A bi-objective location area planning for wireless phone network

Saoussen Krichen
LARODEC Laboratory,
Institut Supérieur de Gestion,
University of Tunis,
41, Liberty Street, 2000,
Bouchoucha-Bardo, Tunisia
and
Faculty of Law, Economics and Management,
University of Jendouba,
UMA Avenue, 8189, Jendouba, Tunisia
Fax : +216-71-588-487
E-mail: Saoussen.krichen@isg.rnu.tn

Ons Abdelkhalek*
LARODEC Laboratory,
Computer Science Applied to Management Department,
Institut Supérieur de Gestion,
University of Tunis,
41, Liberty Street, 2000,
Bouchoucha-Bardo, Tunisia
E-mail: abdelkhalek.ons@gmail.com
E-mail: abdelkha@uvic.ca
*Corresponding author

Adel Guitouni
Decision Support Systems Section,
Defence Research and Development Canada – Valcartier,
2459 Pie-XI Blvd North, Val-Bélair, Québec G3J 1X5, Canada
and
Peter B. Gustavson School of Business,
University of Victoria,
P.O. Box 1700, Victoria BC V8W 2Y2, Canada
Fax: (250) 721-6067
E-mail: Adel.Guitouni@drdc-rddc.gc.ca
E-mail: adelg@uvic.ca

Abstract: In this paper, we propose an extension of the classical location area planning (LAP) problem in the wireless telephony system including additional objectives and constraints. The bi-objective location area planning (BOLAP) problem considers the minimisation of both distance costs between base
transceiver stations and the base station controllers. A vector evaluated particle swarm optimisation (VEPSO) is then implemented to solve the bi-objective model. This heuristic operates in term of two swarms, each corresponding to one of the two objectives, and exchanging their best experience. A comparison of the proposed algorithm is performed with the statistical method being used by a major wireless telephone company. The empirical results show that the proposed approach dominates the operator’s method on one hand, and that the BOLAP is an effective generalisation of the classical LAP model on the other hand.

**Keywords:** location area planning; LAP; vector evaluated particle swarm optimisation; VEPSO; bi-objective optimisation problem; wireless telephony network; paging; handoff.


**Biographical notes:** Saoussen Krichen is Associate Professor in the Faculty of Law, Economics and Management of Jendouba, Tunisia and member of the LARODEC Laboratory, Tunisia. She holds a PhD in Quantitative Methods. She works on multi-objective optimisation, game theory, coalition formation and supply chain management. She has published papers in *EJOR, Sequential Analysis, CMS, Computers and Industrial Engineering, Applied Mathematical Modeling, IJOR* and other journals of Springer and Elsevier.

Ons Abdelkhalek is a PhD student at the Higher Institute of Management in Tunisia and at the University of Victoria in Business Department. She graduated with a Master in Mathematical Modelling with honours in 2009. She works on multi-objective optimisation, telecommunication network management and integrating systems. She is a member of the LARODEC Laboratory, Tunisia. Her thesis is specialised in network management in next generation network. She has published a paper in IEEE Xplore.

Adel Guitouni is a Defence Scientist with Defence R&D Canada since 1998. He holds a PhD in Operations and Decision Support as well as an MSc in Electrical Engineering from Université Laval. He is a Lead Scientist in command and control. He has contributed to developing several command and control-related concepts, solutions and technologies. His research interests include automation of the planning and scheduling, net-enabled dynamic resource management and supply chain management, classification and machine learning, multiple criteria decision analysis, multi-objective optimisation, collaborative decision making and decision support systems. He is an Adjunct Professor with the University of Victoria and Université Laval where he has guided many PhD, MSc and MBA students. He has published papers in *Journal of Applied Mathematical Modelling, European Journal of Operational Research, Logistics and Transportation, Journal of Advances in Information Fusion, INFOR* and *Military Operations Research Journal.*
1 Introduction

The increasing number in mobile telephony requires the development of wireless network to expand network capacity in order to redistribute the resources before any quality of service (QoS) degradations. One of the most common methods to increase the capacity of the network is to reduce the size of cells. However, as the size of cells decreases, the number of cells and the number of handover (HO) (or handoff) increase. The current location area information about a mobile phone is maintained by a registration procedure, namely location update (LU). Mobile phone signals its presence upon entering a new LA. A paging process is therefore triggered to search for the mobile station on the arrival of a mobile-terminated call. To reduce the cost of the signalling update, the most common method is to cluster cells into areas. Each group of cells is called LA. Location area planning (LAP) is concerned by grouping the network cells into LAs while finding a good trade-off between paging and LU signalling. Therefore, LA must contain an optimal set of cells in order to minimise simultaneously the LU and paging costs. In the other hand, the paging load (PL) in a LA increases when the distance between BTSs and BSCs gets bigger, and LU load decreases. Therefore, the distance between base transceiver station (BTS) and the base station controller (BSC) affects the allocation of cells to LAs.

The problem of LAP was introduced by Markoulidakis and Sykas (1993). The proposed model is based on the trade-off between the amount of signalling generated due to paging and LU. The authors used real data from an area of Helsinki to validate their model. As the LAP problem is NP-complete (Garey and Johnson, 1988), the use of approximative methods becomes relevant. Therefore, many heuristics were proposed to solve the LAP: tabu search (TS) algorithms (Demestichas et al., 2000b), greedy search (GS) algorithms (Plehn, 1995), simulated annealing (SA) (Demirkol et al., 2004; Demestichas et al., 1999), genetic algorithms (GA) (Wang et al., 1998; Chan et al., 2002; Gondim, 1996 cited by Vroblefski and Brown, 2006), and grouping GAs (GGA) (Wang et al., 1998; Vroblefski and Brown, 2006). Nevertheless, all these heuristics have been proposed to solve only the uni-objective LAP model.

Multiple decision-making problems involve more than one objective; multi-objective optimisation models have been extensively studied and applied to a wide spectrum of real world problems (Jajimoggala et al., 2010; Parthiban et al., 2009; Diabat et al., 2009). The aim of this paper is then twofold. First, this paper shows that including a second objective function with other constraints to the LAP problem should lead to better results. Yet, to our knowledge the bi-objective LAP model has not been proposed before. The bi-objective location area planning (BOLAP) includes entire subnetwork of GSM system architecture. Second, we propose a vector evaluated particle swarm optimisation (VEPSO)-based meta-heuristic to solve the BOLAP. We present empirical results from a real case study in a GSM subnetwork including 138 cells. These results provide evidence that the proposed approach dominated solutions adopted by a big Tunisian telephony operator.

The rest of the paper is organised as follows. Section 2 states the LAP problem. Section 3 reviews the related literature. The BOLAP problem is detailed in Section 4. The VEPSO algorithm is presented in Section 5. In Section 6, we present the experimental design, results, and interpretations. This paper is concluded by discussion limitations and research perspectives.
2 Problem description

Modelling and simulation approach is commonly used by telephone operators engineers to address the LAP problem. The approach considered by the operator for this study, consists of an iterative redistribution of cells in LAs by considering indicators like QoS or key performers identifiers (KPI) [e.g., PL and paging retry (PR)]. Figure 1 presents the overall method used by the operator being considered in this paper. The division of a LA or the reallocation of cells aims at reducing traffic for paging channels (PCH) [freeing those for reception and the emission of calls (e.g., AGCH)] in order to redistribute the resources before any QoS degradation. We note that PCH represents 60% of the common control channels (CCCH) load.

