Université de Tunis
Institut Supérieur de Gestion

A Bi-objective Location area Planning in Wireless Phone System

Pour l'obtention du mémoire de mastère en Modélisation

Ons ABDELKHALEK

Sous la direction de
Dr.Saoussen KRICHEN

Année Universitaire: 2008 – 2009

To my parents for their availability and their continous support,

To my brother Anis and my sister Raja,

To all my friends who have provided a lot of moral support...

Acknowledgments

Thanks go to my supervisor Dr. Saoussen KRICHEN for the guidance she gave me during my work. She seems to always give the guidance and encouragement that was needed to solve, what at the time seemed to be insurmountable dilemmas. I greatly appreciate her knowledge as well as her experience.

I would like to thank the LARODEC laboratory to which I belong, especially Professor Khaled MALLOULI, Professor Fouad BEN ABDELAZIZ and Professor Mouhamed LIMAM.

Thanks go to my co-supervisor in TUNISIANA, Mr.Bassem BOUKHRIS, for allowing me to work with him on this subject, for his cooperation and full provision.

Thanks go to Mr.Wadi BELLAAJ in TUNISIANA for his cooperation and the valuable help he has given with regards to this work.

I want to express my sincere gratitude to Orascom Telecom Tunisia (TUNISIANA) and all the people who have contributed to the achievement of this work, as well as those who gave me the opportunity to integrate training, which was very helpful to understand the problem and learn knowledge in the discipline of managing the network communication system.

Last, but definitively not least, I want to thank my parents and all the people who were there when needed for help and encouragement.

Abstract

As the number to mobile phone users increases, new technologies appear and therefore their management becomes essential. Location Area Planning (LAP) problem plays an important role in cellular networks, to minimize the cost of signaling by assigning cells to areas, generated by the procedures of mobility.

We introduced in this research, a new extension of a Location Area Planning (LAP) problem in the wireless phone system, which is a bi-objective allocation of cells into a prefixed number of Location Areas (LAs). This new problem, namely the "Bi-objective Location area Design (BLD)", is modeled as a two-dimensional assignment including the entire Base Subsystem (BSS), and its resolution gives rise to a set of effective solutions. We modeled, first, a new mathematical formulation of our proposed problem taking into consideration engineering issues (location area planning, handover, location update, paging load, etc..), fitting with a real practical case study of our problem. It was proven that the LAP problem is NP-complete, requiring the use of an approximating method. Then, we adapted an algorithm based on Particle Swarm Optimization (PSO), namely the Vector Evaluated Particle Swarm Optimization (VEPSO), as a resolution strategy for our problem. Finally, we implemented and tested our approach, which was proved to be more efficient than the one adopted by a telephone operator to find the suitable assignment.

Key Words: *Location Area Planning problem, Telecommunication network, VEPSO, PSO, Bi-Objective problem, Paging, Handoff*

Résumé

Plus le nombre d'utilisateur de téléphones mobiles augmente, de nouvelles technologies apparaîssent et leurs gestion devient donc indisponsable. Le problème du Location Areas planning (LAP) joue un rôle important dans les réseaux cellulaires, afin de minimiser le coût des signalisations dû à l'affectation des cellules dans des Location Areas (LAs) qui est généré par les procédures de mobilité.
Nous avons présenté, durant cette recherche, une nouvelle extension du problème LAP dans le système de téléphonie sans fil, qui est une répartition biobjective des cellules dans un nombre prédéfini de LAs. Ce nouveau problème, à savoir le "Bi-objective Location area Design (BLD)", gére la totalité du sous-systeme (BSS) et sa résolution donne lieu à un ensemble de solutions efficaces. Nous avons modélisé, en premier lieu, une nouvelle formulation de notre problème en prenant en considération les questions d'ingénierie (location area planning, handover, location update, paging load, etc.), tout en étant conforme à l'étude d'un cas pratique pour notre problème. Et puisque il a été prouvé que la LAP est NP-complet, il est nécessaire d'utiliser une méthode approximative. Nous avons, de ce fait, adapté un algorithme basé sur le Particle Swarm Optimization (PSO), à savoir le Vector Evaluated Particle Swarm Optimization (VEPSO), en tant que stratégie de résolution à notre modèle. Enfin, nous avons testé et implémenté notre approche, qui a été révélé être plus efficace que celle adoptée par Orascom Telecom (Tunisiana), pour trouver l'affectation appropriée.

Mots clés: *Location Area Planning problem, Telecommunication network, VEPSO, PSO, Bi-Objective problem, Paging, Handoff*

Contents

Introduction **1**

1 General Assignment Problems **4**

1.1 Introduction 4

1.2 Mathematical Formulation of the basic AP 5

1.3 Variants of the GAP 6

1.3.1 Uni-objective GAP 6

1.3.2 Multi-Objective GAP 13

1.4 Applications of the GAP 18

1.4.1 Scheduling 18

1.4.2 Transportation and routing 18

1.4.3 Telecommunication 19

1.4.4 Production planning 21

1.4.5 Location 21

1.4.6 Supply Chain 22

1.5 Solution procedures for the GAP 22

1.5.1 Approximate Solution Approaches 23

1.5.2 Relaxation 25

1.5.3 Metaheuristic Solution Approaches 26

1.5.4 Exact Approaches 30

1.6 Conclusion 31

2 Wireless Network Design **32**

2.1 Introduction 32

2.2 Architecture of the GSM network 33
2.2.1 The Base Station Subsystem 33
2.2.2 The Network Subsystem 35
2.2.3 The Operating Sub-System 35
2.3 Cells and Location area concept 36
2.4 Network aspects 38
2.4.1 GSM frame structure 39
2.4.2 Radio Resources Management 40
2.4.3 Mobility Management 40
2.4.4 Connection Management 45
2.5 Location Area Planning (LAP) problem in GSM network 45
2.5.1 Problem description 45
2.5.2 A literature review on the Location area planning (LAP) 47
2.5.3 Tunisiana LAP Model: TLPM 52
2.6 Conclusion 53

3 A Biobjective Location area planning and connection of antennas in GSM networks: A new solution approach 55
3.1 Introduction 55
3.2 Problem formulation 58
3.2.1 Cardinality of BLD problem 65
3.2.2 Multi-objective optimization problem 66
3.3 VEPSO approach for the BLD problem 67
3.3.1 Particle swarm optimization 67
3.3.2 Vector evaluated PSO - VEPSO 71
3.3.3 VEPSO for the BLD 73
3.4 Computational Experiments 79
3.4.1 Implementation of the Algorithm 80
3.4.2 Run of the VEPSO Algorithm 84
3.4.3 Sensitivity analysis 89
3.5 Conclusion 103

General conclusion **105**

A Computational experiment of the BLD **107**

Bibliography **113**

List of Symbols & Abbreviations

Symbols	Description
N	Number of cells to be assigned
M	Number of LAs
Z	Number of BTS
W	Number of BSC to be installed
$\mathbf{C_i}$	Set of cells $i = 1, .., n$
$\mathbf{c_i}$	Call traffic of cell i at busy hour (in Erlangs)
$\mathbf{r_i}$	Number of TRXs for each cell i
$\mathbf{h_{is}}$	Handover rate from i^{th} cell to s^{th} cell
$\mathbf{w_{is}}$	Assignment of ith cell and s^{th} cell in the same or different LAs
$\mathbf{x_{ir}}$	Assignment of the ith cell to the rth LA
$\mathbf{y_{jkr}}$	Assignment of the jth BTS to the kth BSC and the jth BTS to the rth LA
$\mathbf{a_{ij}}$	Assignment of the ith cell to the jth BTS
$\mathbf{v_{ir}}$	Presets: fixing the assignment of the ith cell to the rth LA
$\lambda_\mathbf{i}$	the average number of incoming calls to cell i per unit time
$\mathbf{PB_r}$	The paging bound per LA_r.
$\mathbf{D_{jk}}$	Distance between the jth BTS and the kth BSC
$\mathbf{N_k^{BSC}}$	Maximum number of links to the kth BSC useful for the connection with the BTS
$\mathbf{C_k^{BSC}}$	Call traffic capacity of the kth BSC (in Erlangs)
$\mathbf{R_k^{BSC}}$	TRX capacity constraint of the kth BSC
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station

GSM	Global System for Mobile communication
HLR	Home Location Register
HO	Handover
LA	Location Area
LU	Location Update/Location Registration
PA	Paging Area
MS/MT	Mobile Station/Mobile Terminal
MSC	Mobile Services Switching Center
SIM	Subscriber Identity Module
TRX	Base Station Transmitter
Um	Air-interface
VLR	Visitor Location Register
TCH	Traffic channels
CCCH	Common Control CHannel
SACCH	Slow Associated Control Channel
FACCH	Fast Associated Control Channel
SDCCH	Stand alone Dedicated Control Channel
BCH	Broadcast Channels
CCCH	Common Control Channels
PCH	Paging Channel
WAP	Weighted Assignment Problem
BGAP	The Bottleneck GAP
MGAP	The Multilevel GAP
EGAP	The Elastic GAP
DGAP	The Dynamic GAP
SGAP	The Stochastic GAP
MRGAP	The multi-resource GAP
GQAP	Generalized quadratic assignment problem
GAPS2	GAP with special ordered set of type II
BiGAP	Bi-objective GAP
MKARP	Multiple Knapsack Problem with assignment restrictions

List of Figures

1.1 Assignment of three users to three antennas 16

2.1 Functional Architecture of GSM *[John Scourias, 1996]* 34
2.2 Network area 36
2.3 Clusters of cells 37
2.4 Grouping cells into LA 38
2.5 TDMA frames and multiframes for speech and data 39
2.6 Handover area inter-cells 41
2.7 Paging Procedure in GSM network *[Online's TUNISIANA Document(OTT)]* 42
2.8 Location area update types *[Online's TUNISIANA Document(OTT)]* . . 43
2.9 Location updates and searching of mobile terminals in a GSM Network . 44
2.10 Trade-off between LU and Paging load in LAP problem 46
2.11 Flowchart of TLPM 54

3.1 Location area division 56
3.2 BLD problem: "LAs Design" 59
3.3 BLD Database 83
3.4 LA border in TUNISIANA 86
3.5 Assignment of BTS to BSC in TUNISIANA 87
3.6 One hundred thirty eight cell configurations in VEPSO application . . . 88
3.7 Number of efficient solutions for different sizes of swarms 91
3.8 The CPU time for different sizes of swarms 91
3.9 Number of efficient solutions for different seed cells 93
3.10 CPU times for different seed cells 94

3.11 The variation of the first objective function for different seed cells 94
3.12 The variation of the second objective function for different seed cells . . . 95
3.13 Percentage of paging load/LA for different radius of neighborhood 96
3.14 Number of efficient solutions for different neighboring radius 96
3.15 Variation of the objective function Z_1 for different radius of neighborhood 97
3.16 Variation of the objective function Z_2 for different radius of neighborhood 97
3.17 Number of efficient solutions for different velocity V 99
3.18 Number of efficient solutions for different inertia factor W 99
3.19 Comparision between TLPM and VEPSO of Cells to LA assignment . . . 102
3.20 Comparision between TLPM and VEPSO of BTS to BSC assignment . . 102

List of Tables

2.1 Summary of the literature review on LAP problem 48

3.1 VEPSO parameters 85
3.2 Results for problem size 20 and 50 iterations 85
3.3 Results for variation in the number of particles in each swarm 90
3.4 Results of varying the seed cell 92
3.5 Results for different radius of neighborhood 98
3.6 Results of varying velocity V 100
3.7 Results of varying the inertia factor 101
3.8 Comparison between TLPM and VEPSO algorithm 103

General introduction

The increase in mobile phone use encourages the development of wireless networkS to expand network capacity in order to redistribute resources before any quality of service degradations occurs. One of the must common methods to increase the capacity of networks, is to reduce the size of the cells so as to allow greater frequency of reuse. Nevertheless, as the size of cells decreases, the number of cells and the number of handovers (or handoffs) also increase, which further complicates the advantage of location function in the network.

The current knowledge of the mobile location is maintained by a registration procedure, namely Location Update (LU), in which mobiles announce their presence upon entering a new location area. This LU affects all mobiles in idle mode or after a call completion. The new location area will be stored in a database to be used for paging the mobile the next time the mobile receives an incoming call. The paging process is the search of the mobile station on the arrival of a mobile-terminated call.

To reduce the cost of the signaling update, the most common method is to cluster cells into areas. Each group of cells is called a Location Area (LA). LA planning has a necessary role in cellular networks caused by the trade-off between paging and Location Update (LU) signaling. The split of an LA (or the assignment of cells) is a methode that reduces the traffic on paging channels for freeing them to the reception and the emission of calls.

Cellular network planning under a budgeted cost requires decision-makers to group cells into LAs so that the operating cost is minimized. So, the problem of how to group the cells into LAs is called the location area planning (LAP) problem. The set of new LAs must contain few cells and have a reasonable paging rate, in order to minimize simultaneously the LU and paging costs. High LU traffic can be caused by small location

zones due to high concentration of subscribers or to high mobility between the database service areas.

The allocation of cells is also affected by the distance cost between BTS and BSC. By increasing the distance between BTS and BSC, the paging load in the LAs increases and thus it reduces the LU. Minimizing both LU and BTS-BSC distance cost can lead to better results.

Being taken into consideration the LAP problem by minimizing both the LU under Paging constraints, and distance cost of BTS-BSC under BSC capacity constraints, we present in this research a new problem, namely the "Bi-objective Location area Design (BLD)", that includes the entire of BSS subsystem in the GSM networks architecture.

To resolve this new model, I did an internship in Orascom Telecom (TUNISIANA) to access their personal documents and integrate their workplace in order to better understand the concept of LA and the processor of Paging and LU.

