)</td>
<td>28 605</td>
<td>28 873</td>
</tr>
<tr>
<td>Best objective function $Z_2$ (Distance)</td>
<td>1.027E8</td>
<td>1.024E8</td>
</tr>
<tr>
<td>Paging/LA (%)</td>
<td>(16,25)</td>
<td>(15,28)</td>
</tr>
</tbody>
</table>

includes $BTS_1$2 in $LA_1$. Where, in figure 5, we can notice a better distribution of the different BSC through the network with the GA. This solution is comparable with the telephony operator’s assignment (see [9]).

V. CONCLUSION

In this paper, we proposed a new formulation of the BOLAP problem. We treated this latter as a two–level optimization problem due its two–dimensional nature. We have investigated the use of bi–objective genetic algorithm in order to obtain a population of pareto solutions spread on the pareto frontier. To evaluate the proposed algorithm, a set of 25 different problem instances were performed using a large highway LA belonging to a big telephony operator. For all the instances, the proposed algorithm yielded nondominated solution that achieves a good LU amount with a feasible BTS-BSC assignment cost and paging load percentage.

It was observed that the VEPSO is not promising compared to the GA. The GA outperformed the VEPSO in almost all the solution sets. Nevertheless, additional tests should be carried out in future work to support our results.

REFERENCES

<table>
<thead>
<tr>
<th>Ph.</th>
<th>Cells</th>
<th>BTS</th>
<th>BSC</th>
<th>CPU(s)</th>
<th>$P_{\text{ND}}$</th>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>Paging/LA (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>9.813</td>
<td>3</td>
<td>2296</td>
<td>2.14E7</td>
<td>(11,10)</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>6</td>
<td>2</td>
<td>17.127</td>
<td>3</td>
<td>7831</td>
<td>3.26E7</td>
<td>(09,10)</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>8</td>
<td>2</td>
<td>22.343</td>
<td>2</td>
<td>10476</td>
<td>4.30E7</td>
<td>(12,10)</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>9</td>
<td>2</td>
<td>24.35</td>
<td>3</td>
<td>11198</td>
<td>5.40E7</td>
<td>(12,10)</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>11</td>
<td>2</td>
<td>26.58</td>
<td>2</td>
<td>13062</td>
<td>6.50E7</td>
<td>(12,11)</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>13</td>
<td>2</td>
<td>38.626</td>
<td>2</td>
<td>14345</td>
<td>7.63E7</td>
<td>(11,14)</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>14</td>
<td>2</td>
<td>45.054</td>
<td>4</td>
<td>15518</td>
<td>8.71E7</td>
<td>(11,15)</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>16</td>
<td>2</td>
<td>52.524</td>
<td>3</td>
<td>17620</td>
<td>9.20E7</td>
<td>(12,13)</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>18</td>
<td>2</td>
<td>57.432</td>
<td>5</td>
<td>18070</td>
<td>9.81E7</td>
<td>(12,14)</td>
</tr>
<tr>
<td>10</td>
<td>55</td>
<td>20</td>
<td>2</td>
<td>60.28</td>
<td>4</td>
<td>18870</td>
<td>10.53E7</td>
<td>(13,11)</td>
</tr>
<tr>
<td>11</td>
<td>60</td>
<td>22</td>
<td>2</td>
<td>60.75</td>
<td>3</td>
<td>19084</td>
<td>10.81E7</td>
<td>(13,12)</td>
</tr>
<tr>
<td>12</td>
<td>65</td>
<td>24</td>
<td>2</td>
<td>61.84</td>
<td>6</td>
<td>19723</td>
<td>11.75E7</td>
<td>(14,15)</td>
</tr>
<tr>
<td>13</td>
<td>70</td>
<td>26</td>
<td>2</td>
<td>62.087</td>
<td>7</td>
<td>20519</td>
<td>12.49E7</td>
<td>(15,10)</td>
</tr>
<tr>
<td>14</td>
<td>75</td>
<td>29</td>
<td>2</td>
<td>63.56</td>
<td>7</td>
<td>21008</td>
<td>13.36E7</td>
<td>(14,16)</td>
</tr>
<tr>
<td>15</td>
<td>80</td>
<td>31</td>
<td>2</td>
<td>65.13</td>
<td>5</td>
<td>21971</td>
<td>15.11E7</td>
<td>(14,16)</td>
</tr>
<tr>
<td>16</td>
<td>85</td>
<td>33</td>
<td>2</td>
<td>65.87</td>
<td>6</td>
<td>22441</td>
<td>16.46E7</td>
<td>(15,14)</td>
</tr>
<tr>
<td>17</td>
<td>90</td>
<td>34</td>
<td>2</td>
<td>66.46</td>
<td>8</td>
<td>22964</td>
<td>18.58E7</td>
<td>(14,17)</td>
</tr>
<tr>
<td>18</td>
<td>95</td>
<td>35</td>
<td>2</td>
<td>66.94</td>
<td>5</td>
<td>23104</td>
<td>19.21E7</td>
<td>(15,16)</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>37</td>
<td>3</td>
<td>67.08</td>
<td>8</td>
<td>23510</td>
<td>43.88E7</td>
<td>(15,18)</td>
</tr>
<tr>
<td>20</td>
<td>110</td>
<td>41</td>
<td>3</td>
<td>69.58</td>
<td>8</td>
<td>25162</td>
<td>55.42E7</td>
<td>(16,19)</td>
</tr>
<tr>
<td>21</td>
<td>115</td>
<td>44</td>
<td>3</td>
<td>70.37</td>
<td>6</td>
<td>25742</td>
<td>60.87E7</td>
<td>(14,21)</td>
</tr>
<tr>
<td>22</td>
<td>120</td>
<td>45</td>
<td>3</td>
<td>70.76</td>
<td>8</td>
<td>26270</td>
<td>75.28E7</td>
<td>(15,20)</td>
</tr>
<tr>
<td>23</td>
<td>125</td>
<td>48</td>
<td>3</td>
<td>71.18</td>
<td>8</td>
<td>26865</td>
<td>88.67E7</td>
<td>(15,20)</td>
</tr>
<tr>
<td>24</td>
<td>130</td>
<td>49</td>
<td>3</td>
<td>71.84</td>
<td>9</td>
<td>27076</td>
<td>96.53E7</td>
<td>(16,20)</td>
</tr>
<tr>
<td>25</td>
<td>138</td>
<td>53</td>
<td>3</td>
<td>80.06</td>
<td>9</td>
<td>28605</td>
<td>1.027E8</td>
<td>(16,25)</td>
</tr>
</tbody>
</table>