Figure 1 The operator LAP model
To avoid extreme situations (lack of resources or overload in the network) the operator executes cutting procedure as soon as the paging reaches 50% of PL rate. There is no available data to approximate the registration load calculation. The required data is based on mobile flow rate between each pair of cells, or after call completion. In other words, the HO attempts could be closely correlated with LU. If the PL exceeds 50% of the capacity load, a process of LA redefinition is therefore executed by the operator. Based on a set of QoS indicators, the base station system (BSS) team proposes a set of possible reconfiguration options of the LA. For each option, they perform a set of experiments and testing to find the balance between the LU and PL based on different usage scenarios. If the retained option satisfies all constraints (in particular minimisation of paging and HO) it will then be further studied by the radio engineering team. The most suitable option has to satisfy all the engineering rules and constraints (e.g., geographical constraints, paging, handoff, LU). Afterward, a lengthy process of monitoring and modification will be executed in order to implement the new LA architecture.

Figure 2  Location area design (see online version for colours)

Since the option generation process takes on average between two and three days, application of optimisation techniques and decision support system concepts should allow to speed up the process, to improve the accuracy of decision making and provide a diversified set of options. In this paper, we present an effective decision support approach to LAP problem based on multi-objective optimisation techniques. We propose to generalise the original LAP problem by considering an additional objective function which minimises the BTS-BSC distance cost. Therefore, the proposed bi-objective LAP problem considers the two following objective functions: the minimisation of both the LU and the distance cost. Moreover, the BOLAP must also take into consideration both
proximity constraints of cells, such as cells of the same BTS must be in the same LA, and the capacity constraints, such as the allocation of the traffic and the signalling channels to the correct transceiver (TRX), traffic and the number of links. The aim of BOLAP is to cluster the cells into two (disjoint) LAs: \{LA_1, LA_2\} such that the PL in all LAs does not exceed fixed thresholds while triggering minimal LU load between the different LAs (approximated with HO), and minimal distance cost between BTSs to BSCs and BTSs to LAs (see Figure 2). The minimisation of these two objective functions should improve the quality of signal transmission by minimising LU costs and allow a best clustering of cells into LAs by including BSC frontiers by minimising BTS-BSC distance costs.

3 Review of related work

Markoulidakis and Sykas (1993) introduced the first single objective mathematical model based on the trade-off between signalling costs and paging. As the LAP problem is NP-complete (Garey and Johnson, 1988), the use of approximate methods becomes relevant. Over the last years, many heuristics have been proposed to efficiently obtain good solutions to this problem.

Plehn (1995) proposed later a two phase’s algorithm: a merge and exchange weighted GS algorithm to design of LAs for a German operator (GSM-network D1 operated by the DeTeMobil). The algorithm iteratively tries to merge two LAs using the maximum profit to minimise the number of LUs. They proved that their algorithm performs significantly better than the previously used methods. They reduced the LA design cost with a reasonable paging threshold by 53% of its original value in the D1 network.

Varsamopoulos and Gupta (2004) proposed to dynamically adjust the coverage of LAs based on the mobility patterns of mobile terminals (MTs). They proposed an extension to the personal communication services (PCS) networks by dynamically overlap LAs and showed, by this method, that they minimise considerably the LU by mobiles and, through simulation, that the new scheme is able to adapt to the locality in mobility and call patterns. Researchers in location planning agree in general that the road map and the traffic flows have significant impacts on the mobility patterns.

Bhattacharjee et al. (2000) examined two intelligent paging approaches called sequential intelligent paging and parallel-to sequential intelligent paging where, in paging process, one cell is polled at the time. They used an occupancy probability vector to determine the cells that will be paged. The new schemes outperform the conventional paging schemes in term of reducing the PL signalling at the cost of some degradation in the average delay performance. Then, Bhattacharjee et al. (2003) propose a greedy heuristic algorithm that constructs LA by assigning cells in an iterative way. They proved that this new approach outperformed some previously proposed algorithms.

Demestichas et al. (1999) defined and solved a different version of the LAP problem that includes, in the objective function, the minimisation of the paging cost, with an imposed set of constraints adopted by the operators. They proposed to limit the expected paging traffic by a threshold and not a set of constraints. They designed three efficient solutions based on SA, TS, and GA. Based on their experimentations, the TS show an improved performance depending on the mobility levels. As mobility levels increase, the TS algorithm shows an even better performance. Then, Demestichas et al. (2000a), added mobility conditions to the LAP problem and focused on real-time algorithms to design area’s border. Bejerano and Cidon (2001) proposed to dynamically adjust the coverage of
LAs by predicting the speed of MTs on highways. As the number of mobile users increase, they reduced in the size of cells thus increase in the number of cells in the LA.

Escalle et al. (2002) proposed the concept of big-LAs (BLA) that incorporates three neighbouring LAs. A MT stores the identification of a BLA in its local memory and LUs performance each time MTs leave the BLA. Paging is carried out as follows: first, where the LA of most recent interaction is paged, second, if the MT is not located, all remaining LAs in the BLA are paged. After comparing this new algorithm with the classical one, they showed that the new one outperform the others location management strategies in term of reducing the LU and paging costs and can easily be implemented on any existing cellular network.

Chan et al. (2002) adopted the GA with an efficient searching technique to solve the ‘hard handoff minimisation problem’, reducing the weighted sum of three objective functions: the total weighted-distance connection cost from BSs to BSCs, the total weighted-distance connection cost from BSCs to personal communication services exchanges (PCXs) and the hard handoff cost. This problem is quite similar to the LAP problem that takes as objective function the minimisation of the LU. The main objective in Chan et al.’s (2002) model is to find the optimal assignment of BSs with BSCs and BSCs to PCXs. They concluded that the GA leads to better solutions than SA to reduce the three objective functions. Also, the multiple objective approaches are more suitable than the single objective formulation as it generates a set of efficient solutions, so the planner can select his preferred solution.

Bejerano et al. (2003) developed a clustering algorithm for the LAP problem. The algorithm minimises LU and paging costs by fixing an initial seed node (cell) and then growing regions around it. The solution is then improved by doing greedy exchanges in the borders of LAs neighbouring. They showed that the proposed algorithm outperforms the other greedy methods. The trade-off between paging and LU costs can be mapped to a graph partitioning problem in which nodes represent cells and the arcs correspond to the traffic passing between cells.

Demirkol et al. (2004) proposed a SA algorithm to solve both the LAP and cell-to-switches assignment (CSA) problem in cellular networks. The proposed algorithm outperforms GS random generation methods.

Vroblefski and Brown (2006) developed a grouping GA for the registration area planning (RAP) problem (i.e., LAP problem). The purpose is to group cells into LAs in order to minimise the LU costs while fixing the paging thresholds and preset constraints to reflect the determination that two cells are assigned, or not, in the same LA. The main objective of grouping in GA is to deal with drawbacks and enable the evolution of high quality solutions. In addition, the GGA allows supporting any number of constraints by including penalties to the fitness function and reducing redundant search of the solution space. The proposed algorithm is proved to be robust and give good quality solutions in an optimal running time.