The problem of LAP was first introduced by Markoulidakis and Sykas (1993) as a model based on the trade-off between the amount of signaling generated due to paging and location updating. They used real data from an area of Helsinki to validate their model. For these large problems, exact solution methods are impractical and hence many heuristics has been introduced to get better solutions to the problem. The heuristics introduced include random generation methods, Tabu Search algorithms (Desmestichas et al, 1999, 2000), Greedy Search algorithms (Plehn, 1995), Simulated Annealing [(Demirkol et al, 2004); (Desmestichas et al, 1999); (Saraydar and Rose, 1998)], Genetic Algorithms (GA) [(Wang et al, 1998); (Gondim, 1996); (Chan et al, 2000)], and Grouping Genetic Algorithms [(Wang et al, 1998); (Vroblefski and Brown, 2006)].

These approaches previously defined, solved the uni-objective problem of assigning cells into Las, trying to minimize LU with fixing a paging boundery. In our case, we discuss the bi-objective case for the assignment cells, minimizing both the LU and the distance cost of BTS-BSC.

As our problem is Multiobjective, we adpte the Vector Evaluated Particle Swarm Optimization (VEPSO) heuristics to our model. It was proven that VEPSO is suitable for multiobjective optimization problems (Hassan.R et al, 2005), produces solutions of high quality with few parameters to adjust and gets better results in a faster and cheaper way compared with other methods.

Because the BLD is essentially a problem of assigning cells to LAs, the first chapter of this research is devoted to the introduction of the General Assignment Problem (GAP), its variant and its different applications. In the second chapter, we give an overview on phone wireless network by presenting its different components useful for understanding our problem. Then we will introduce the basic LAP problem, providing details on previously applied solution methodologies and the strategy adopted by TUNISIANA for LA Redistribution (TLPM). In the third chapter, we present an extention of the LAP problem, namely the BLD problem, based on mathematical formulation. Then, we describe the VEPSO algorithm resolution strategy, the experimental design, results, and analysis. Conclusions are summarized in the final section.

Chapter 1

General Assignment Problems

1.1 Introduction

The Assignment Problem (AP) is one of the fundamental combinatorial optimization problems since it can modal several practical applications such as supply chain, telecommunication, production planning, etc. It's a classical operations research model, which deals with the assignment of a number of tasks to a group of agents (Hillier and Lieberman, 1995).

The General Assignment Problem (GAP) consists of finding the optimal assignment of n items (tasks) into m knapsacks (agents) without exceeding the capacity of any knapsack. It is required to perform all tasks by assigning exactly one agent to each task in such a way that the total cost of the assignment is minimized.

There exist many combinatorial optimization problems which have the same structure as the GAP. Ferland (1997) noted that the basic assignment structure is present in several combinatorial optimization problems such as the GAP, the Timetabling Problem, the Graph Coloring Problem and the Graph Partitioning Problem, so the GAP can be considered as a generalization of the Assignment Problem.

The GAP is known to be NP-hard, however the problem of finding a feasible assignment is NP-complete.

Since Ross and Soland (1975) have introduced the GAP, much researchs has been published on this popular combinatorial optimization problem. Cattryse and Van Wasenhove (1992) published a work which is focused on the solution algorithm for the GAP. A gen-

eral overview of heuristics and metaheuristics approaches for the GAP was summarized by Osman (1995), followed by Yagiura and Ibaraki (2004,2007).
Ferland (1997) and Öncan (2007) have presented a survey on the general assignment problem and its applications. GAP has also been considered as the generalization of the Pairing Problem (Toktas, 2004), the Semi Matching Problem (Harvy et al, 2006) and the Parallel Machine scheduling Problem (Shmoys and Tardos, 1993).
In this chapter, we start by intoducing a mathematical formulation of the GAP in section (1.2). Next we mention, in section (1.3), variants of the GAP and present some of the combinatorial optimization problems that have the same structure as the GAP. Then, we try to present in section (1.4) some of the real-world applications of the GAP. Finally, in section (1.5), we summarize some of the recent developments of the solution approaches for this problem.

1.2 Mathematical Formulation of the basic AP

The basic formulation of the GAP is represented as follows:

$$\textbf{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \qquad (1.1)$$

S.t.

$$\sum_{j=1}^{n} R_{ij} X_{ij} \leq b_i \;\; i = 1, .., m \qquad (1.2)$$

$$\sum_{i=1}^{m} X_{ij} = 1 \;\; j = 1, .., n \qquad (1.3)$$

$$X_{ij} \in \{1, 0\} \quad \forall i, j \qquad (1.4)$$

Where $\{C_{ij}\}$ is the cost of assigning job j to agent i, bi is the capacity of the agent i and $\{R_{ij}\}$ is the weight of the job j if assigned to agent i.

- The objective function (1.1) refers to the minimization of the total assignment cost of items (jobs) to knapsacks (agents).
- Constraints (1.2) define the capacity availability restriction of each knapsacks (agent) and are simply referred to as the capacity constraints.
- Constraints (1.3) ensure that each item (job) is assigned to exactly one knapsack (agent)and are known as the semi-assignment constraints.

- Finally, in constraints (1.4), the variables $\{X_{ij}\}$ represent the assignment of item j to knapsack i, by taking value 1 if the assignment is made and 0 otherwise.

1.3 Variants of the GAP

We propose, in what follows, a survey on the GAP for both single and multiobjective context. The single objective, as stated in (1.2), was widely studied with several variations. The multiobjective GAP considers additional conflicting objectives that should be optimized simultaneously.

1.3.1 Uni-objective GAP

The single objective assignment problem (AP) is an integer programming problem that can be formulated as a linear program, due to the total unimodularity of the constraint matrix. Efficient algorithms to solve it, such as the Hungarian method or the successive shortest paths method (Papadimitriou and Steiglitz, 1982; Ahuja et al., 1993), provide the optimal solution.
We will present some of the variants of the uni-objective GAP.

Weighted Assignment Problem -WAP-

Ross and Zoltners (1979) formulated the GAP as a special case of the Weighted Assignment Problem (WAP) since it contains a generalization of both the constrains and the variables of the GAP.
The authors introduced several models equivalent to the WAP, like Variable Agent Fixed Capacity Model, and then discussed resolution approach and applications of the WAP.
They define a binary variable X_{ijk} as follows:

$$X_{ijk} = \begin{cases} 1, & \text{if task } j \text{ is completed by agent } i \text{ at performance level } k \\ 0, & \text{otherwise.} \end{cases}$$

The WAP formulation is:

$$\textbf{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k \in K_{ij}} C_{ijk} X_{ijk} \tag{1.5}$$

S.t.

$$\sum_{i=1}^{m} \sum_{k \in K_{ij}} X_{ijk} = 1 \quad j = 1, .., n \tag{1.6}$$

$$a_i \leq \sum_{j=1}^{n} \sum_{k \in K_{ij}} r_{ijk} X_{ijk} \leq b_i \quad i = 1, .., m \tag{1.7}$$

$$\sum_{i=1}^{m} \sum_{k \in K_{ij}} S_{ijk} X_{ijk} \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} e_j \quad j = 1, .., n \tag{1.8}$$

$$X_{ijk} = \{1, 0\} \quad \forall i, j, k \tag{1.9}$$

- Constraints (1.6) ensure that each task j is performed by agent i at a certain performance level k. And a performance levels K_{ij} are fixed for each agent i and tasks j.
- Constraints (1.7) refers to the resource consumption limits of each agent.
- Constraints (1.8) indicate the boundaries of the used resource when the job j is completed.

It's clear that when there is no boundary on the used resource ($a_i = 0$), the WAP becomes a GAP.

The bottleneck Gap -BGAP-

The bottleneck Gap is a variation of the GAP proposed by Mazzola and Neebe (1988,1993) and Martello and Toth (1995). It tries to minimize the maximum penalty of assigning n items into m knapsacks. In this case, the objective function (1.1) is replaced by (1.10):

$$\textbf{Min}(\textbf{Max} C_{ij} X_{ij}) \quad i = 1, .., m \;\; j = 1, .., n \tag{1.10}$$

Applications of the BGAP, presented by Martello and Toth (1995), belong to the public sector like n urban areas have to be assigned to m schools and the objective is to minimize the maximum travel time.

Non linear capacity constrained GAP

Discussed by Mazzola (1989), with replace constraints (1.2) by the following nonlinear capacity constraints:

$$fi(X_{i1}, ..., X_{in}) \leq b_i \; i = 1, .., m \; (1.11)$$

Where $fi(X_{i1}, ..., X_{in})$ is a real-valued polynomial function. This capacity function may be useful in the case that there is nonlinear interaction for the set of tasks assigned to the same agent.

The multilevel GAP -MGAP-

The multilevel generalized assignment problem (MGAP) is a variant of the GAP in which agents can execute tasks at different efficiency levels and with different costs. It was first described by Glover et al (1979) in the context of large-scale allocations of tasks to machines. It deals with the determination of the minimum cost assignment of tasks to agents with varying efficiency levels. More than the classical GAP, agents can perform tasks at more than one level.

One additional constraint are inserted in the basic modal (1.1)-(1.4) and constraints (1.11):

$$\sum_{j=1}^{n} \sum_{k \in K_{ij}} R_{ijk} X_{ijk} \leq b_i i = 1, .., m \quad (1.12)$$

MGAP can also be considered as a special case of the WAP when $a_i = 0$.

A fuzzy genetic multi-objective optimization algorithm for the MGAP, was proposed by Hajri-Gabouj (2003) and applied on the task-operator-machine assignment problem in clothing industry. The main objective is to minimize the total execution time and the secondary ones are to minimize the deviation from perfect load balance among the operators, to limit the inter-operator communication cost, the number of machines and the number of operators. In order to avoid the non-linearity and the complexity of the secondary objectives, he relaxed them using fuzzy penalty functions then proposed the genetic multi-objective algorithm for the solution of the relaxed problem.

Caselli and Righini (2006) developed a first exact algorithm for the MGAP based on branch-and-price. They decompose it into master problem with set-partitioning constraints and a pricing subproblem which is a multiple-choice knapsack problem.

The Elastic GAP -EGAP-

The Elastic Gap (EGAP) was studied by Nauss (2004). In this variant, agents are allowed to violate capacity constraints at additional cost. The objective function of the

elastic Gap is as follows:

$$\textbf{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} + \sum_{i=1}^{m} (d_i u_i + e_i v_i) \quad (1.13)$$

Where variables u_j and v_j define respectively nonnegative undertime and overtime, and which state the idle resource of agent i and the additional resources used by agent i. Constraints (1.2) in the classical GAP are replaced with the following:

$$\sum_{j=1}^{n} R_{ij} X_{ij} + u_i - v_i = b_i \; i = 1, .., m \quad (1.14)$$

Nauss (2004) has defined upper bounds for the predefined variables as follows:

$$0 \leq u_i \leq g_i \; i = 1, .., m$$
$$0 \leq v_i \leq h_i \; i = 1, .., m$$

He proposed a branch and bound algorithm by using linear programming cuts, Lagrangian relaxation and penalties tests. He also performed feasible solution generators to decrease the upper bound.

The dynamic GAP -DGAP-

Introduced by Kogan et al (1997), the dynamic assignment problem is a fundamental problem in routing and scheduling. It's a special case of the dynamic vehicle routing problem, without the complexities of vehicle consolidation.
The difference from the classical GAP is that the sequence in which the agents perform the jobs are taken into account.
Considering the due date and completion time of each job of a scheduling problem, the main objective function is to minimize the inventory, shortage and assignment costs.

The stochastic GAP -SGAP-

Two types of stochasticity can be pointed out in the GAP. In the first, the capacity of agent and the amount of resource needed to process the job are not known in advance. This stochasticity may arise in the case of job assignment to programmers, where the time to finish the computer programs is not deterministic and the execution time of the computer programs varies.
In the second, the presence or absence of a job are uncertain and we can model the

resource requirement of each job as a random variable with Bernoulli distribution.
Albareda-Sambola et al (2006) have addressed the second type of stochasticity by considering the uncertainty about which jobs have to be actually performed once they are all assigned.
Albareda-Sambola and Fernández (2000) proposed an approximation model based heuristics for the second type of stochasticity, where the objective is to minimize the assignment cost and the expected penalty costs incurred due to reassignment and/or unfinished jobs.
Kogan et al (2005) extended the dynamic GAP dealing with the stochastic demands, job-agent limitations and periodic review of completed jobs. They assumed multiple-task relationships at each time interval, limiting both the number of agents performing a job and the number of jobs that can be performed by an agent. They also presented two-world examples of this problem. The first example is the allocation of problems (jobs) to a set of cooperation computers (agents) where the objective is to minimize delays and processing costs. The second example was related to the scheduling of pre-media printing operations over the workers in a flexible production system.

The multi-resource GAP -MRGAP-

The MRGAP is another variant of the GAP, where a number of different constraining resources are associated with each agent. It was studied by Mazolla and Wilcox (2001), Gavish and Pirkul (1991) and Yagiura et al (2004). In this variant, the capacity constraints (1.2) of the classical GAP are replaced by the following:

$$\sum_{j=1}^{n} R_{ijq} X_{ij} \leq b_{iq} \qquad i = 1, .., m \; q = 1, .., Q \quad (1.15)$$

Where each agent i is constrained by a set of resources $q = 1, .., Q$. Then for each agent i and for each resource q, there is b_{iq} unit of resources available. For each resource q, R_{ijq} is the resource required by the task j and used by the agent i.
Various real case studies can be modeled as MRGAP, the typical one arises as a subproblem in solving the Vehicle Routing Problem, where the others appear in distributed computer system (Gavish and Pircul,1986), Job shop scheduling and telecommunication network design.
Shtub and Kogan (1998) have discussed the dynamic MRGAP where the demands for

tasks and capacity assignment are dynamic. Toktas et al (2006) have discussed a generalization of the MRGAP, namely the Collectively Capacitated GAP, where different constraining resources are collectively associated with all agents rather than with each agent individually.
Campbell and Langevin (1995) described the problem to find the optimal assignment of snow removal sectors to snow disposal sites, named the Snow Disposal Assignment Problem (SDAP). In the SDAP, the resources used by the sites are the annual snow capacity and the maximum snow receiving rate.

The Generalized Multi-Assignment Problem

The Generalized Multi-Assignment Problem was introduced by Park et al (1998) as a generalization of GAP, in which we are able to assign an item to different knapsacks. Constraints (1.2) of the classical GAP are replaced by the following:

$$\sum_{i=1}^{m} X_{ij} \geq t_j \;\; j = 1, .., n \quad (1.16)$$

Where t_j is a parameter such that $t_j \leq m$ for $j = 1, .., n$. When it's equal to 1, the Generalized Multi-Assignment problem becomes the classical GAP.
Park et al (1998) proposed a Lagrangian dual based branch and bound algorithm. They mentioned a real-world application of this problem in a distributed database system where each file is stored in multiple sites for more reliable and faster response time.

Generalized Quadratic Assignment Problem -GQAP-

Introduced by Li and Ma (2003), the Generalized Quadratic Assignment Problem (GQAP) deals with the assignment of a set of facilities $j = 1, .., n$ to a set of sites $i = 1, .., m$, while the assignment cost and traffic cost are minimized and do not exceed the capacity of the total weight of all facilities assigned.
The objective function (1.1) of the classical GAP will be replaced by:

$$\textbf{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} + \gamma \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{o=1}^{m} \sum_{p=1}^{n} \alpha_{io} \beta_{jp} X_{ij} X_{op} \quad (1.17)$$

where α_{io} is the distance between sites i and o, β_{jp} is the traffic intensity between facilities p and j and γ is the unit traffic cost.

Authors have proposed a linearization for the GQAP based on the Quadratic Assignment Problem linearization by Frieze and Jadegar (1983). A mimetic algorithm for the GQAP was proposed by Gordeau et al (2006). They also mentioned an application in the management of container yards where the problem consists of the location of container groups in the storage area so that the yard handling movements are minimized.

GAP with special ordered set of type II -GAPS2-

"We said that a set of variables $\{x_1, x_2, .., x_n\}$ *is a special ordered set of type II, if* $x_i x_j = 0$ *whenever* $|i - j| \geq 2$*"* (Öncan, 2007).

A Gap with special ordered set of type II (GAPS2) was first introduced by Farias et al (2000). It deals with the allocation of tasks to time period, where x_{ij} denote the fraction of task j allocated to time-period i.

We replace constraints (1.4) in the classical GAP with the following:

$$\begin{cases} x_{i-1\ j} = 0 \ \ and \ \ x_{i+1\ j} = 1 - x_{ij} \\ or \\ x_{i+1\ j} = 0 \ \ and \ \ x_{i-1\ j} = 1 - x_{ij} \end{cases} \quad (1.18)$$

Farias et al (2000) mentioned the fibre optics cable to produce scheduling application and allowed to share jobs between adjacent time periods. An exact solution methods based on polyhedral approaches, was proposed by the authors, to solve small instances to optimality.

French and Wilson (2007) have recently adapted the algorithm of Trick (1992) and proposed an LP-based heuristic.

Other variants

Another extention of the GAP is the Load Balancing Problem (LBP). The purpose is to find an optimal assignment of a set of jobs to a set of machines. There is different type of objective function for this problem: the minimization of makespan, the maximization of the fairness in job assignment.

Another variant of the GAP is the Covering Assignment Problem (CAP) where the objective is to determine the allocation of farms to factories with minimum transportation

cost such that each farm is assigned to exactly one factory and the demand of each factory is satisfied.
Dobson and Nambimadon (2001) discussed the *Batch Loading* and the *scheduling Problem*: the Batch Loading problem is the problem of assigning jobs to batches, where the Batch Scheduling Problems is the problem of sequencing batches.
The Batch Loading problem is a special case of the basic GAP with $a_j = r_{ij}$ where a_j is the volume of job j, and $b = b_i$ where b is the capacity of the processor. $X_{ij} = 1$ if and only if job j is assigned to batch i. Constraints (1.2) of the classical GAP becomes:

$$\sum_{j=1}^{n} a_j X_{ij} \leq b \quad i = 1, .., m \quad (1.19)$$

Cheng et al (1996) have studied an other application on the GAP in the cellular manufacturing named the Machine Assigned Problem (MAP). The objective is to minimize the utilization cost of allocated machines and the cost of intercellular moves. In this problem at least one machine have to be allocated to one cell and the total operating time of all machines allocated to a machines is less than the available operating time of the cell. In the MAP the constraints are the same as the GAP (1.2)-(1.4) with $R_{ij} = 1$.
Nutov et al (2006) addressed a special case of the GAP: Max-profit GAP with a fixed profit where each item j has a fixed profit p_j in every knapsack i. The objective function is as follows:

$$\textbf{Max} \sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} \quad (1.20)$$

1.3.2 Multi-Objective GAP

Multi-Objective GAP deals with a set of efficient solutions. Because there is no unique global best solution in the multi-objective problem, to find a solution for this kind of problem is to find a set of solutions (Pareto Optimal Set).
We will now introduce different variants for the multi-objective GAP.

Bi-objective GAP -BiGAP-

Described by Zhang and Ong (2007), they proposed an LP-based heuristic as a resolution approach to this problem. The problem can be formulated as follows:

$$\textbf{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}^{k} X_{ij} \;\; k = 1, 2 \quad (1.21)$$

S.t.

$$\sum_{i=1}^{m} X_{ij} = 1 \;\; j = 1, .., n \quad (1.3)$$

$$\sum_{j=1}^{n} X_{ij} = 1 \;\; i = 1, .., m \quad (1.22)$$

$$X_{ij} \in \{1, 0\} \quad \forall i, j \quad (1.4)$$

where all objective function coefficients C_{ij}^{k} are non-negative integers and $X = (X_{11}, .., X_{mn})$ is the matrix of decision variables.

Let X denote the set of all feasible solutions of BiGAP. $X \subset \{1, 0\}^{n^2} \subset R^{n^2}$ is also called the feasible set in the decision space. $Z = \{z(x) : x \in X\} \subset N^2 \subset R^2$ is called the feasible set in the objective space.

In bi-objective optimization, there is generally no feasible solution which minimizes both objectives simultaneously. The authors mentioned also several applications for the BGAP as a production planning application, where the C_{ij}^{1} and C_{ij}^{2} respectively represent the cost and the time where the job j is assigned to machine i.

An other application of the BiGAP is the emergency service zone, where C_{ij}^{1} and C_{ij}^{2}, respectively, represent the profit and traveling cost incurred by assigning zone j (customer) to service unit i (vehicle).

Ulungu and Teghem (1995) proposed a two-phase algorithm for the bi-objective assignment problem (BAP). The two phase method is a general framework for solving Multi-Objective Combinatorial Optimization (MOCO) problems (Ulungu and Teghem, 1995), although it has never been applied for problems with more than two objectives.

Ulungu and Teghem (1995) denote by S the set of efficient solutions found by the algorithm. $S \leftarrow \{x^1, x^2\}$ is initialized with the two lexicographic optimal solutions corresponding to $lexmin_{x^1 \in X}(z_1(x), z_2(x))$ and $lexmin_{x^2 \in X}(z_2(x), z_1(x))$, respectively.

The main idea is to use efficient algorithms for the single objective problem to compute efficient solutions.

- ❏ Phase 1: supported efficient solutions are computed. It is based on Geoffrions theorem (Geoffrion, 1968) stating that an optimal solution of (BAP) with an efficient positive weights.

- ❏ Phase 2: informations from supported efficient solutions identified by Phase 1, are used to reduce the search space for computing the non-supported efficient solutions.

In implementations, Phase 2 is generally enumerative.

In the bi-objective case (Przybylski et al. 2007), Phase 2 explores the triangles defined by two consecutive supported solutions in objective space.
The method has been applied to a number of problems other than (BAP), e.g., network flow (Lee and Pulat, 1993; Sedeño-Noda and González-Martín, 2001), knapsack (Ulungu, 1993; Visée et al., 1998), and spanning tree (Ramos et al., 1998). Some experimental results were published by Tuyttens et al. (2000). But the results obtained by Degoutin and Gandibleux (2002), using a Mixed-Integer Programming (MIP) solver, demonstrated more efficient solutions for the same instances. This put the validity of the implementation in question.

Multiple Knapsack Problem with Assignment Restrictions -MKARP-

The Multiple Knapsack Problem with Assignment Restrictions (MKARP) was introduced by Dahl and Foldnes (2006). The presence of assignment restrictions is a generalization of the well known multiple knapsack (MK) problem introduced by Martello and Toth (1990).
MKARP is a special case of the GAP. In MKARP, we give a set $N = \{1, ..., n\}$ of items, and a set $M = \{1, ..., m\}$ of knapsacks. With every item $i \in N$ is associated a weight $w_i \geq 0$ and a profit $p_i \geq 0$, and every knapsack $j \in M$ has a capacity $c_j \geq 0$. In addition there are assignment restrictions, given by sets $A_i \subseteq M$ for $i \in N$. These sets specify to which knapsacks an item is assignable. In other words, A_i is the set of knapsacks that can hold item i. For each knapsack j we let $B_j \subseteq N$ denote the set of items that are assignable to j. This defines a bipartite graph G with color classes M and N and with the neighbors of $j \in M$ given by A_i.
The problem can be formulated as follows:

$$\textbf{Max} \sum_{i \in N} \sum_{j \in A_i} p_i x_{ij} \qquad (1.23)$$

S.t.

$$\sum_{i \in b_j} w_i x_{ij} \leq C_j \;\; j \in M \qquad (1.24)$$

$$\sum_{j \in a_i} x_{ij} \leq 1 \;\; i \in N \qquad (1.25)$$

$$X_{ij} \in \{1, 0\} \qquad (1.4)$$

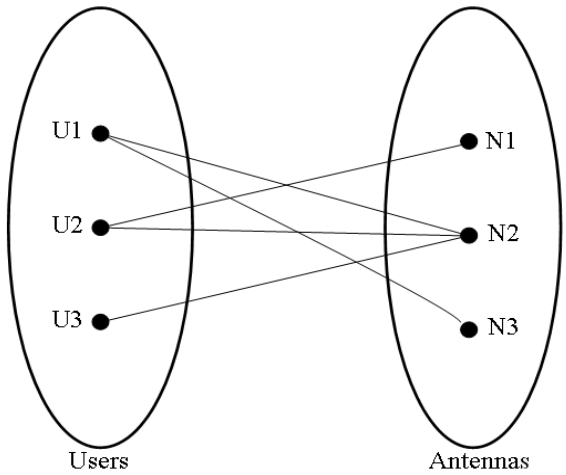

Figure 1.1: Assignment of three users to three antennas

The situation where $w_i = p_i$ for all items i, was first studied as an independent problem by Dawande et al. (2000). The assignment of users to antennas is a MKARP model, where we interprete the antennae as knapsacks and mobile phone users as items (see Figure 1.1).

The main objective is to decide for each user whether a connection can be given to him, and, if this is the case, to which antenna he can be connected. This must be done so that the total communication flow (given by some reasonable objective function) in the whole region is maximized.

Dawande et al. (2000) assumed that, when $w_i = p_i$ for all items i, MKARP is still NP-hard. Furthermore, this problem without assignment restrictions is called the Multiple Subset Sum Problem (Caprara, Kellerer and Pferschy, 2000).

Ferreira et al. (1996) present a cutting plane method and heuristics for a slightly more general problem than MKARP, together with some interesting real-life applications.

All variants of the GAP mentioned above are summarized in the following table.

Variants of the GAP	Main idea
Weighted Assignment Problem *(WAP)*	• Generalize both the constrains and the variables.
The Bottleneck GAP *(BGAP)*	• Minimize the maximum penalty of the assignment of n items into m knapsacks.
Non linear Capacity constrained GAP	• Used when there is nonlinear interaction for the set of tasks assigned to the same agent.
The Multilevel GAP *(MGAP)*	• Agents can execute tasks at different efficiency levels with different costs.
The Elastic GAP *(EGAP)*	• Agents are allowed to violate capacity constraints at additional cost.
The Dynamic GAP *(DGAP)*	• It's a fundamental problem in routing and scheduling and the sequence in which the agents perform the jobs are taken into account.
The Stochastic GAP *(SGAP)*	∘Type 1: The capacity of the agent and the amount of resource needed to process the job are not known a priori, ∘Type 2: The presence or absence of job are uncertain
The multi-resource GAP *(MRGAP)*	• A number of different constraining resources are associated with each agent.
The Generalized Multi-Assignment GAP	• Can assign an item to different knapsacks.
Generalized Quadratic Assignment Problem *(GQAP)*	• Assignment of a set of facilities to a set of sites while the assignment cost and traffic cost are minimized while not exceeding the capacity of the total weight of all facilities assigned.
GAP with special ordered set of type 2 *(GAPS2)*	• Deals with the allocation of tasks to time period.
Load Balancing Problem *(LBP)*	• Find an optimal assignment of a set of jobs to a set of machines.
Covering Assignment Problem *(CAP)*	• Determine the allocation of farms to factories.
Batch Loading problem	• The problem of assigning jobs to batches.
Machine Assigned Problem *(MAP)*	• Minimizing the utilization cost of allocated machines and the cost of intercellular moves.
Bi-objective GAP *(BiGAP)*	• There is no feasible solution which minimizes both objectives simultaneously.

Multiple Knapsack Problem with Assignment Restrictions *(MKARP)*	• With every item are associated a weight and a profit and every knapsack has a capacity.