Shyu et al. (2006) developed an ant colony optimisation (ACO) algorithm to minimise the hybrid costs associated with handoffs and cabling costs (i.e., CSA problem) to solve the cell assignment problem. They demonstrate that the proposed ACO algorithm is an effective and competitive approach compared with most existing heuristics or meta-heuristics with respect to solution quality and running time.

Paik and Samit (2006) proposed a hybrid SA and GS heuristic for finding optimal location and paging area (PA) configuration for mobile networks in order to minimise the LU. They used a two-layered zone-based location registration and paging scheme, in
which the cost of LU and paging signalling traffic are reduced. They considered that the two-layered scheme allows the LA to be divided into multiple PAs, whereas in the single layered scheme the PA of a mobile station is always equal to the LA of the mobile station.

James et al. (2007) developed a hybrid GGA (HGG) for the LAP problem. This hybridisation is accomplished by adding an improvement TS method to the GGA (Vrobleński and Brown, 2006) to get an improved grouping applied to the initial population as well as to the new solution created by the crossover. Compared with the simple GA, this method gave better solution quality for a large number of cells. GAs have been adopted to solve LAP problem by Gondim (1996, cited by Vrobleński and Brown, 2006), Wang et al. (1998) and Demestichas et al. (1999).

Table 1 summarises the contributions reviewed. Most of the reviewed literature addresses the classic single objective optimisation LAP problems to minimise the cost of the LU and paging messages. However, only Chan et al. (2002) introduced the multi-objective optimisation problem where it is usually used to reduce the weighted sum optimisation or multi-layered optimisation. We think that it is more appropriate to adopt the multi-objective structure of the problem in order to generate efficient or potentially efficient solutions in order to provide to the decision maker all flexibility and alternatives. Moreover, the particle swarm optimisation (PSO) has never being used to solve the LAP. In this paper, we are concerned with minimising both the LU and the distance cost of BTS-BSC.

Table 1 Summary of the literature review on LAP problems

<table>
<thead>
<tr>
<th>Solution procedures</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>Demestichas et al. (1999)</td>
</tr>
<tr>
<td></td>
<td>Demestichas et al. (2000b)</td>
</tr>
<tr>
<td>SA</td>
<td>Demirkol et al. (2004)</td>
</tr>
<tr>
<td></td>
<td>Demestichas et al. (1999)</td>
</tr>
<tr>
<td></td>
<td>Paik and Samit (2006)</td>
</tr>
<tr>
<td></td>
<td>Demestichas et al. (2000b)</td>
</tr>
<tr>
<td>GS</td>
<td>Plehn (1995)</td>
</tr>
<tr>
<td></td>
<td>Bhattacharjee et al. (2003)</td>
</tr>
<tr>
<td>Clustering and greedy</td>
<td>Bejerano et al. (2003)</td>
</tr>
<tr>
<td>GA</td>
<td>Wang et al. (1998)</td>
</tr>
<tr>
<td></td>
<td>Gondim (1996)</td>
</tr>
<tr>
<td></td>
<td>Demestichas et al. (1999)</td>
</tr>
<tr>
<td></td>
<td>Chan et al. (2002)</td>
</tr>
<tr>
<td></td>
<td>Demestichas et al. (2000b)</td>
</tr>
<tr>
<td>Grouping GA</td>
<td>Gondim (1996)</td>
</tr>
<tr>
<td></td>
<td>Wang et al. (1998)</td>
</tr>
<tr>
<td></td>
<td>Vrobleński and Brown (2006)</td>
</tr>
<tr>
<td>Ant colony</td>
<td>Shyu et al. (2006)</td>
</tr>
<tr>
<td>Hybrid grouping GA</td>
<td>James et al. (2007)</td>
</tr>
<tr>
<td>Dynamic adjustment</td>
<td>Bejerano and Cidon (2001)</td>
</tr>
<tr>
<td></td>
<td>Varsamopoulos and Gupta (2004)</td>
</tr>
</tbody>
</table>
4 The BOLAP problem

The assignment process of cells to LAs should achieve ‘the best’ trade-off between the level of LU and paging on one hand, and the distance cost between BTSs and BSCs on the other hand. A high mobile concentration or a high mobility between visitor location registers (VLR) in small LAs generates high LU traffic. Thus, by increasing the distance cost between BTSs and BSCs the PL in the LAs increase and thus reduces the LU. So, we can conclude that these two objective functions are conflicting.

We, therefore, propose a new bi-objective formulation of the LAP problem that takes into consideration the two main objective functions: the distance between BTSs and BSCs and the LU between LAs while fixing a paging threshold. The proposed BOLAP formulation is a generalisation of the LAP problem (Vroblefski and Brown, 2006) with two objective functions. The proposed model also includes additional constraints which were not considered in Vroblefski and Brown’s formulation like proximity constraints. These constraints are important to optimise the transmission between channels, and thus prevent lack of resources and the degradation of the signalling quality. We consider that the BOLAP in this paper should lead to better management of networks. The BOLAP problem will also generate the assignment solution of the entire BSS subsystem in the GSM networks architecture.

4.1 Notations

Before presenting our BOLAP problem, we need to define the following notations reported in Table 2.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of cells</td>
</tr>
<tr>
<td>M</td>
<td>Number of LAs</td>
</tr>
<tr>
<td>Q</td>
<td>Number of BTSs</td>
</tr>
<tr>
<td>W</td>
<td>Number of BSCs</td>
</tr>
<tr>
<td>Ci</td>
<td>Set of cells i = 1, 2,…, N</td>
</tr>
<tr>
<td>ci</td>
<td>Call traffic of cell i at busy hour (in Erlang)</td>
</tr>
<tr>
<td>rj</td>
<td>Number of TRXs for each cell i</td>
</tr>
</tbody>
</table>
| his     | HO rate from i
|         | th cell to s
|         | th cell |
| wis     | Assignment of
|         | i
|         | th cell and s
|         | th cell in the same or different LAs |
| xir     | Assignment of the i
|         | th cell to the r
|         | th LA |
| yjk      | Assignment of the j
|         | th BTS to the k
|         | th BSC and the j
|         | th BTS to the r
|         | th LA |
| aij     | Assignment of the i
|         | th cell to the j
|         | th BTS |
| vlr     | Presets: fixing the assignment of the i
|         | th cell to the r
|         | th LA |
| λi      | The average number of incoming calls to cell i per unit time |
| PBr     | The paging threshold per LAi |
| djk     | Distance between the j
|         | th BTS and the k
|         | th BSC |
| NkBSC   | Maximum number of links of the k
|         | th BSC useful for the connection with the BTSs |
| CkBSC   | Call traffic capacity of the k
|         | th BSC (in Erlang) |
| RkBSC   | TRX capacity constraint of the k
|         | th BSC |
4.2 Mathematical formulation

The mathematical formulation of the BOLAP problem is stated through equations (1) to (13):

\[
\text{Min } Z_1(X) = \sum_{i=1}^{N} \sum_{s=1}^{N} h_{is} w_{is}
\]

\[
\text{Min } Z_2(X) = \sum_{k=1}^{W} \sum_{j=1}^{Z} \sum_{r=1}^{M} d_{jk} y_{jk}
\]

s.t.