1.4 Applications of the GAP

Several applications of the GAP are of interest. In this section we will present some of the applications and real-life problems where the GAP appears as a subproblem.

1.4.1 Scheduling

Zimokha and Rubinstein (1988) discussed many applications of the GAP appearing in resource scheduling. Drexl (1991) has addressed the job assignment problem arising in non-preemptive resource constrained project networks where the objective is to find the minimum cost assignment of jobs (items) to a set of resources (knapsacks) for a limited period of time.

1.4.2 Transportation and routing

Ruland (1999) solved the GAP to determine the assignment of passengers to flights. The primary objective function is to minimize the number of nights that the passenger remains overnight, and the second one is to minimize the flying time or the route length. The GAP arises as a subproblem in solving the Vehicule Routing Problem (VRP), and Fisher and Jaikumar (1981) proposed a generalized assignment approach for it. They first determined seed positions, then they solved GAP to find the assignment of cities to seed positions. Finally they solved the Traveling Salesman Problem (TSP) for each set of cities.

Later, Barker and Sheasby (1999) extended Fisher and Jaikumar's approach, by combining it with a Lagrangian heuristic to guarantee near optimal GAP solutions. DeMaio and Roveda (1971) and Srinivasan and Thompson (1973) considered the Traveling Problem (TP), that is one on the variant of the GAP, as a special case of the Transportation Problem and have proposed implicit enumeration algorithms for this problem.

In the Transportation Problem, each demand point (customer) is required to satisfy

its demand from exactly one resource without exceeding the resource's maximum supply quantity. The objective function is to minimize the total cost associated with the material transportation from the assigned resources to the demand points.

1.4.3 Telecommunication

Bressoud et al (2003) introduced applications for Border Gateway Protocol (BGP) that play a crucial role in the control of transit traffic flows from customers and providers. They have addressed the optimization problem of the routing traffic cost and noted that the single Egress variant problem is similar to the GAP, when egress edge links and prefixes are regarded as machines and jobs.

Another telecommunication application, introduced by Barbas and Marin (2004), is the maximal covering code multiplexing access telecommunication networks with capacity constraints. The goal is to find the optimal assignment of terminals to base stations.

Battiti, Bertossi and Brunato (2001) mentioned the problem of Cellular Channel Assignment and introduced a new localized and distributed strategy that improves the search time for the global minimum of the objective function previously solved by search heuristics.

Mireille et al. (2006) presented a frequency Assignment Problem (FAP) in a Hertzian telecommunication network and defined a new method for frequency assignment with cumulative interference constraints. This new approach considers the case of cumulative disruptions that are modeled through a unique non-binary constraint. They proposed extensions of classical integer linear programming formulations. For a set of realistic instances, they proposed hybrid constraint programming and large neighborhood search solution methods to solve minimum interference and minimum span frequency assignment problems.

Kyung Hee, Jang Hyun and Che Soong (2008) studied the Zone-based Registration (ZBR) as a location registration scheme. They suggested a new model based on cell-by-cell location registration area rather than the ring-by-ring location registration area in order to determine the optimal size of the location registration area that minimizes the signaling traffic on radio channels.

CSA Problem

The cell-to-switch assignment (CSA) problem, similar to Location Area Planning (LAP) problem (see Section 2.5), is the task of grouping cells into Location Areas (LA) during switch level planning of a cellular wireless network, and it's an NP-Hard problem.
Merchant and Sengupta (1995) used a meta-heuristic technique (termed H-I) to solve the CSA problem. The H-I algorithm had two phases. In the first phase, the cells were ordered in a descending order of call volume, and feasible cell assignments were generated. In the second phase of the algorithm, the assignment of each cell was changed in order to obtain a feasible assignment with improved cost.
Hedible et al. (2003) proposed a genetic algorithm (GA) to solve the CSA problem. Their resolution gave better results than the one generated by Merchant and Senguptas' heuristics (1995). In the other hand, Amoussou et al. (2001,2003) developed an exact algorithm that can be used to solve small-scale problems (e.g. 50 cells and 4 switches).
André et al. (2002) improved this exact method by reducing its execution time and have also produced another algorithm which hybridizes their own tabu search implementation with constraints programming. This algorithm produced results that were close to the optimum.
Pierre and Houeto (2002) adopted a tabu search (TS) approach for solving the CSA problem. Their technique starts with a solution obtained by assigning each cell to the closest switchs, ignoring any feasibility constraints. The algorithm didn't guarantee the feasibility of the first solution, but performed a neighborhood search to improve the solution quality till a sub-optimal solution was found.
Demirkol et al. (2001) solved the CSA problem by using a simulated annealing (SA) algorithm. The SA algorithm started with an initial feasible solution. From this point in the search space, a random move was made in the neighborhoods of the initial point. If the move introduced a better point (lower cost), it was accepted. If the move introduced a worse point (higher cost), it was accepted with a decreasing probability over time. The algorithm terminates when a fixed number of non-improving iterations are realized in a predefined number of temperature decrease.
Fournier and Pierre (2005) adopted a new ant colony optimization meta-heuristic to this problem. The objective is to minimize implementation and operational costs: the link

cost linking cells to switches and the handoff cost between cells.
Mandal et al. (2007) applied a new algorithm called the Block Depth First Search (BDFS) to solve the CSA problem, in addition to a lower bound heuristics. For this problem, the BDFS takes as input the number of cells and finds the number of LAs required for grouping these cells, in order to optimize the registration cost. The algorithm BDFS, used in this search process, is similar to two well-known search algorithms, namely A^* and IDA^*.
Saha, Bhattacharjee and Mukherjee (2007) proposed an heuristic to solve the CSA problem for the quasi-static/dynamic LA design, then, they modified the CSA problem by including the load balancing factor among switches and developed the CSA algorithm with load balancing (CALB). They concluded that the CALB algorithm is faster and more performer in balancing the traffic among switches than in cost optimization.

1.4.4 Production planning

Lee (1992) considered a single-sided storage rack in a man-on-board automated storage system. The Storage Layout Problem consists in determining the requested items during a planning horizon, such that each storage location is assigned to exactly one item and an item i is assigned to a fixed number of locations. The objective function is to minimize the total order picking time per period of the storage machine.
Lee and Kim (1986) studied the Multi-Period Order Selection Problem, which is the problem of finding an assignment of a set of orders to a set of periods with the minimum earliness, tardiness and subcontracting cost. Two sets of capacity constraints are taken into account: capacity machine and tool magazine capacity constraints. The authors proposed a binary programming formulation for the original problem and a Lagrangian Heuristics algorithm as a solution.

1.4.5 Location

Many location problems can be modeled as a GAP. Klastorin (1979) noted how the Maximal Covering Location Problem can be formulated as a GAP. Ross and Soland (1980) discussed the Multi-criteria Public Facility Location problem, where we have to select a subset of m sites to establish public facilities, such that each out of n client is

served by exactly one facility.
The main objective function is to minimize costs, dangerous effects, response times and travel distance.

1.4.6 Supply Chain

Higgins (1999) defined a new problem, which is to find the optimal harvest time interval of a set of farm paddock for a planning horizon. Benjafaar et al (2004) mentioned an other generalization of the GAP, the Demand Partitioning Problem (DPP), in which we have to find the minimum cost assignment of a set of products to a set of suppliers where the demand of each product must be satisfied by exactly one supplier. The objective function is to minimize the inventory holding cost of product i per time period, back-ordering cost of product i per time period and production and transportation costs of products i per suppliers j.
The Multi-Period Single Sourcing Problem tries to optimally assign, a set of customers to warehouses, such that each customer is assigned to one warehouses in each problem. The objective function is to minimize the transportations and the inventory costs.
Jalisi and Cheddad (2000) discussed a new variety of the Third Party Routing Problem, where Kalagnanam et al (2001) formulated the Market Cleaning Problem and its relationship with the GAP, The Multiple Knapsack Problem and the bin Packing problem.

1.5 Solution procedures for the GAP

Because the GAP is an NP-Hard problem, there is no polynomial time algorithm which finds a feasible solution to a reasonable execution time and exact solution approaches can deal only with small size instances.
To overcome the limitation of the exact methods, several heuristics as well as metaheuristics methods was devised. In this section we will present some of the solution approaches for the GAP.

1.5.1 Approximate Solution Approaches

Polynomial Time Approximation Algorithms

Shmoys and Tardos (1993) proposed the first approximation algorithm for the GAP, they defined the GAP as a parallel machine scheduling problem where the objective is to find the minimum cost assignment of each job j to one machine i, with processing time r_{ij} and processing cost C_{ij}, such that the capacity constraint b_i of the machine i is not exceeded.

Greedy Heuristics

Proposed by Martello and Toth (1981,1990), they considered a regret measure f_{ij} of assigning task j to agent i. Their heuristic iteratively consider all unassigned tasks and picks tasks j which has the maximum difference between the smallest and second smallest f_{ij}. This tasks is assigned to an agent for which the regret measure is the minimum. At the end, an improvement scheme by local exchanges is applied to the current solution.
Wilson (1997) proposed a dual based solution algorithm for the GAP. He, first, solved a relaxed version of the problem by ignoring the capacity constraints (1.2). Then, he tried to allocate tasks which violate the capacity constraints to other agents in order to restore feasibility while increasing the total assignment costs as small as possible.

Set Partitioning Heuristics

Proposed by Cattryse et al (1994), Set Partitioning Heuristic is based on column generation techniques which output both lower and upper bounds. Authors formulated the GAP as a Set Partitioning Problem where each column corresponds to a feasible assignment of a subset of tasks to an agent.
They solved the knapsack problem in order to generate column. The coefficients are deduced from the dual variable in the LP relaxation of the Set Partitioning Problem which is obtained by a dual ascent procedure.

Lagrangian Relaxation Based Heuristics

Loreno and Narsico (1996) proposed six relaxation based heuristics using both Lagrangian and surrogate relaxation of the GAP. Both of the relaxations are obtained by relaxing the capacity constraints (1.2). They used a general subgradient heuristic in which they adopted two heuristics:

- A modified Martello and Toth heuristic, where the improvement phase is iteratively repeated until no further improvement is possible.
- Narsico and Lorena (1999) derived tree Lagrangian/surrogate relaxation for the GAP by relaxing capacity constraints (1.2), semi-assignment constraints (1.3) and Lagrangian decomposition constraints.

Haddadi (1999) proposed a Lagrangian decomposition based heuristic. He first added substitution variables y_{ij}, coupling constraints $y_{ij} = r_{ij}x_{ij}$ and the constraints of the classical GAP. In order to solve it in polynomial time, the author decomposed the problem into a set of knapsack and transportation problems.
Haddadi and Ouzia (2001) proposed an other Lagrangian heuristic. They dualized the capacity constraints (1.2) and applied subgradient optimization procedure. In each iteration, they first adopted the heuristic approach of Martello and Toth (1981) to find the initial assignment, then they adopted iterative feasibility and solution improvement heuristics of Mazzola (1989) knapsack coupled with a tree way interchange procedure in order to obtain a feasible improvement solution.
Jeet and Kutanoğlu (2007) proposed a Lagrangian relaxation based on problem search space heuristics, where they combined both iterative search capability of the subgradient optimization and problem space search by perturbation.

Linear Programming Relaxation Based Heuristics

Devised by Trick (1992) proposed a heuristics that iteratively removes redundant variables (variables x_{ij} corresponding to r_{ij} with $r_{ij} \geq b_i$), solves the LP relaxations with the remaining variables, fixes the variables with $x_{ij} = 1$ and updates the capacity and deletes the assigned job. This procedure is repeated until no more redundant variables is found.

Dahl and Foldnes (2006) present tree LP based heuristics for the Multiple Knapsack Problem with Assignment Restrictions applied in the telecommunication fields.

1.5.2 Relaxation

Linear programming Relaxation

The linear programming relaxation of a $0-1$ integer program is the problem that arises to replace the constraint that each variable must be 0 or 1 by a weaker constraint, that each variable belong to the interval $[0, 1]$. Benders and Van Nunen (1983) reported that the LP relaxation bound of the GAP becomes tight, for instances with loose capacity constraints (1.2) and where the number of tasks is larger then the number of agents.
Their approach is to assign split tasks among more then one agents to exactly one agent based on heuristic rules.

Lagrangian Relaxation

Was first introduced by Held and Karp (1971), Lagrangian relaxation is a relaxation technique which works by moving constraints into the objective functions as to exact penalize if they are not satisfied.
There is two possible Lagrangian relaxation bounds depending on the constraints set dualizes. Both capacity constraint (1.2) and semi-assignment constraint (1.3) can be dualized. When the capacity constraints (1.2) are dualized, the capacity matrix of the relaxed problem consists of the semi-assignment constraint (1.3), in that case, the capacity matrix becomes totally unimodular and hence the Lagrangian relaxation bound will not be better then the bound obtained by Linear Programming Relaxation.
Ross and Soland (1975) deleted the capacity constraints (1.2) to obtain a lower bound. Then they computed minimum penalties which were got by the reassignment of tasks in order to satisfy the capacity constraints (1.2). Finally, the lower bound is improved by the addition of these penalties.
Chalmet and Gelders (1976) and later Fisher et al (1986) obtained lower bounds for the GAP by dualizing the semi-assignment constraints (1.3). The Lagrangian relaxations bounds obtained by dualizing the semi-assignment constraints (1.3) are as good as the Lagrangian relaxations bounds obtained by dualizing the capacity constraints (1.2).

Lagrangian Decomposition

Jörnsten and Nasberg (1986) developed a decomposition approach where they reformulated the GAP by introducing new decision variables (i.e.,substitution variables) y_{ij}, and new constraints (i.e.,coupling constraints) $y_{ij} = X_{ij}$. The problem is then separated into knapsack problems and semi assignment problems.
Authors shown that the bounds obtained by Lagrangian Decomposition is as good as the bounds obtained by the Lagrangian Relaxation. Barcia and Jörnsten noted that their approach has tightened the bounds obtained with the Lagrangian Decomposition.

1.5.3 Metaheuristic Solution Approaches

Tabu Search

The Tabu search is a metaheuristic algorithm that can be used to solve combinatorial optimization problems. Tabu search uses a local or neighborhood search procedure to iteratively move from a solution x to a solution x' in $N^*(x)$ the neighborhood of x, until some stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure, tabu search modifies the neighborhood structure of each solution as the search progresses.
Tabu Search schemes for the GAP was developed by Díaz and Fernández (2001) and by Osman (1995). A dynamic Tabu Search algorithm was proposed by Higgins (2001). Yagiura et al (2004) developed an ejection chain problem embedded with a Tabu Search, later they adopted the Path Relinking Approach in their ejection chain algorithm in which two phases were proposed by Alfandari et al (2003):

- Phase1 : A population of solution is obtained by solving the linear programming of the GAP, adding cut and applying local search.

- Phase 2: They applied a Path Relinking algorithm which is a generalization of scatter search to perform the combination of elite solutions.

Simulated annealing

It is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It is often used when the search space is discrete. For certain problems, simulated annealing may be more effective than exhaustive enumeration provided that the goal is to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

The method was independently described by Kirkpatrick, Gelatt and Vecchi (1983), and by Cerný (1985). The method is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by Metropolis et al (1953).

Simulation annealing (SA) can be viewed as a process which, given a neighborhood structure, attempts to move from the current solution to one of its neighbors. SAs major advantage over the pure local search method is the ability to avoid becoming trapped in a local solution. For minimization problems, a typical SA procedure is:

```
Let x be an initial solution
Select a value for an initial temperature T_1
k = 0
Repeat
  Repeat
    Generate a neighborhood solution y of x
    If cos(y) < cos(x), then x := y (downhill move)
    Else if random(0,1) < exp((cos(y) - cos(x))/T_k), then x := y (uphill move)
  Until equilibrium is approached sufficiently
  k = k + 1
Until stopping criteria is satisfied
```

To apply the SA method to a specific problem, a variety of decisions have to be made that are usually classified into two factors: problem-specific and generic. The former contains the initial solutions and neighborhood solution structure. Several parameters

related to the cooling schedule such as initial temperature, temperature decrement rule, the number of iterations at each temperature, and stopping rule, are included in the latter.
To attempt reduction of computation time and to enhance solution quality, a variety of methods for setting the SA parameters was proposed in the literature. Most of these methods mainly deal with the careful tuning of cooling schedule parameters.
SA for the GAP were proposed by Cattrysse (1990) and Osman (1995).

Genetic Algorithm

First pioneered by John Holland in the 60s, Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering world. Recently a constructive genetic algorithm was devised by Lorena et al (2005) for the GAP.
Genetic Algorithms (GA) are adaptive of heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GA is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest.
A GA takes population of solutions and produces new ones by using genetic operation. The newly produced solution are evaluated and good solutions are selection for reproduction. The algorithm stopped when there is no better solution or if the time limit exceeds.
For the applications of the GAP, we refers to the work of Chu and Beasley (1997) and Wilson (1997).

Neural Networks

The term neural network (NN) had been used to refer to a network or circuit of biological neurons. The modern usage of the term refers to artificial neural networks, which are composed of artificial neurons or nodes. NN are mainly adaptive learning processes that continuously update some weight until a feasible solution is reached.
Li and Luyuan (1991) was the first whose solved the GAP using competition based neural network. They defined and $m * n$ matrix, where it exists a neuron at each position which compete to become active. Monfared and Etemadi(2004) proposed an other appli-

cation in which they tried four different methods to structure the energy function of the neural network: exterior penalty function method, augmented Lagrangian method, dual Lagrangian method and interior penalty function method. The augmented Lagrangian method was the better one.

Ant System and Grasp

The Ant colony optimization algorithm, introduced by Marco Dorigo (1992), is a probabilistic technique for solving computational problems which can be reduced in finding good paths through graphs. They were inspired by the behavior of ants in finding paths from the colony to food.
Loureno and Serra (2002) proposed a new metaheuristics based on hybrid approach that is the MIN-MAX Ant System which is an improvement of the Ant System. They have also proposed a Greedy Randomized Adaptive Search Heuristic (Grasp) which is a search of neighbors algorithm that applies a local improvement scheme a lot of times, with different starting feasible solution generated randomly.

Variable Depth Search Heuristics

Inspired by the well-known Lin and Kernighan (1973) algorithm devised for the TSP, Amini and Racer (1994) developed a Variable Depth Search Heuristics for the GAP. Their approach yield a lower solution costs then the ones obtained with the Greedy Heuristics of Martello and Toth (1990).
In order to improve the computation time, Amini and Racer (1995) have proposed a hybrid heuristic which takes into account both the speed advantage of Martello and Toth (1990) approximate algorithm and the accuracy advantage of the Variable Depth Search Heuristics.
Amini et al (1998) performed a sensitivity analysis to investigate the robustness of the Variable Depth Search Heuristics.

1.5.4 Exact Approaches

Polyhedral Analysis

The polyhedral analysis of the GAP is very important for other combinatorial optimization algorithm that contain it. Gottlieb and Rao (1990) derived several valid inequalities for the GAP polyhedron based on multiple knapsack constraints.
Farias et al(2000) proposed three families of facet inducing inequalities for the GAP with the special ordered sets, and devised a Branch and Bound algorithm for its solution.

Branch and Price and Branch and Cut algorithms

Branch and cut is a method of combinatorial optimization for solving integer linear programs, which is a linear programming problems, where some or all the unknowns are restricted to integer values. The algorithm is an hybridization of branch and bound and cutting plane methods.
Savelsberg (1997) proposed a Branch and Price algorithm for the GAP. Another branch and price algorithm was implemented by Nemhauser et al (1994) using MINTO - Mixed INTeger Optimizer. Pigatti et al (2004) proposed a Branch-and-Cut-and-Price scheme using a stabilization technique to alleviate the convergence difficulties of the column generation.

Branch and bound algorithms

The method was first proposed by Land and Doig (1960) for linear programming, Branch and bound is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded by using upper and lower estimated bounds of the quantity being optimized.
Some of the Branch and Bound algorithm for the GAP was proposed by Ross and Soland (1975), Martello and Toth (1981,1990) and Guignard and Rosenwein (1989).
Cattryse et al (1998) proposed a Branch and Bound algorithm for the maximization case of the GAP, they developed a cutting plane algorithm using a derivative inequalities from constraints (1.2). The minimization case was addressed by Nauss (2003) where

he proposed to increase the lower bound by using linear programming cuts, Lagrangian relaxation, penalties, feasibility based tests and Logical feasibility tests.
The maximization case of the GAP have also been considered by Haddadi and Ouzia (2004), who have employed a breadth-first search strategy in their branch and bound algorithm.

1.6 Conclusion

There exist many of the real-life applications of the GAP, from production planning to supply chain, from telecommunication to facility layout.
In this chapter, we presented some variants, applications and solution procedures of the GAP to have a general idea about the assignment problems, especially in the telecommunication fields. Through this survey, we distinguished alternative GAP with which we will work, the scope and the method which we will adopt to resolve our problem.
We will discuss a new problem namely The "Bi-objective Location area Design (BLD)" that is a Bi-objective assignment problem of cells into location area (LA) in order to minimize, in the first hand, the location update by taking few and small cells in each location area while keeping a reasonable rate of paging and, in the other hand, minimizing the distance costs between BTS and BSC by assigning BTS to BSC and to LAs.
In the next chapter, we'll introduce the wireless architecture in order to have an idea about the scope of our problem and to present the various elements of the networks that will be used to define our problem. In addition, we will introduce the basic LAP problem, providing details on previously applied solution methodologies and the strategy adopted by TUNISIANA for LA Redistribution (TLPM).

Chapter 2

Wireless Network Design

2.1 Introduction

During this chapter, we present the basic stucture and terminology of wireless network, which will be useful for understanding our problem.
GSM network is a very complex standard, but it's probably the price that must be paid to achieve the level of integrated service and quality offered. So, as the number of mobile users continues to increase, bandwidth become the scarcest resource in the GSM network. The most common method to increase the capacity of the network is to reduce the size of cells, thereby increasing the frequency reuse. But with increasing frequency reuse, there are also increase loads on the system. For instance, as the cell size decreases and the number of cells increase, the number of handoffs, when a mobile moves from one cell to another, also increase, consuming valuable bandwidth. This increasing number of cells also complicates the mobile location function.
The solution would be to redefine the border of the areas by assigning cells into Location Areas (LAs). This problem, namely the Location Area planning problem, was first introduced by Markoulidakis and Sykas (1993).
Many heuristics was introduced to efficiently obtain good solutions to the LAP problem. These heuristics include: random generation methods, tabu search algorithms (Desmestichas et al, 1999, 2000), greedy search algorithms (Plehn, 1995), simulated annealing [(Demirkol et al, 2004); (Desmestichas et al, 1999); (Saraydar and Rose, 1998)], genetic algorithms (GA) [(Wang et al, 1998); (Gondim, 1996); (Chan et al, 2000)], and grouping

genetic algorithms [(Wang et al, 1998); (Gondim, 1996); (Vroblefski and Brown, 2006)]. We overview in this chapter the basic terminology of phone cellular network by presenting each of these components and their roles. Then, we introduce the Location area planning (LAP) problem with an overview on the existent literature.
The rest of the chapter is organized as follows. We begin, first, by introducing the different subsystems in GSM network structure describing the relationship between these components and there usefulness. Then, we present the most important concepts for our problem which are: "cell concept", "Paging" and "Location Update" (LU) during section 2.3 and 2.4. Finally we introduce, in section 2.5, the LAP problem with a review on literature and the resolution strategy adopted by TUNISIANA to solve the LAP problem.

2.2 Architecture of the GSM network

A GSM network (John Scourias, 1996) is composed of several functional entities, whose functions and interfaces are specified by: the mobile station (MS), the subscriber identity module (SIM), the base transceiver station (BTS), the base station controller (BSC), the mobile services switching center (MSC), the home location register (HLR), and the visitor location register (VLR).
These components are grouped into three subsystems namely *BSS*, *NSS* and *OMC*. Figures 2.1 and 2.5 provide an overview of the GSM subsystems.

2.2.1 The Base Station Subsystem

The Base Station Subsystem (BSS) is composed with the MS, BTS and BSC.

Mobile station

GSM network contains as many MSs as possible, available in various styles and power classes. In GSM networks we distinguishe between the identity of the subscriber and that of the mobile equipment, through the ID of the phone and the SIM card.
The SIM, a database on the user side, determines the directory number and the calls billed to a subscriber, and it consists of a chip, which the user must insert into the GSM

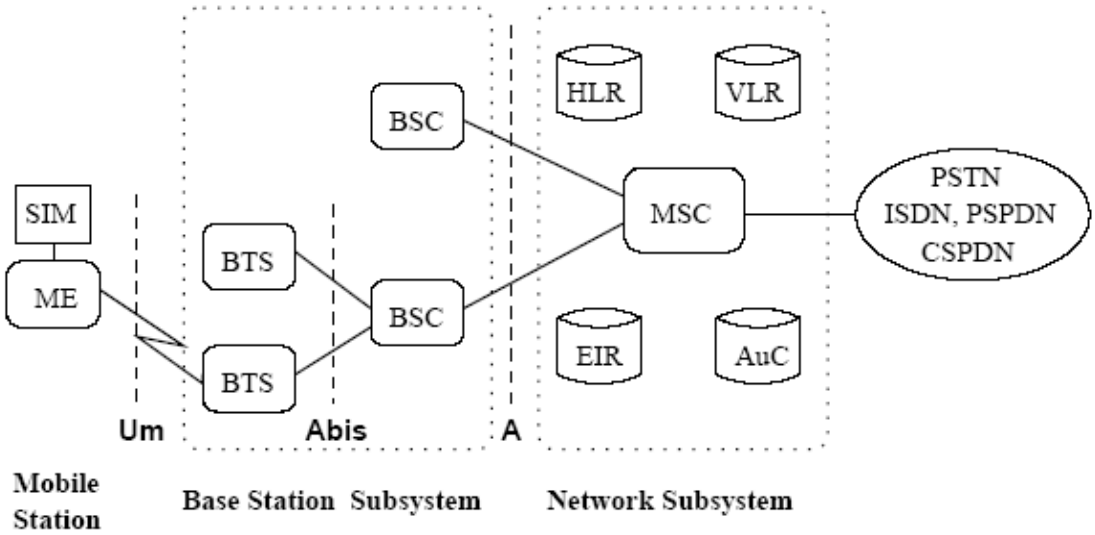

Figure 2.1: Functional Architecture of GSM *[John Scourias, 1996]*

telephone before it can be used. The SIM communicates directly with the VLR and indirectly with the HLR.

BTS

The base transceiver station (BTS) takes care of the radio-related tasks and provides the connectivity between the network and the mobile station via the Air-interface (Um). The BTSs of an area (e.g., the size of a medium-size town) are connected to the BSC via an interface called the Abis-interface. The area covered by a BTS defines a cell.

BSC

The base station controller (BSC) takes care of all the central functions and the controller of the subsystem, referred to as the base station subsystem (BSS). The BSS comprises the BSC itself and the connected BTSs. A large number of BSCs are connected to the MSC via the A-interface.

2.2.2 The Network Subsystem

The Network Subsystem (NSS) comprise MSC, HLR and VLR.

MSC

The mobile service switching center (MSC) is very similar to a regular digital telephone exchange and is accessed by external networks exactly the same way. The major tasks of an MSC are; the routing of incoming and outgoing calls and the assignment of user channels on the A-interface. The MSC is only one subcenter of a GSM network.