\[
\sum_{r=1}^{M} x_{ir} = 1 \quad i = 1, 2, \ldots, N,
\]

\[
\sum_{r=1}^{M} \sum_{k=1}^{W} y_{jk} = 1 \quad j = 1, 2, \ldots, Z,
\]

\[
w_{is} = \sum_{r=1}^{M} y_{is} x_{ir} \quad i, s = 1, 2, \ldots, N,
\]

\[
\sum_{k=1}^{W} y_{jk} = \sum_{i=1}^{N} a_{i} a_{jr} \quad j = 1, 2, \ldots, Z, 
\]

\[
x_{ir} = v_{ir} \quad \forall v_{ir} = 0 \text{ or } 1
\]

\[
\sum_{j=1}^{Z} a_{ij} a_{jr} - \sum_{r=1}^{M} x_{ir} x_{sr} \leq 0 \quad i, s = 1, 2, \ldots, N,
\]

\[
PB_{r} = \sum_{i=1}^{N} x_{ir} \lambda_{r} \geq 0 \quad r = 1, 2, \ldots, M,
\]

\[
\sum_{r=1}^{M} \sum_{j=1}^{Z} \sum_{i=1}^{N} a_{ij} y_{jk} c_{ij} \leq C_{k}^{BSC} \quad k = 1, 2, \ldots, W,
\]

\[
\sum_{r=1}^{M} \sum_{j=1}^{Z} \sum_{i=1}^{N} a_{ij} y_{jk} r_{i} \leq R_{k}^{BSC} \quad k = 1, 2, \ldots, W,
\]

\[
\sum_{r=1}^{M} \sum_{j=1}^{Z} y_{jk} \leq N_{k}^{BSC} \quad k = 1, 2, \ldots, W,
\]

\[
x_{ir} \in \{0, 1\} \quad i = 1, 2, \ldots, N, 
\]

where \(x_{ir} = 1\) if cell \(C_i\) is in \(LA_r\), and 0 otherwise. \(w_{is} = 0\) if cells \(C_i\) and \(C_s\) are in the same \(LA\), 1 otherwise. \(y_{jk} = 1\) if \(BTS_j\) is assigned to \(BSC_k\) and to \(LA_r\), 0 otherwise. \(x_{ir} = 1\)
if \( BTS_j \) is assigned to \( LA_r \), 0 otherwise. In objective functions \( Z_1 \) and \( Z_2 \), the solution
\[
X = \{x_{ir}, y_{jkr}\}.
\]

4.2.1 Objective functions

- The first objective function \( (Z_1) \) minimises the total registration signalling. We approximate this aggregate mobile flow behaviour to the HO rate between two cells \( C_i \) and \( C_s \) to minimise the total LU cost. A LU must be performed when a mobile crosses a border between two LAs. It provides the estimated total number of mobiles crossing between LAs borders per time unit.

- The second objective function \( (Z_2) \) minimises the total distance cost linking BTSs to BSCs and BTSs to LAs in order to improve the quality signal transmission when BTS is assigned to a particular LA.

4.2.2 Constraints

Equation (3) enforce that each cell should be assigned to only one LA. Similarly, each BTS must be assigned to only one BSC and one LA is constrained by equation (4). Equation (5) ensure that \( w_{ij} = 0 \) if cells \( C_i \) and \( C_s \) are in the same LA and 1 otherwise. We assume in equation (6) that the number of \( BTS_j \) assigned to \( LA_r \) is equal to the total number of BTSs in the LA. In some situations, the network designer or the decision maker would like to pre-assign cells \( C_i \) and \( C_s \) to \( LA \), which is captured by equation (7). Equation (8) are the proximity constraints which ensure that if cells \( C_i \) and \( C_s \) belongs to the same \( BTS_j \), then they must be assigned to the same \( LA_r \). The paging bounds are satisfied through equation (9). Equations (10), (11) and (12) represent the call traffic capacity, the TRX capacity and links limitations for each BSC. The decision variables \( x_{ir} \) are binary as captured in constraint (13).

Equations (1), (3), (5), (7), (9) and (13) are already presented in the classic LAP model proposed by Vrobleski and Brown (2006), where Demirkol et al. (2004) used equations (10) to (12) to solve a model that integrate LAP and CSA problem. The CSA is similar to LAP problem. It is about grouping cells into LAs during switch level planning of a cellular wireless network. It is NP-hard problem. Equations (2), (4), (6) and (8) are additional constraints we introduced to generalise the LAP: BOLAP problem. By reducing the BOLAP to the LAP problem, we can assume that it’s also NP-hard and thus cannot be solved optimally.

The efficient solution is a combination of LA-cell \( (X \) matrix) topology and BTS-BSC-LA topology \( (Y \) matrix) which ‘best’ minimises the LU and BTS-BSC cost subject to the previous mentioned constraints. ‘Best’ in this case refers to Pareto-optimality. Yapicioglu et al. (2007) defined the Pareto preference or dominance preference relationship as “a solution, \( X_{(2)} \), is dominated by solution, \( X_{(1)} \), if \( X_{(1)} \) is not worse than \( X_{(2)} \) according to all objectives \( (K) \), and for at least one of the objectives, \( X_{(1)} \) is strictly better than \( X_{(2)} \),” represented as follows:

\[
Z_{(k)} (X_{(1)}) \leq Z_{(k)} (X_{(2)}) \quad \text{for} \quad k = 1, 2, 3, \ldots, K \tag{14}
\]

\[
Z_{(k)} (X_{(1)}) < Z_{(k)} (X_{(2)}) \quad \text{for some} \ k \tag{15}
\]
Therefore, solving the BOLAP problem should generally lead to a set of efficient – exact method – or potentially efficient solutions – approximate method – (Shyu et al., 2006). As mentioned before, previous contributions to LAP have either adopted a single objective optimisation model or transformed the multi-objective problem into a single objective problem using a weighted sum. In our opinion, could be risky approach for many reasons. In particular, aggregation of conflicting and non-commensurable functions should be done carefully. Moreover, the optimal solution found might not be the best compromise for the decision maker. In case of non-linear objective functions, the solution might not even be efficient. Therefore, this paper adopts the Pareto-optimisation principles. Given that it is an NP-hard problem, we consider evolutionary algorithms for solving the BOLAP problem. We therefore propose in the next section an evolutionary algorithm (VEPSO) that simultaneously Pareto-optimises the two objectives.

5 A VEPSO algorithm for the BOLAP

The algorithm generates the set of potentially efficient solutions for a network topology which includes the assignment of cells to LAs and BTSs to BSCs and BSCs to LAs. For a large number of cells, Hassan et al. (2005) proved that the HGGA performed better than the simple GA. However, he also proved that the PSO algorithm is more effective than GA in most cases. So, PSO is likely to be a good algorithm for dealing effectively with multi-objective optimisation problems (Ongsakul et al., 2011). The VEPSO, introduced in Parsopoulos et al. (2004) and Omkar et al. (2008), is a multi-swarm variant of PSO, inspired from the vector evaluated genetic algorithm (VEGA). The basic idea in VEPSO is that each swarm is evaluated using only one of the objective functions of the problem under consideration. The information possesses for this objective function is communicated to the other swarms through the exchange of their best experiences. Thus, exchanging theses information among swarms will converges to Pareto optimal point and then improves the capability of the algorithm to better explore the search space.

VEPSO is very suitable to solve BOLAP problem. VEPSO searches multiple efficient solutions in a large solution space. The key issue with VEPSO is that the fitness of an individual into a population depends on the best particles of a different swarms. A VEPSO algorithm for discrete variables was developed by Omkar et al. (2008) for the multi-objective optimised design of composites. The problem is formulated with multiple objectives for minimising the weight and the total cost of the composite component to achieve a specified strength. It was proven that in the majority of considered cases that the VEPSO algorithm generates superior composite designs. The authors proved also that this approach does not impose any limitation on the number of objectives and constraints allowing any changes in design parameters.

We adopt the VEPSO algorithm in order to solve BOLAP problem. Our approach supports non-linear constraints with discrete design variables. The proposed algorithm minimises both LU and distance costs by growing regions around an initial seed cell to guarantee that all assigned cells are neighbours. We then improve the solution by assigning neighbouring cells by fixing a radius size $R$. Cell $C_i$ and $C_s$ are said to be neighbours if $\text{dist}(i, s) \leq R$ where $R$ is a fixed values of neighbourhood radius and depends on the decision maker needs. Since there are two objective functions to be optimised, two swarms $\{S_1, S_2\}$ of $L$ particle for each swarm are therefore used. $S_i$
evaluates the LU cost function where $S_2$ evaluates the distance cost function. The global best solution of the second swarm $S_2$ is used to calculate the new velocity of the first swarms $S_1$ and vice versa.

5.1 Construction of a particle

A particle consists of a solution represented by a $D$-length that contains the sequence of $N$ cells and $W$ BSC to be assigned. We define a particle $i$ at iteration $t$ as:

$$X^p_{it} = \left\{ \left( x^{p,1}_{it}, x^{p,2}_{it}, \ldots, x^{p,N}_{it} \right), \left( y^{p,1}_{it}, y^{p,2}_{it}, \ldots, y^{p,W}_{it} \right) \right\},$$

$x^{p,j}_{it} = r$ if cell $j$ of particle $i$ is assigned to LA $r$ for the objective $p$, $p = \{1, 2\}$, where $r = \{1, 2, \ldots, M\}$ defines the number of LAs and $j = \{1, 2, \ldots, N\}$ define the number of cells.

$y^{p,l}_{it} = (f, r)$ if BSC $l$ of particle $i$ is assigned to BTS $f$ and to LA $r$ for the objective $p$, $p = \{1, 2\}$, where $f = \{1, 2, \ldots, Z\}$ define the number of BTS and $l = \{1, 2, \ldots, W\}$ define the number of BSC. For example, consider the sequence $X^1_{it} = \{1, 2, 1, 2, 1, 1, 2, 1\}$ for the first objective, where we consider the assignment of three cells, two LAs, two BTSs and two BSCs (see Figure 3).

![Figure 3](image)

**Figure 3** Particle representation for assigning three cells, two BTSs and two BSCs in two LAs

<table>
<thead>
<tr>
<th>LA</th>
<th>Cell$_1$</th>
<th>Cell$_2$</th>
<th>Cell$_3$</th>
<th>BSC$_1$</th>
<th>BSC$_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2, 1</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

We suppose that cells 1 and 3 are assigned to BTS 2, and cell 2 is assigned to BTS 1. By this definition we have $x^{1,1}_{i,t} = 1, x^{1,2}_{i,t} = 2, x^{1,3}_{i,t} = 1$ and $y^{1,1}_{i,t} = (2, 1), y^{1,2}_{i,t} = (1, 2)$. In the following representation, cells 1 and 3 are assigned to LA 1, cell 2 is assigned to LA 2, BTS 2 is assigned to BSC 1 and to LA 1, and BTS 2 is assigned to BSC 2 and to LA 2.

5.2 Initial population

The population is initialised by growing regions around an initial seed cell and then grouping neighbouring cells into LAs while maintaining the paging bound.

5.2.1 Swarm $S_1$

Each initial solution $y^{p}_{it}$ is then evaluated with the fitness function that eliminates infeasible solutions (not satisfy the capacity constraints).

The initialisation operator works as follows:

1. Generate an initial population $x^{p}_{it}$. We choose an initial cell to assign in the new LA (choose randomly between LA$_1$ and LA$_2$).

2. Determine the nearest neighbours of assigned cell and add them to the LA until no more can be added without violating the paging bound.
3 Determine cells that are assigned to the same BTS and add them to the same LA without violating the proximity and paging constraints.

4 Repeat Step 2 until all cells have been assigned.

5 Build the $y_{ij}^p$ from the $x_{ij}^p$ assignment, were BTSs are assigned randomly without respecting BSCs capacity constraints.

6 Repeat Step 5 until all BTSs were assigned.

5.2.2 Swarm $S_2$

Each initial solution $x_{ij}^p$ is evaluated with the fitness function that eliminates infeasible solutions (not satisfy the paging constraints).

The initialisation operator works as follows:

1 Generate an initial population $x_{ij}^p$ where cells are assigned randomly in the new LA.

2 Check the paging and proximity constraints. This operator determines the neighbours of the added cell assigned to the same BTS, then add them to the LA until no more can be added without violating the paging bound (proximity and paging constrains).

3 Repeat Step 2 until all cells were assigned.

4 Build the $y_{ij}^p$ from the $x_{ij}^p$ repaired as follows: We choose a BTS, determine the nearest neighbours of it and assign them to the BSC, until no more can be added without violating BSC capacity constraints (call traffic, TRX, link constraints). We, then, assign BSCs to the same LA where their cells have been assigned.

5 Repeat Step 4 until all BTSs and LAs were assigned.

For these two swarms, the particle vectors will be iteratively modified based on collective experiences in order to improve their solution’s quality for the two objective functions.

During the generation phase of the two initial populations, we consider neighbors of the added cell and assign them to the LA. Particles compound two main parts:

- **Assignment of cells to LAs** ($x_{ij}^p$): It represents the initial population generated by construction, for the first swarm, and randomly generated for the second swarm. Then, paging and the proximity constraints are tested.

- **Assignment of BTSs to BSCs and BTSs to LAs** ($y_{ij}^p$): This second step consists of testing BTSs of the assigned cells, and then checking TRX, traffic and links of BSCs capacity constraints. The same operation is repeated for the second swarm based on the steps detailed above.