HLR

The home location register (HLR), is a repository that stores the data of a large number of subscribers. An HLR can be regarded as a large database that administers the data of literally millions of subscribers. The HLR contains the locations of all the MSs subscribed to the system. Every GSM network requires at least one HLR.

VLR

The visitor location registers VLR is devised so that the HLR would not be overloaded with inquiries on data about its subscribers. Like the HLR, a VLR contains subscriber data, but only part of the data in the HLR and only while the particular subscriber roams in the area for which the VLR is responsible. It maintains the location information of the MS currently within the group of cells managed by it.
The geographic area of the VLR consists of the total area covered by those BTSs that are related to the MSCs for which the VLR provides its services. When the subscriber moves out of the VLR area, the HLR request removal of the data related to a subscriber from the VLR. When searching an MS, the current VLR of the caller will be searched first and then HLR will be looked up if the MS cannot be found in the current VLR.

2.2.3 The Operating Sub-System

The Operating Sub-System (OSS) provides management and oversight of the network. In the Operation and Maintenance Center (OMC), we distinguish OMC/R (Radio) which

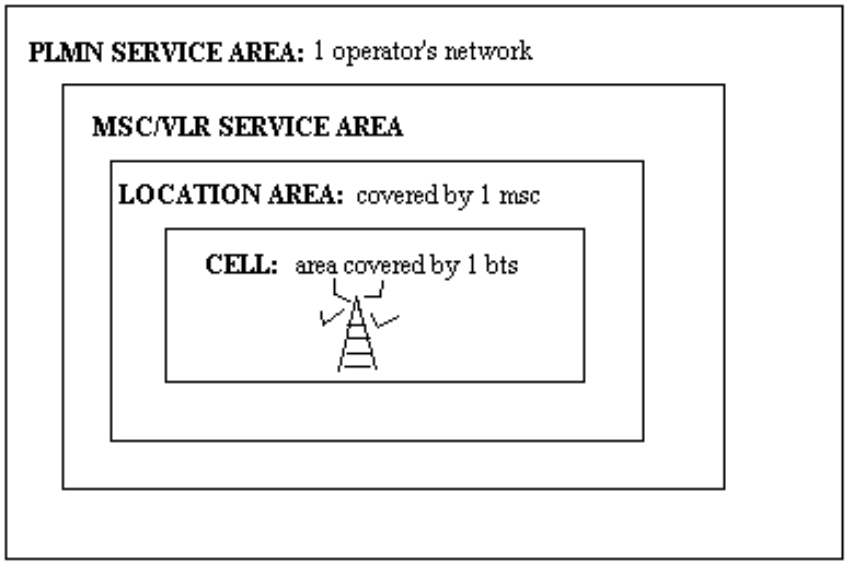

Figure 2.2: Network area

is connected to all entities of the BSS through the BSC, the OMC/S (System) which is connected to subsystem NSS through the MSC. Finally, the OMC/M (Maintenance) that controls the OMC/R and the OMC/S.

2.3 Cells and Location area concept

Cells concept was introduced with the arrival of GSM network ,and now is adapted in 3G networks.

We define a cell as an area covered by one base transceiver station (BTS). The first generation networks had large cells in the center of which stood a base station (antenna). At first, the system allocates a frequency band of statically for each user who was in the cell that needs it or not. This system does provide a service to a number of users equal to the number of frequency bands available. The first improvement was to allocate a channel to a user, only from the point at which it needed thus increase "significantly" the number of subscribers, with the understanding that everyone is not making a call at the same time.

In addition, in order to avoid interference, two adjacent cells can not use the same

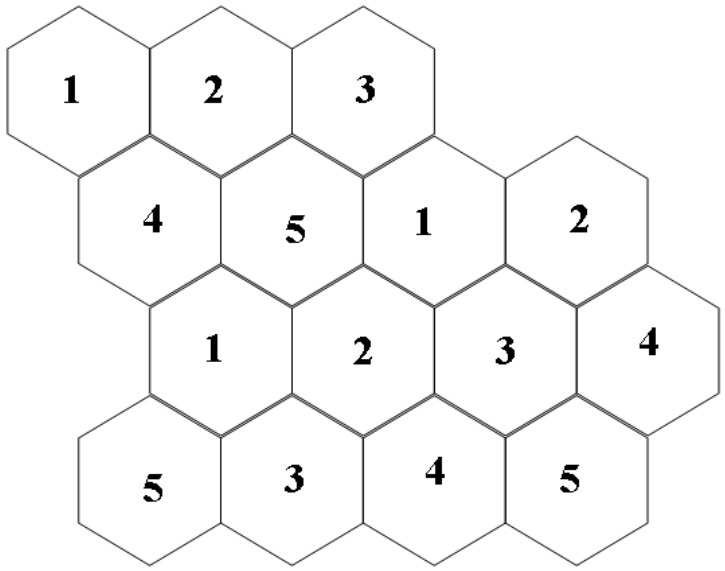

Figure 2.3: Clusters of cells

frequencies. This organization of the network uses the frequency spectrum in a sub-optimal.

The purpose was to divide the territory into small areas called cells and to share radio frequencies between them. Thus, each cell consists of a base station (linked to the switched telephone network) to which combines a number of frequency channels narrow band, summarily appointed frequencies. As before, these frequencies can not be used in adjacent cells to avoid interference (see Figure 2.3).

The location area (LA) consists of a set of adjacent cells (see Figure 2.4) and managed the MSC. It is the area in which the subscriber is paged and each LA is served by one or more base station controllers, yet only by a single MSC.

To enable a user from one cell to another to keep its communication, it is necessary that coverage areas overlap from 10% to 15%, which reinforces the constraint of not having the same band frequency in two adjacent cells.

In summary, a cell is characterized by:

- Its nominal power output (which translates into a coverage area within which the electric field is above a certain threshold)
- The carrier frequency used for radio transmitting and

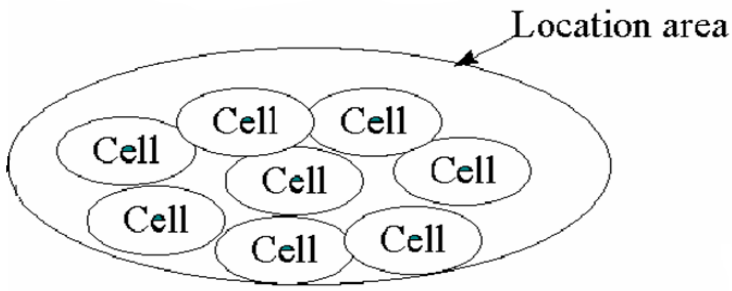

Figure 2.4: Grouping cells into LA

- The network to which it is interconnected.

It should be noted that the cell size is not the same throughout the territory. Indeed, it depends on:

- The number of potential users in the area,
- The configuration of the land (topography geographical presence of buildings,...)
- The nature of structures (houses, buildings, concrete buildings,...) And
- The location (rural, suburban or urban) and thus the density of buildings.

2.4 Network aspects

The transmission of voices or datas over the radio links is part of the function of a cellular mobile network, and the fact that the location area is divided into cells necessitates the implementation of a handover mechanism. Also, because the mobile can move in GSM network, this requires that registration and location updating functions exist in the GSM network.
GSM network ensure three major functions that are: Radio Resources Management, Mobility Management and Connection Management.
These functions are handled through signaling protocols between different GSM entities. Let first introduce frame structure in GSM to understand different frames allocated during the process of location registration or paging.

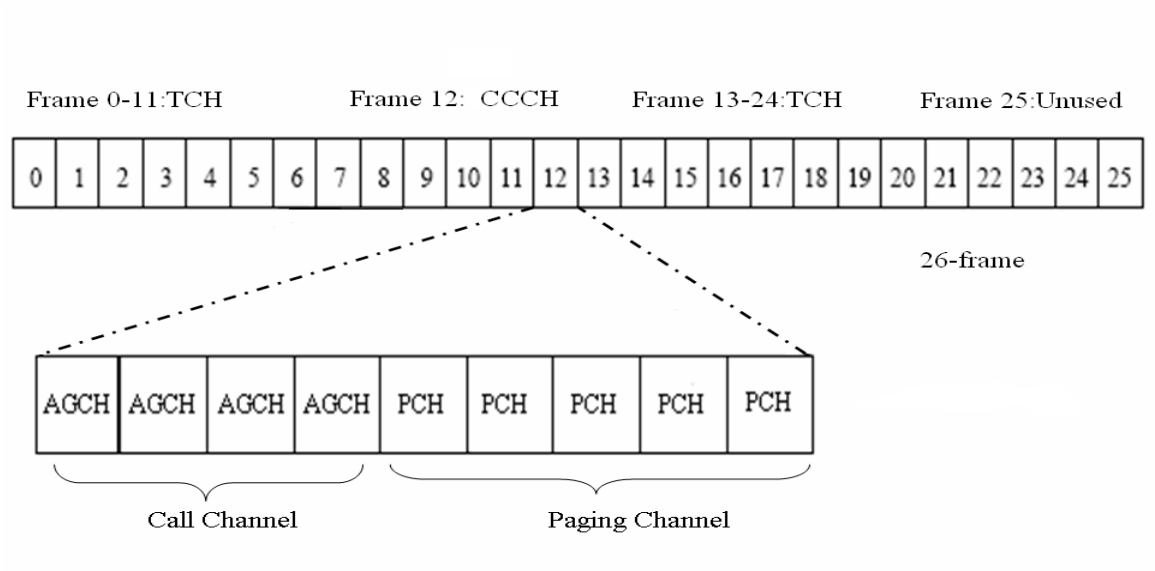

Figure 2.5: TDMA frames and multiframes for speech and data

2.4.1 GSM frame structure

In GSM network, the operator adopted a combination between Time and Frequency-Division Multiple Access (TDMA/FDMA) to involve the division by frequency. One or more carrier frequencies are assigned to each base station. Each of these carrier frequencies is then divided in time, using a TDMA scheme. There exist two types of channels:

- **Traffic channels (TCH)**: used to carry speech and data traffic and are defined using a 26 TDMA-frame, a multi-frame to carry TCH, Slow Associated Control Channel (SACCH) and Fast Associated Control Channel (FACCH). Out of the 26 frames, 24 are used for traffic, one is used for the Slow Associated Control Channel (SACCH) and one is currently unused (see Figure 2.5).

- **Control channels**: can be accessed both by idle mode and dedicated mode. Defined within 51 TDMA-multi-frame and is used to carry Broadcast Channels (BCH), Common Control Channels (CCCH) (Paging Channel (PCH),..), Stand alone Dedicated Control Channel (SDCCH) and SACCH.

2.4.2 Radio Resources Management

Radio Resources Management (RR), is the first and most important function of the cellular mobile network. It Controls the setup, maintenance, and termination of radio and fixed channels, including handovers. The RR layer oversees the establishment of a link between the mobile station and the MSC. The main functional components involved are the MS and the BSS.

Handover

In a cellular network, the radio and fixed links are not permanently allocated for the duration of a call. During a communication, an MS can be in moving (as seen in Figure 2.6), so it's important to guarantee the continuity of the communication outside the cell where the conversation began. In fact, when the station where the MS is attached can't insure a good service quality, so it will be necessary to change the Base Station (BTS). The idea in Handover implies the existence of covering area where many frequence of channel exists from different BTS. Handover, or handoff, is the switching of an on-going call to a different channel or cell.
There are four different types of handover in the GSM system, which involve transferring a call between:

- *inter-Cells*: channels (time slots) in the same cell,
- *inter-BTS*: cells (Base Transceiver Stations) under the control of the same Base Station Controller (BSC),
- *inter-BSC*: cells under the control of different BSCs, but belonging to the same Mobile services Switching Center (MSC), and
- *inter-MSC*: cells controlled by different MSCs.

2.4.3 Mobility Management

The second function of the cellular mobile network, Mobility management, consists of two main procedures: the Location registration (LR) or Location Update (LU) and paging.