5.3 Particle vector modification

At each evolutionary iteration, the particle $P_i$ updates its velocity $V_{(id,i)}^{(Sp)}$ and position $X_{(id,i)}^{(Sp)}$ referring to the personal best experience $P_{(id,i)}^{(Sp)}$ and the swarm’s best experience
also named global best experience. \( d = \{1, \ldots, N, N+1, \ldots, W\} \) defines the length of the particle. The particles velocity and position update equations are as follows (Poli et al., 2007):

\[
V_{(id,t+1)}^{S_p} = K^{S_p} \left\{ \omega V_{(id,t)}^{S_p} + c_1 r_1 \left( P_{(id,t)}^{S_p} - X_{(id,t)}^{S_p} \right) + c_2 r_2 \left( P_{g}^{S_p} - X_{(id,t)}^{S_p} \right) \right\}
\]

(16)

\[
X_{(id,t+1)}^{S_p} = X_{(id,t)}^{S_p} + V_{(id,t+1)}^{S_p}
\]

(17)

where \( p = \{1, 2\} \) and \( q = \{2, 1\} \) simultaneously for \( S_1 \) and \( S_2 \), \( c_1 \) and \( c_2 \) are the acceleration coefficients, \( r_1 \) and \( r_2 \) are random numbers in \( U[0, 1] \). In order to guarantee the convergence of the VEPSO, we use a constriction coefficient \( K \), calculated as follows by Naka et al. (2003):

\[
K = \frac{2}{2-(c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)}} \quad \text{s.t.} \quad (c_1 + c_2) \geq 4
\]

(18)

The inertia weight parameter \( \omega \) is adjusted dynamically during the optimisation. A starting value of \( \omega_{\text{max}} = 1 \) is used for a more global search. Then it is dynamically reduced to \( \omega_{\text{min}} = 0.4 \) as follows:

\[
\omega = \omega_{\text{max}} - \left\{ \left( \omega_{\text{max}} - \omega_{\text{min}} \right) i t \right\} i t
\]

(19)

with \( \omega_{\text{max}} \) the initial weight factor, \( \omega_{\text{min}} \) the final weight factor, \( it \) the current iteration number and \( it_{\text{max}} \) the maximum number of iterations. When the stopping criteria are met, the best experienced position by the entire swarm is reported as the final solution.

### 5.4 Fitness function

The fitness function \( F(X) \) measures to what extent the particle solution \( X \) satisfies the objective of the constrained optimisation problem. Yin et al. (2006) defined two important criteria for evaluating \( F(X) \): first, solution with best objective values, second, solution with highest level of constraints satisfaction. Consequently, we will define a penalty function \( J(X) \) to penalise non-feasibility level of a particle solution. So, we adopt the following fitness function used by Yin et al. (2006):

\[
F(X) = \left( Z(X) + J(X) \right)^{-1}
\]

(20)

where \( Z(X) = (Z_1(X), Z_2(X)) \) are the objective functions of the BOLAP problem [see equations (1) and (2)] and \( J(X) \) is the quantified amount of mismatch if \( X \) is infeasible; otherwise, \( J(X) = 0 \). Therefore, the higher the fitness value of \( X \), the better is the solution. Since constraints (2) to (7) and constraint (13) are always satisfied if our particle representation scheme is adopted (as explained in sub-Sections 5.1 and 5.2), the penalty function is only related to constraints (8) to (12) \( (\text{Yin et al., 2006}) \), and it is given by:
A bi-objective location area planning for wireless phone network

\[ J(X) = \sum_r \max \left( 0, \sum_i l_{ri} x_i - PB_i \right) \]
\[ + \sum_x \max \left( 0, \sum_r \sum_j \sum_i d_{ij} y_{ijk} - C_{ijk}^{BSC} \right) \]
\[ + \sum_k \max \left( 0, \sum_r \sum_j \sum_i d_{ij} y_{ijk} - R_{ik}^{BSC} \right) \]
\[ + \sum_k \max \left( 0, \sum_r \sum_j y_{ijk} - N_k^{BSC} \right) \]

(21)

5.5 Velocity update coefficient

The VEPSO parameters used in the current case are equal to 2 for the acceleration parameters \( c_1 \) and \( c_2 \), a starting value of the inertia weight \( \omega = 1 \) that is dynamically reduced to \( \omega = 0.4 \) [as in equation (19)]. The values of the coefficients \( r_1 \) and \( r_2 \) [equation (16)] are randomly generated between 0 and 1. When a coefficient is multiplied by a velocity, it indicates the probability of each movement to be applied.

As we fixed \( c_1 = c_2 = 2 \), the constriction coefficient \( K \) [equation (18)] will be equal to 1 and then, it will no more affect the velocity update. We fix a constant \( V_{\text{max}} = 4 \) to limit the range of velocity.

5.6 Stopping criteria

The VEPSO algorithm stops after it has run for a preset maximum number of iterations or when the sum of the variation of the percentage ratio of the two objectives is less or equal to \( \epsilon \). It is computed as follows:

\[ \delta_1 = \frac{Z_1(X_{i+1}^*) - Z_1(X_i^*)}{Z_1(X_i^*)} \quad \text{and} \quad \delta_2 = \frac{Z_2(X_{i+1}^*) - Z_2(X_i^*)}{Z_2(X_i^*)} \]

thus, \( \delta_1 + \delta_2 \leq \epsilon \) \hspace{1cm} (22)

The proposed VEPSO algorithm is detailed in what follows.

<table>
<thead>
<tr>
<th>INPUTS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D ): size of the particle, i.e., number of cells and BSCs to be assigned</td>
</tr>
<tr>
<td>( L ): size of the swarm in ( S_1 ) and ( S_2 )</td>
</tr>
<tr>
<td>( \iota_{\text{min}} ): minimum number of iterations</td>
</tr>
<tr>
<td>( \iota_{\text{max}} ): maximum number of iterations</td>
</tr>
<tr>
<td>( c_1 = c_2 = 2 )</td>
</tr>
<tr>
<td>( r_1, r_2 \in [0, 1] )</td>
</tr>
<tr>
<td>( t = 0 )</td>
</tr>
<tr>
<td>( C_i ): seed cell</td>
</tr>
<tr>
<td>( R ): radius of neighbourhood</td>
</tr>
<tr>
<td>( \text{NbrBTS} ): number of BTSs to be assigned</td>
</tr>
<tr>
<td>( \text{NbrLA} ): number of final LAs (equal to 2 in our study case)</td>
</tr>
</tbody>
</table>
OUTPUTS:

\( i_t \): number of iterations at time \( t \)

\( X[i] \): the position of the \( i^{th} \) particle in the population \( \forall i \in \{1, 2, \ldots, L\} \)

\( X[i][d] \): LA number to which the \( d^{th} \) cells or BSC in the \( i^{th} \) particle is assigned; \( \forall i \in \{1, 2, \ldots, L\} \) and \( \forall d \in \{1, 2, \ldots, D\} \)

\( F[i] \): fitness function of the particle \( i \) [equation (20)]

\( V[i] \): velocity of the \( i^{th} \) particle

\( P_{global} \): index to global-best position

\( P_{local}[i] \): index of the local-best position in the \( i^{th} \) particle

VEPSO algorithm:

For each \( \{S_1, S_2\} \)

Initialisation:

/* the initialisation is done according to subsection 5.2. Initial population */

\( \circ \) Step 1

- For \( D \) cells and BSCs, \( d \in \{1, 2, \ldots, D\} \);
  \( X[i][d] \leftarrow \text{Random}(\) from \([1, NbrLA]\) for \( d \in \{1, 2, \ldots, D\}\)
  \( X[i][d] \leftarrow \text{Random}(\) from \(([1, NbrBTS], [1, NbrLA])\) for \( d \in \{N + 1, \ldots, W\}\).
- All particles are checked to satisfy all the constraints.
- \( V[i] \leftarrow \text{Random}(\) in \( U[0, 1]\).
- Evaluate \( F[i] \)
- \( P_{local}[i] \leftarrow X[i], \forall i \in \{1, 2, \ldots, L\} \)
- \( P_{global} \leftarrow \text{Max}(F[1], F[2], \ldots, F[L]) \) /* Get the index of the particle */

Iteration:

\( \circ \) Step 2

Loop \( (d = 1 \text{ to } D) \)

- Generate the inertia weight \( (\omega) \) [equation (19)]
- Update the particle velocity \( F[i][d] \) [equation (16)]
- Update the particle position \( X[i][d] \) [equation (17)]
- After each update check whether the variables of each particle of both the swarms satisfy the constraints.
- Evaluate \( F[i] \) [equations (20) and (21)]

End Loop

\( \circ \) Step 3

Loop \( (i = 1 \text{ to } L) \)
IF \( F[i] \geq F(P_{local}[i]) \) THEN /* identification of the personal best solution */

Loop \((d = 1 \text{ to } D)\)

- \( P_{local}[i] \leftarrow X[i][d] \)
- Next \( d \)

End Loop

End IF

Step 4

Loop \((j = \text{MaxP}(i) \text{ to } L)\)

IF \( F[i] > F(P_{global}) \) THEN /* identification of the global best solution */

- \( P_{global} \leftarrow j \)
- Next \( j \)

End IF

End Loop

Repeat Step 2

End Loop.

Until \((it = it_{\text{max}}) \text{ or } (\delta_1 + \delta_2 \leq \epsilon)\)

6 Computational experiments

The VEPSO algorithm is coded in JAVA because of its modularity and flexibility, using Netbeans as a tool for programming and MySQL to manage the database. All computations are carried out on Pentium IV with 3.20 GHz processor and 512 Mb of RAM. The operating system is Microsoft Windows XP. To better manage the computer memory, we use ‘threads’ programming. Execution time depends on both the size of particles in each swarm and on VEPSO parameters. Experiments are repeated 1,000 times for each parameter set and each instance is solved with 30 independent runs in order to report the average results. The performance of the proposed VEPSO algorithm is after compared with the results of the cellular phone operator.

6.1 Run of the VEPSO algorithm

We consider an existing GSM networks topology with 138 cells, 54 BTSs, 3 BSCs and 1 MSC in one LA. After many set of experimentation, we get VEPSO parameters listed in Table 3.

The VEPSO algorithm is fine tuned by testing different values for the radius of neighbourhood \( R \) and varying seed cell around which the region grows. The results obtained by VEPSO are then compared to the solution of the telephone company’s model. Figure 4 illustrates the run of our VEPSO optimiser’s application with 138 cell configurations.
Table 3  Optimal VEPSO configurations

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive learning rate</td>
<td>$c_1 = 2$</td>
</tr>
<tr>
<td>Social learning rate</td>
<td>$c_2 = 2$</td>
</tr>
<tr>
<td>Inertia factor</td>
<td>$\omega = [1, \ldots, 0.4]$, dynamically reduced from 1 to 0.4 with each iteration</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>$V_{\text{max}} = 4$, control the maximum travel distance during each iteration</td>
</tr>
<tr>
<td>Maximum number of iterations</td>
<td>$I_{\text{max}} = 1,000$</td>
</tr>
<tr>
<td>Number of swarm particles</td>
<td>$N = 30$</td>
</tr>
<tr>
<td>Default radius of neighbourhood/m</td>
<td>$R = 120,000$ m</td>
</tr>
<tr>
<td>Thresholds of paging (%)</td>
<td>$\begin{cases} 20% \text{ LA}_1 \ 25% \text{ LA}_2 \end{cases}$</td>
</tr>
<tr>
<td>Default seed cell</td>
<td>$C_{16}$ from $\text{BTS}_{16}$</td>
</tr>
</tbody>
</table>

Figure 4  138 cell configurations in VEPSO application (see online version for colours)

6.2 Computational results

Table 4 presents the comparison between the results obtained by the VEPSO algorithm (using parameters listed in Table 3) to the telephone company solutions.

The results clearly show that the proposed approach outperforms the operators design method. The proposed BOLAP formulation solved by the VEPSO algorithm produces good quality (efficient) solutions in shorter time. The VEPSO algorithm converges faster to the Pareto-optimal front. The proposed approach generates several ‘good’ solutions (set of potential Pareto solutions). So, the decision maker has more flexibility and degrees of freedom for choosing among different good options in order to achieve better performances. Choosing $C_{16}$ from $\text{BTS}_{16}$ as a seed cell with a radius of 120 km achieves the best performance in this empirical application. To de-cluster the display of the solution on the map, we have chosen to show only the BTSs and LAs in Figure 5 and 6.
Table 4  The comparison between the telephone company and VEPSO algorithm to solve a BOLAP with 138 cells

<table>
<thead>
<tr>
<th>Comparing parameters</th>
<th>Telephone operator results</th>
<th>VEPSO results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution time</td>
<td>2 days</td>
<td>75.1 sec</td>
</tr>
<tr>
<td>Number of efficient solutions (scenarios)</td>
<td>2 or 3</td>
<td>7</td>
</tr>
<tr>
<td>Set of efficient solutions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(m28,874; 1.35788337026E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(339,854; 1.32230071501E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(28,873; 1.36357867723E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(28,890; 1.30752353469E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(424,122; 1.25972498936E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(453,339; 1.03395523989E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(458,181; 1.02371309392E8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best objective function $Z_1$ (LU)</td>
<td>251 141</td>
<td>28 873</td>
</tr>
<tr>
<td>Best objective function $Z_2$ (distance)</td>
<td>1.49E8</td>
<td>1.024E8</td>
</tr>
</tbody>
</table>

Figure 5  BTS to LA’s assignment, (a) using telephone company model (b) using VEPSO optimiser (see online version for colours)
Figure 5  BTS to LA’s assignment, (a) using telephone company model (b) using VEPSO optimiser (continued) (see online version for colours)

Figure 6  BTS to BSC’s assignment, (a) using telephone company model (b) using VEPSO optimiser (see online version for colours)
Expanding $L A_1$ boundary include $B T S_{13}$ improves the quality of solution as it is shown in Figure 5. This solution generates a rate of PL of ($\approx 15\%$, $\approx 28\%$) for $L A_1$ and $L A_2$ in comparison with the telephone operator solution ($\approx 18\%$, $\approx 32\%$). Only one solution from the set of efficient solutions is shown.

The telephone company solution divides the LA into new two smaller LAs which decreases the PL, but the number of LU and traffic increase. However, the solutions found with VEPSO method minimise both LU and distance cost functions. The Pareto-set found by the VEPSO algorithm is diverse (Table 4). The results show the efficiency of the proposed model and algorithm to solve BOLAP problem and produce multiple Pareto-optimal solutions where, in the other side, the method adopted by the operator generates in average only around two or three different scenarios to split the area. It is safe to conclude that the proposed model and VEPSO algorithm dominates the telephone company approach. The proposed model and algorithm will be implemented into a user friendly decision support system in order to choose one of the generated efficient solutions according to the decision maker preferences.