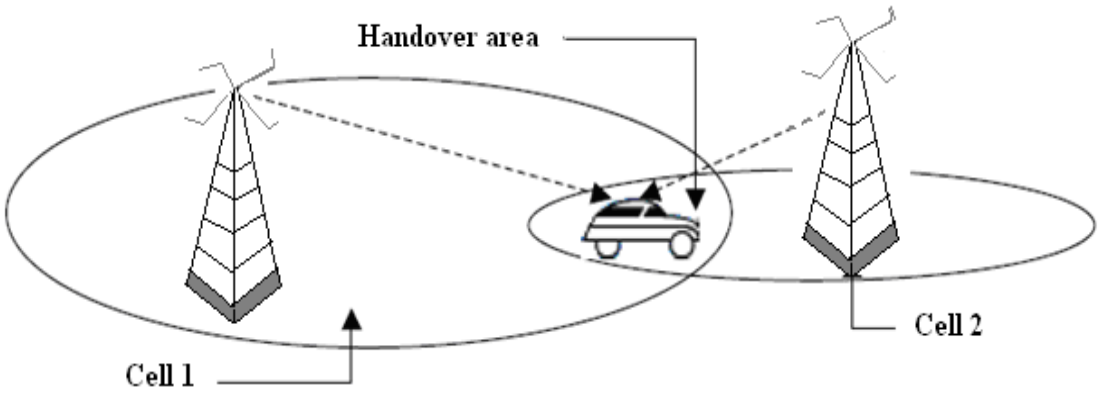

Figure 2.6: Handover area inter-cells

Paging

In cellular communication systems, on the arrival of a mobile-terminated call, the system tries to find the mobile terminal by searching it among a set of base stations (BSs) using the current region knowledge of the mobile (VLR). This search when an MS is paged, is the paging message. It is sent from the MSC to each Base Station Controller (BSC) belonging to that MSCs service area (global paging), or to those BSCs serving at least one cell belonging to the LA where the MS is registered (local paging). For each paging message received by the BSC, Paging Command messages have to be sent to all cells belonging to the LA where the target MS is registered. This paging process is summarized in Figure 2.7.
The paging message is a set of requests sent to each MS. They are generally three types of paging:

- Paging Load: equal to the number of paging request sent, divided by the total number of paging requests by LA (in our case the maximum is equal to 229787 requests for a given LA for one hour).
- Paging Success: equal to the number of paging request received after the first attempt divided by the total number of paging requests by LA.
- Paging retry: equal to the number of paging request received after a maximum of

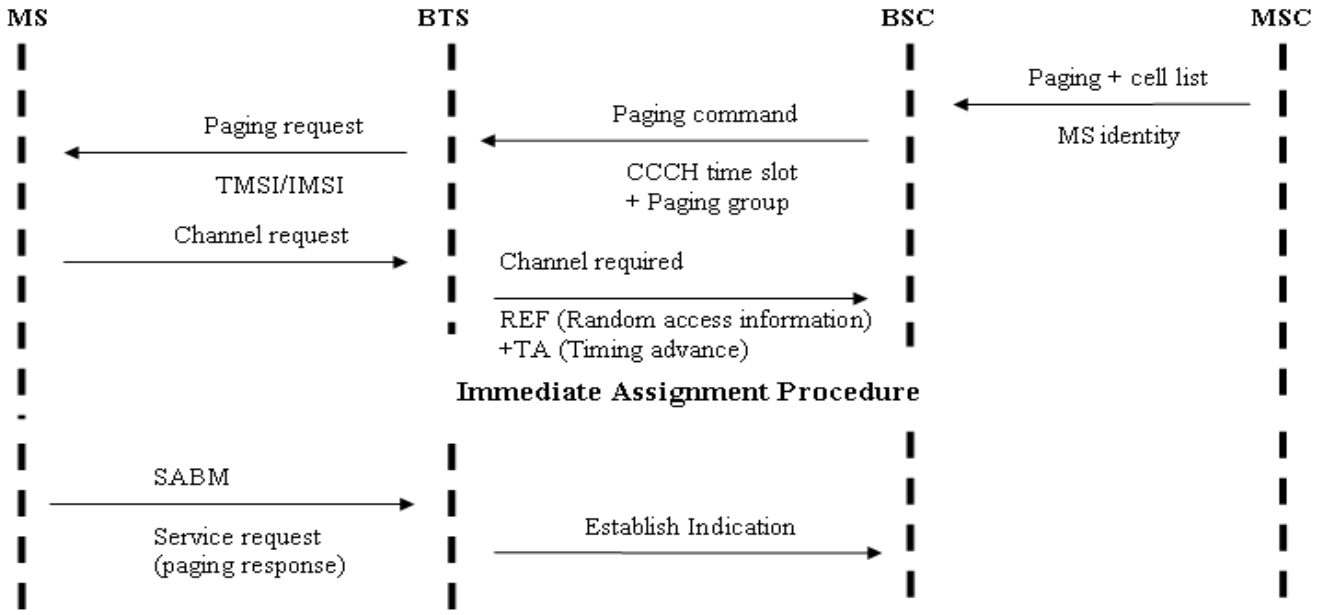

Figure 2.7: Paging Procedure in GSM network *[Online's TUNISIANA Document(OTT)]*

three attempts, divided by the total number of paging requests by LA.

Base Transceiver Station (BTS) has to broadcast all the incoming pages. The paging request messages (DuPont, 1998) are sent via the Paging CHannel (PCH) on the Common Control CHannel (CCCH). Too large LAs may lead to a too high paging load in the BTS resulting in congestion and lost pages. In the other hand, Smaller LAs reduce the paging load in the BTSs as well as in the BSCs. However, smaller LAs also mean a larger number of LA border cells in the network.

Location updating or location registration

The location updating procedures implies the use of the MSC and the Home and Visitor location registers (HLR and VLR). When a mobile station is switched in a new location area, or it moves to a new location area or different operators, it must register with the network to indicate its current location. In the normal case, a location update message is sent to the new MSC/VLR, which records the location area information, and then sends the location information to the subscriber's HLR. The location update affect the load on the signaling subchannels, SDCCH, in the LA border cells.

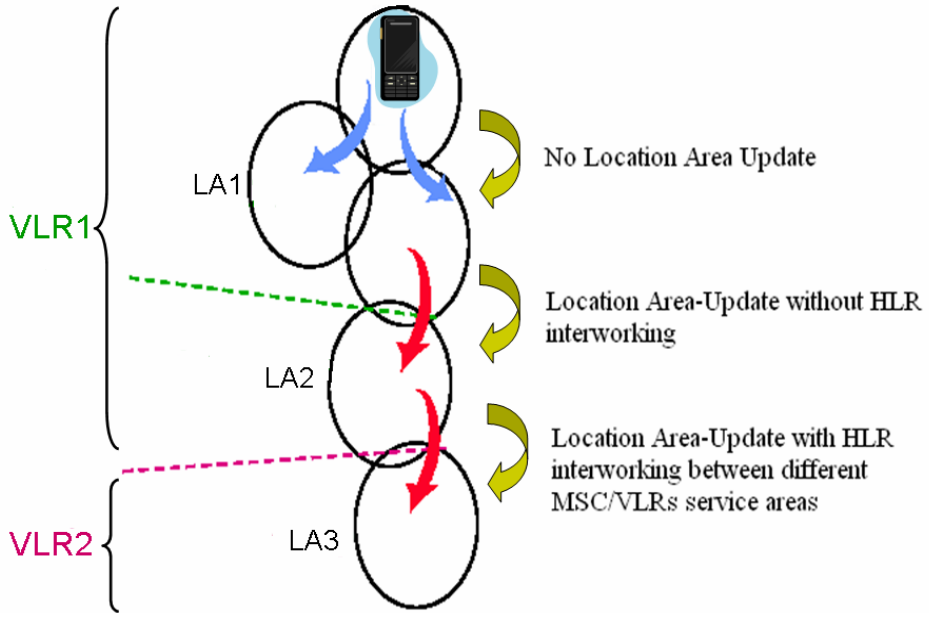

Figure 2.8: Location area update types *[Online's TUNISIANA Document(OTT)]*

There is two types of LU (see Figure 2.8):

❶ LU with HLR interworking: If the mobile changes the location area zone and a VLR service area (i.e. the two location area zones belongs to two different MSC/VLR service areas), the resulting LU is called "with HLR interworking since the HLR has to be informed of the new VLR ID that controls the mobile.

❷ LU without HLR interworking: If a mobile changes a location area zone without changing the VLR service area (i.e. the two location area zones belongs to the same MSC/VLR service area), the resulting LU is called "without HLR interworking since the HLR is not notified of such a transaction. The LU without HLR interworking has big impact on the paging load.

While the MS is idle, the HLR "has to know" at any time the VLR ID and the VLR has to know at any time the LA.

Figure 2.9 summarizes the whole process of Paging and Location Update when moving between different LAs.

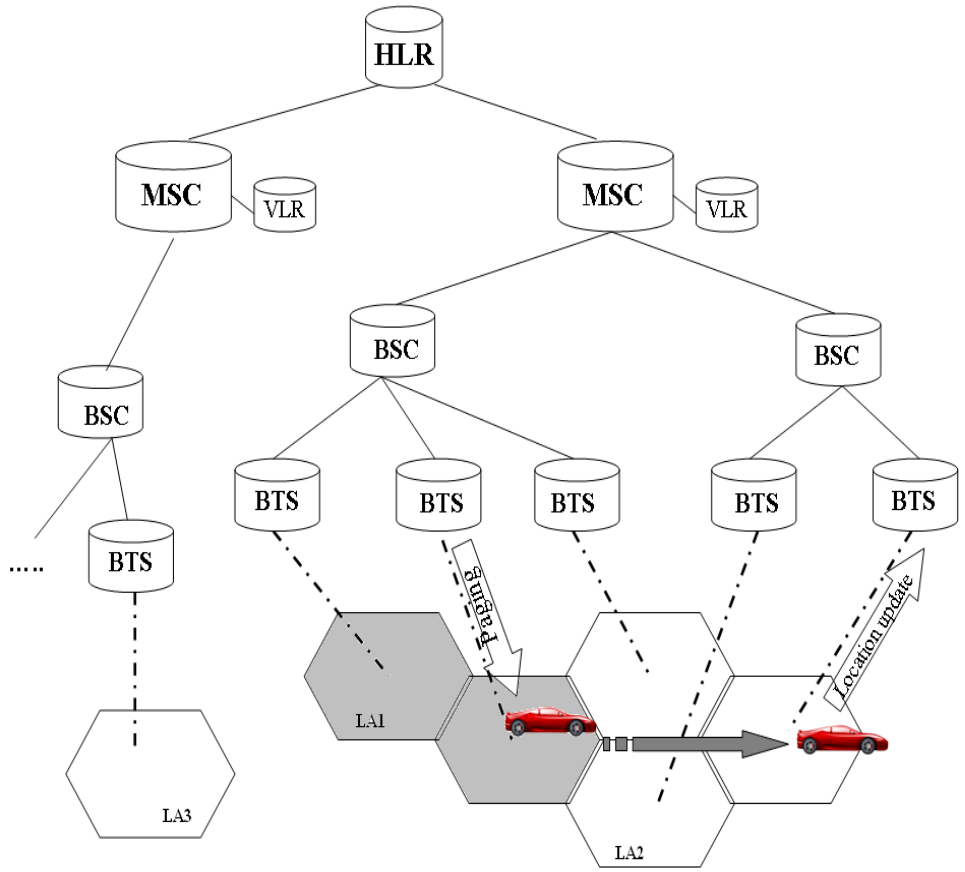

Figure 2.9: Location updates and searching of mobile terminals in a GSM Network

2.4.4 Connection Management

The third function made by the cellular mobile networks is the Connection Management. It handles call controls, manages Supplementary Services and the Short Message Service.

2.5 Location Area Planning (LAP) problem in GSM network

During this section we presents the basic LAP problem with an overview of the existant literature of this problem and the strategy adopted by a telephone operator for LA Redistribution.

2.5.1 Problem description

Location area planning (LAP) 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 is 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.
There is two extreme LA planning scenarios for cellular mobile systems:

❶ One LA covers the whole system area: no LU is performed while called mobile terminals (MT) are paged in the whole system area, thus the network will have to bear the total paging signaling.

❷ An LA equals a cell area: the MT location is known to the greatest possible accuracy; paging is reduced to a minimum, that alerts the MT for the incoming call; however, in this case LU is performed every time a MT changes cell leading thus to maximum LU requirements.

These LA planning scenarios provide a clear view on the conflict between the amount of LU and paging (Figure 2.10): As we increase the size of the LAs, the higher is the paging per LA and the lower is the total amount of LUs.
This trade-off is important to fixe the size of the location area and it is also important

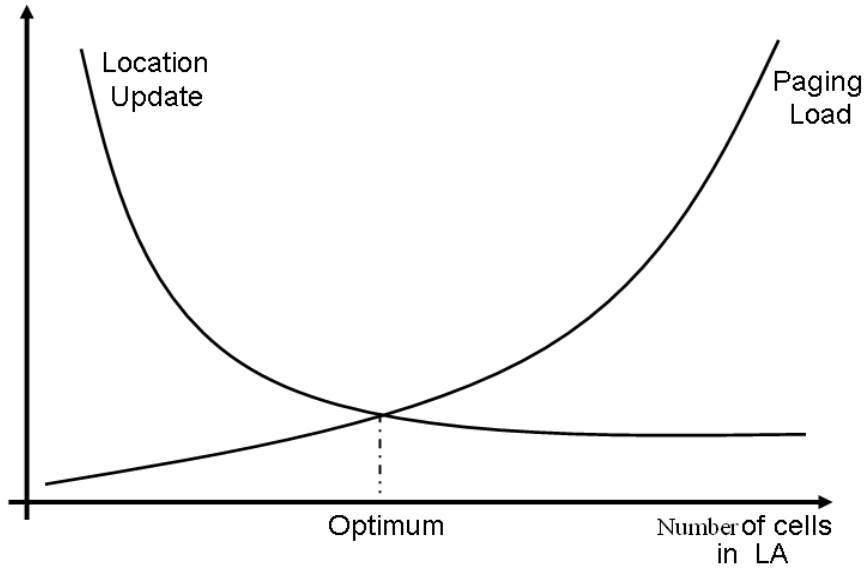

Figure 2.10: Trade-off between LU and Paging load in LAP problem

to reduce the cost of paging and registration signaling. LAP problem has as purpose to minimize the LU costs while fixing a threshold of paging load for new LAs.

Mark Vroblefsk, Evelyn C. Brown(2006) modelled the LAP problem as follows:

$$\textbf{Min } Cost = \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} h_{ij} \tag{2.1}$$

S.t.

$$PB_r - \sum_i x_{ir} page_i \geq 0 \qquad \forall r = 1, .., K \tag{2.2}$$

$$w_{ij} = v_{ij} \qquad \forall v_{ij} = 0 \;\; or \;\; 1 \tag{2.3}$$

$$w_{ij} = 1 - \sum_r x_{ir} x_{jr} \geq 0 \qquad \forall i, j = 1, .., N \tag{2.4}$$

$$\sum_r x_{ir} = 1 \qquad \forall i = 1, .., N \tag{2.5}$$

$$x_{ir} \in \{0, 1\} \qquad \forall i, r \tag{2.6}$$

where they defined $page_i$ as the average number of incoming calls per unit time to cell i, PB_r as the paging bound for LA_r. To represent LU cost, they define h_{ij}, as the average number of mobile phones crossing the boundary between cells i and j per time unit. Eq.(2.1) represents the objective function minimizing LU cost subject to a paging bound constraint for each LA. The paging cost for an LA, in Eq.(2.2) is proportional to the sum of the incoming calls of the cells in that LA and must be less than or equal to the paging bound. Also included in the problem formulation are presets. They used in Eq.(2.3)

presets v_{ij} that are constraints that reflect the network designer s determination that two cells must belong to the same LA or that two cells must belong to different LAs. In Eq.(2.4), w_{ij} takes 1 if cells i and j are assigned to different LAs and 0 otherwise. In Eq.(2.5) they assume that each cell must be affected to only one LA. Eq.(2.6) represents the binary constraints for the decision variables x_{ir} where $x_{ir} = 1$ if cell i is assigned to LA r and 0 otherwise.