It is important to mention that the algorithm efficiency depends mainly on the number of iterations run, size of efficient solutions found, seed cells, and radius of neighbourhood. The seed cell and radius of neighbourhood are the parameters that affect at most the behaviour of our VEPSO algorithm (see Appendices A and B).
7 Conclusions

In this paper, we proposed a new formulation of the BOLAP problem, which minimises simultaneously LU and BTS-BSC distance costs under wireless phone network constraints. The BOLAP optimises both horizontal (by minimising the LU between LAs) and vertical perspective (by minimising the distance costs between BTSs and BSCs). The proposed model is a generalisation of the original LAP introduced by Markoulidakis and Sykas (1993). It includes the majority of practical system constraints. It is NP-hard problem. Therefore, we proposed a meta-heuristic based on VEPSO as a resolution approach. The VEPSO algorithm allows swarms to converge easily to the Pareto frontier. However, it depends on the initial seed cell around which region grows with a fixed radius. The algorithm improves the solution by performing global-best exchanges between the two swarms allowing faster convergence to the Pareto-optimal solutions. Each swarm is exclusively evaluated according to one of the two objective functions. Information obtained from swarms are exchanged to influence movement in the solution space.

The empirical results show that the proposed model and the VEPSO algorithm outperform the approach employed by the telephone company. The telephone operator has validated the model and a decision support system will be considered to optimise the LAP process. This work will be extended in order to improve of the optimisation algorithm. The BOLAP will be improved by considering additional engineering rules as well as uncertainty. The proposed method is applicable to other types of networks as 3G wireless networks, sensor networks and surveillance networks.

Acknowledgements

The authors would like to acknowledge the many helpful suggestions of the employees from the Tunisian telephony operator and also for the availability of data that were very helpful to run our programme and do a set of experimentations. We also would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References


A bi-objective location area planning for wireless phone network


Appendix A

Find the unit seed cell size

It can be observed from the results, summarised in Table 5, that different optimal LA designs are got for different seed cells by comparing the percentages of PL in each LAs. We select five different cells spread into different locations in our LA, in order to have different frontiers in the LA. The first column of Table 5 refers to the initial cell number from which we start the assignment (referred to as with presets in the mathematical formulation of the model). The second column denotes the computing time of our algorithm. The third column refers to the rate of PL in each LA, while the fourth counts the number of efficient solutions for each initial cell. The two last columns represent the best objective functions for $Z_1$ and $Z_2$.

Table 5 Variation of the seed cell

<table>
<thead>
<tr>
<th>Seed cell</th>
<th>CPU time (sec)</th>
<th>Paging/LA (%)</th>
<th>Number of efficient solutions</th>
<th>$Z_1$</th>
<th>$Z_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_5$</td>
<td>75.781</td>
<td>(19; 21)</td>
<td>9</td>
<td>106,587</td>
<td>1.028E8</td>
</tr>
<tr>
<td>$C_{33}$</td>
<td>93.968</td>
<td>(10; 35)</td>
<td>7</td>
<td>139,621</td>
<td>1.028E8</td>
</tr>
<tr>
<td>$C_{16}$</td>
<td>90.718</td>
<td>(15; 30)</td>
<td>5</td>
<td>33,749</td>
<td>1.039E8</td>
</tr>
<tr>
<td>$C_2$</td>
<td>92.437</td>
<td>(16; 29)</td>
<td>7</td>
<td>52,381</td>
<td>1.043E8</td>
</tr>
<tr>
<td>$C_{11}$</td>
<td>97.234</td>
<td>(11; 34)</td>
<td>8</td>
<td>173,216</td>
<td>1.033E8</td>
</tr>
</tbody>
</table>

We select the following cells: cell $C_5$ belongs to $BTS_5$, cell $C_{33}$ belongs to $BTS_{33}$, cell $C_{16}$ belongs to $BTS_{16}$, cell $C_2$ belongs to $BTS_2$ and cell $C_{11}$ belongs to $BTS_{11}$.

By comparing paging rates, the assignment that satisfies at most constraints of paging is the one that starts from seed cell $C_{16}$, with a PL of (≥15%, ≥30%) for a thresholds of
A bi-objective location area planning for wireless phone network

20% and 25% respectively for the LA and LA₂. Moreover, comparing to the objective function costs, choosing C₁₆ as initial cell, gives the lowest LU cost that minimises at most the first objective and then give the better border shape for our LA (see Figure 7).

Figure 7 The variation of the two objective functions for different seed cells (see online version for colours)

The best assignment of BTS to BSC minimising the second objective cost, is the one starting form cells C₅ and C₃₃. However, based on geographical constraints (LA is crossed by a highway) that decide on the way of how we will made the redefinition of our LA, we have to check on the feasibility of the proposed split with interaction with the decision maker. The assignment starting from cells C₅ and C₃₃ are not feasible due to the geographical location of different cells and the major road that crosses the LA (see Figure 7). Due to these geographical constraints, we chose the seed cell C₁₆. This scenario is the one that better satisfies all the constraints; therefore we adopt this alternative with an initial radius size equal to 115,000 m.

Appendix B

Impact of the radius R size

Based on experimentations above to determine the best seed cell, we choose cell C₁₆ belonging to BTS₁₆. We will vary, during these experiments, the radius of neighbourhood to get the best frontier for our LAs that will minimise both the LU and BTS-BSC distance.
costs. The first column of Table 6 refers the radius of neighbourhood, where the second refers to the computing time of our algorithm. The third columns, refers to the rate of PL in each LA. The fourth columns, represents the number of efficient solution for each size of radius. The two last columns represent the best value of objective functions $Z_1$ and the best value of fitness function $Z_2$.

Table 6  Radius size variations

<table>
<thead>
<tr>
<th>Radius size (m)</th>
<th>CPU time (sec)</th>
<th>Paging/LA (%)</th>
<th>Number of efficient solutions</th>
<th>$Z_1$</th>
<th>$Z_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000</td>
<td>75.109</td>
<td>(12; 33)</td>
<td>11</td>
<td>33,749</td>
<td>1.0288E8</td>
</tr>
<tr>
<td>120,000</td>
<td>75.178</td>
<td>(15; 28)</td>
<td>10</td>
<td>28,873</td>
<td>1.0237E8</td>
</tr>
<tr>
<td>125,000</td>
<td>75.078</td>
<td>(16; 29)</td>
<td>10</td>
<td>56,156</td>
<td>1.033E8</td>
</tr>
<tr>
<td>130,000</td>
<td>90.359</td>
<td>(16; 29)</td>
<td>8</td>
<td>56,156</td>
<td>1.028E8</td>
</tr>
<tr>
<td>150,000</td>
<td>95.296</td>
<td>(19; 26)</td>
<td>7</td>
<td>166,661</td>
<td>1.034E8</td>
</tr>
<tr>
<td>160,000</td>
<td>98.515</td>
<td>(19; 26)</td>
<td>8</td>
<td>477,642</td>
<td>1.028E8</td>
</tr>
<tr>
<td>170,000</td>
<td>90</td>
<td>(19; 26)</td>
<td>8</td>
<td>480,508</td>
<td>1.038E8</td>
</tr>
</tbody>
</table>

Figure 8  Variation of the two objective functions for different radius of neighbourhood (see online version for colours)
For a radius size equal to 120 km, we have an assignment with the lowest values of both objective functions (see Figure 8), satisfying thus our constraints, mainly the paging constraints. As we set a paging threshold of 20% and 25% for each LAs, we can see that for this radius size we have a better proportion of PL in each LAs. We conclude then that the best radius of neighbourhood, is a radius size of 120 km satisfying both objective functions $Z_1$ and $Z_2$. 