In this formulation, Vroblefsk and Brown (2006) didn't take into consideration the distance cost between the BTS and BSC that has an important impact in the management of LAs. They also omit the proximity constraints that is essential in optimizing the traffic between BTS and BSC transmission channels, and thus to better prevent from the lack of resources and the degradation of the signalling quality. By adding these new constraints, we solve the problem on a higher level, allowing a better management of networks.

2.5.2 A literature review on the Location area planning (LAP)

As the problem the LAP problem is NP-complete, it imposes computational limitations on getting optimal solutions for large problem sizes and there is no exact methode that can solve the problem in a polynomial time. Heuristic approaches will be the most suitable methods for this kind of problem.

Over the past 10 years, an abundance of heuristics was introduced to efficiently improve solutions of the problem. The heuristics introduced include random generation methods, tabu search algorithms, greedy search algorithms, simulated annealing, genetic algorithms (GA), and grouping genetic algorithms.

The following Table 2.1 represent a resume of the most important author that treated the LAP problem with a review on the algorithms adopted to solve it, classified by Networks structure.

Markoulidakis and Sykas (1993) were the first to introduce a model based on the tradeoff between the amount of signalling.

Table 2.1: Summary of the literature review on LAP problem

	Network Structure		
Resolution approach	**Two-tier structure**	**Hierarchical structure**	**Main idea**
Tabu search	- Demestichas et al(1999)		(J)
	- P. Demestichas, N. Georgantas, E. Tzifa, V. Demesticha, M. Striki, M. Kilanioti, M. Theologou (2000)		(K)
Simulated annealing	- Demirkol,Ersoy,Caglayan,Delic(2004)		(F)
	- Saraydar and Rose (1998)		(J)
	- Demestichas et al (1999)		(J)
	- Chun-hyun Paik,Samit Soni (2007)		(M)
	- P. Demestichas, N. Georgantas, E. Tzifa, V. Demesticha, M. Striki, M. Kilanioti, M. Theologou (2000)		(K)
Greedy search	- Plehn (1995)		(H)
clustering and greedy method	-Bejerano et al. (2003)		(C)
Set covering algorithm		- Shi-Wu Lo, Tei-Wei Kuo, Kam-Yiu Lam, Guo-Hui Li (2004)	(O)
Genetic algorithm (GA)	- Wang T, Hwang S, Tseng C(1998)		
	- Wang et al(1998)		(J)
	- Gondim (1996)		(J)
	- Demestichas et al (1999)		(j)
	- T.M. Chan, S. Kwong, K.F. Man, K.S. Tang (2002)		(D)
	- P. Demestichas, N. Georgantas, E. Tzifa, V. Demesticha, M. Striki, M. Kilanioti, M. Theologou (2000)		(K)
Grouping GA	- Gondim(1996).Wang,Hwag,Tseng(1998)		(J)
	- Mark Vroblefsk, Evelyn C. Brown(2006)		(E)
Hybrid Grouping GA	- Tabitha James, Mark Vroblefski, Quinton Nottingham (2007)		(L)
Graph theory	- Gamst(1991)		(G)
Hill climbing	-E. Rich and K. Knight(1992)		
Dynamic adjust	- Bejerano and Cidon (2001)		(B)
	- Varsampoulos and Gupta (1999)		(A)
Prediction area		- Bejerano Y, Cidon I(1998)	(P)
Intelligent paging strategies(PSIP)	- Bhattacharjee, Saha, Mukherjee(1999)		(I)
Profil based stategy combined with a built-in memory method		- A. Quintero, O.Garcia,S.Pierre(2004)	(N)

(A)-Varsampoulos and Gupta (1999) proposed to dynamically adjust the coverage of location areas based on mobility patterns of the mobile terminals. Researchers in location planning, in general, agree that the road map and the traffic flows have significant impacts on the mobility patterns.

(B)-Bejerano and Cidon (2001) proposed to dynamically adjust the coverage of LAs by predicting the speed of the mobile terminals on highways. Although these approaches may reduce the network traffic by moving or overlapping the coverage areas of LAs to fit the traffic speed and movement of mobile terminals.

(C)-Bejerano et al. (2003) developed a clustering algorithm for the location area planning problem. Their polynomial time approximation algorithm attempted to minimize LU and paging costs by growing regions around an initial seed node (cell). They then attempted to improve the solution by performing greedy exchanges around the bound-

aries between the neighboring LAs. Their method is shown to perform better compared to other greedy methods. As mentioned previously, this tradeoff between paging and location updating (LU) costs can be mapped to a graph partitioning problem in which the nodes represent cells and the arcs correspond to the traffic passing between cells.

(D)- T.M. Chan et al.(2002) adopted the Genetic Algorithm (GA) with efficient searching technique, to reduce both the weighted-distance costs and hard handoff cost, where they used integer encoding methods. Single and multiple objective approaches of GA are applied and both of them perform in a certain environment.

(E)- Vroblefsk and Brown (2006) developed a Grouping Genetic Algorithm (GGA) for the registration area planning problem. The purpose is to group cells into LA in order to minimize the location update costs while fixing the paging thresholds and preset constraints. Creating groups in GA is a way to deal with drawbacks and enable the evolution of a high quality solutions. They used a cell-based approach for chromosome encoding method.

(F)- James et al.(2007) developed an Hybrid Grouping Genetic Algorithm for the registration area planning problem. This hybridization is accomplished by adding an improvement Tabu Search method to the GGA 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 has better results for a big number of cells.

(G)- Gamst (1991) used a graph theoretical model combined with a greedy method. Although his approach does not uncover a globally optimal solution, it does emphasize the need to focus on those cells with the most LU traffic.

(H)- Plehn (1995), later, proposed a two-phase weighted greedy algorithm to merge two LAs (i.e., groups of cells) in each iteration using the maximum profit to minimize the number of location updates.

(I)- Bhattacharjee et al. (1999) examine two approaches which they call sequential intelligent paging (SIP) and parallel-to sequential intelligent paging (PSIP). They use an occupancy probability vector to determine the cells to be paged. Their solution schemes are able to reduce the paging signaling load, but this reduction comes at the cost of extra processing power.

(J)- Demestichas et al. (1999) performed a tabu search, while a simulated annealing method is examined by Saraydar and Rose (1998), Demestichas et al. (1999), and

Demirkol et al. (2001). GAs was used by Gondim (1996), Wang et al. (1998), and Demestichas et al. (1999).

(K)- Demestichas et al. (2000) defined and solved a new extention of the LAP problem which is a partitioning location area planning. They Started from a formal definition and an optimal formulation of the problem, then three efficient algorithms suitable for application in the network design are presented and compared: Simulated Annealing, Tabu Search, and Genetic Algorithm paradigm.

- Escalle et al.(2002) explored the creation of big-location areas (BLA) consisting of three neighboring LAs. A mobile stores the identification of a BLA in its local memory and LUs are performed when the mobile leaves the BLA. Paging is carried out in a two-step process, where the LA of most recent interaction is paged first. If the MT is not located, the remaining LAs in the BLA are paged.

- Shyu et al. (2006) developed an ant colony optimization algorithm to minimize the hybrid costs associated with handoffs and cabling costs. The clustering of cells to minimize these hybrid costs is often referred to as the cell assignment problem in the literature.

(L)- Demirkol et al. (2004) presented a Simulated Annealing (SA) algorithm to solve both problems of Location area planning (assigning cells to LA) and Cell-to-switch assignment (assigning cells to switches) in cellular networks. They proved that this method give better results than Greedy search, random generation methods and heuristics algorithm to find optimal LAs to minimize LU and Paging costs.

(M)- Chun-hyun Paik and Samit Soni (2007) presented a simulated annealing based solution approach, along with a greedy heuristic is adopted for finding optimal Location and Paging area configuration for mobile networks in order to minimize the location update. They used a two-layered zone-based location registration and paging scheme, in which the costs of location updates and paging signaling traffic are reduced. They considered that the two-layered scheme allows the LA to be divided into multiple Paging Areas (PA), whereas in the single layered scheme the PA of a mobile station is always equal to the LA of the MS.

The previous works on location area planning, mainly assumed the use of the traditional HLR/VLR database structure to manage the location databases. To minimize the workload at the HLR and to improve the system performance, a hierarchical loca-

tion database structure is proposed to organize the location databases in the system. However, it is still an open problem on how to organize the databases into a hierarchical structure with the objective to minimize the update processing cost.
- Li et al. (2004) proposed and summarized methods for location updates over a hierarchical location database structure. They explored the possibility on delaying the location update of mobile terminals, in order to reduce update traffics and database workloads.
(N)- Quintero et al.(2004) reduced the signaling cost during the location update by increasing the intelligence of location procedure. They added a small database to the memory (Intermediate Location Database-ILD) between the VLR and the HLR with a pointer table for each LA (Local Signal Transfer Point-LSTP). This proposed strategy associates to every user a list of cells to determine his moves at each period of time. A *Local list* of a set of location area is stored in the ILD and associated in the LSTP. A *Global list* of a set of which there is LAs that belong to different LSTP is stored in the HLR. Users are divided in three classes: deterministic, quasi-deterministic and random, according to the predictability of their routines.
(O)- Shi-Wu Lo et al.(2004) modeled and resolved the LAP problem as a set-covering problem, in which the relationships between the cells are represented by a graph. The main advantage of this approach is that it can be adapted to the changing mobility of the mobile terminals. They developed the set-covering-based location area planning (SCBLP) algorithm to minimize the total number of location updates, in which the benefit-cost functions are based on the coupling and cohesive functions among neighboring cells.
(P)- Bejerano and Cidon (1998) examine both hierarchical location management scheme as well as an integrated strategy that uses results from traffic flow theory to incorporate location prediction with the location area approach. The concept of using prediction areas to improve mobility management schemes was also explored by Biesterfield and Jobmann (1998).
We can conclude, regarding the summary literature in Table 2.1, that neither PSO approach nor the bi-objective definition of the problem where mentioned. These approaches previously defined, solved the uni-objective problem of assigning cells into LA, trying to minimize LU while fixing a paging boundary. In our case, we will discuss the multi-objective problem of assigning cells into LAs, minimizing both the LU and the distance

cost of BTS-BSC by setting a paging boundery as in the LAP problem and adding other constraints.
Nevertheless, as we will apply our algorithm on a specific uses case, we have to compare our results with those of the telephone operator using existing resolution strategy. Lets,firts, present the strategy adopted by the company.

2.5.3 Tunisiana LAP Model: TLPM

TUNISIANA's company performed the redistibution of cells in LAs by taking into consideration only the uni-objective case based on the Key Performers Identifiers (KPI) criteria namely Qos indicators , represented by the Paging Load (PL) and Paging Retry (PR) (previously presented in section 2.4.3), for the monitoring of a LA.
The division of a LA (or the reallocation of cells) has as purpose to reduce traffic for paging channels (PCH), freeing those for reception and the emission of calls (AGCH), in order to redistribute the resources before any quality of service degradation. As we saw in Figure 2.5, Paging Channels constitute a significant part of 60% from CCCH channels. To avoid extremes, which can lead to a lack of resources and therefore a saturated networks, TUNISIANA launches cutting procedure when the paging increases and reaches at most 50% of paging load rate while the paging success decreases.
As there is no available data to approximate the registration load calculation. The required data is based on mobile flow rate between each cell pair, or after call completion. In other words, The HO attempts could be closely correlated with location updates.
We can see referring to the Flowchart in Figure 2.11, the procedure for Location Area Redistribution in Orascom Telecom (TUNISIANA). If the paging load exceeds 50% load, they lunch the process of LA redefinition. Based on a set of observation of the Qos indicators, the BSS team proposes a set of possible scenario for the redefinition of the LA. For each scenario of cutting, they perform a set of calculations and testing to find the balance between the LU and Paging. If these satisfy all constraints, namely the minimization of paging and HO, so they present it to radio teams for a second study, otherwise they get rid of it.
The most suitable scenario has to satisfy all the engineering rule constraints (geographical constraints, paging, handoff, LU..), neverless, they adopte an other one. After execution,

a long process of monitoring and modification (if needs) stars in order to establish the new shape the LA.
Since this process takes in average one week to get a redistribution of a LA, his automation will be a major concern, because it will save them a lot of time since it is done manually as it is based on statistical studies with complex formulae.
Based on these weakness, we decided to accelerate this process by adopting an appropriate algorithm for a new model, which includes the minimization of the BTS-BSC cost as a second objective, thus alleviating the workload generally required to redefine the LA. This new problem which is "Bi-objective Location area Design (BLD)", will count two major objectives: the minimization of the LU and the distance cost.

2.6 Conclusion

In this chapter we provided an overview on GSM networks system, then we presented the LAP problem with a review on its literature and the redistribution model adopted by Orascom Telecom (TUNISIANA).
In the next chapter, we will be interested by a new extension the LAP problem, namely the BLD problem, which accouts two objective functions to cope with the growth in the number of cells in order to minimize first, the cost of the location update and second the distance cost assignment of BTS to BSC while setting a threshold for paging and satisfying capacity constraints.

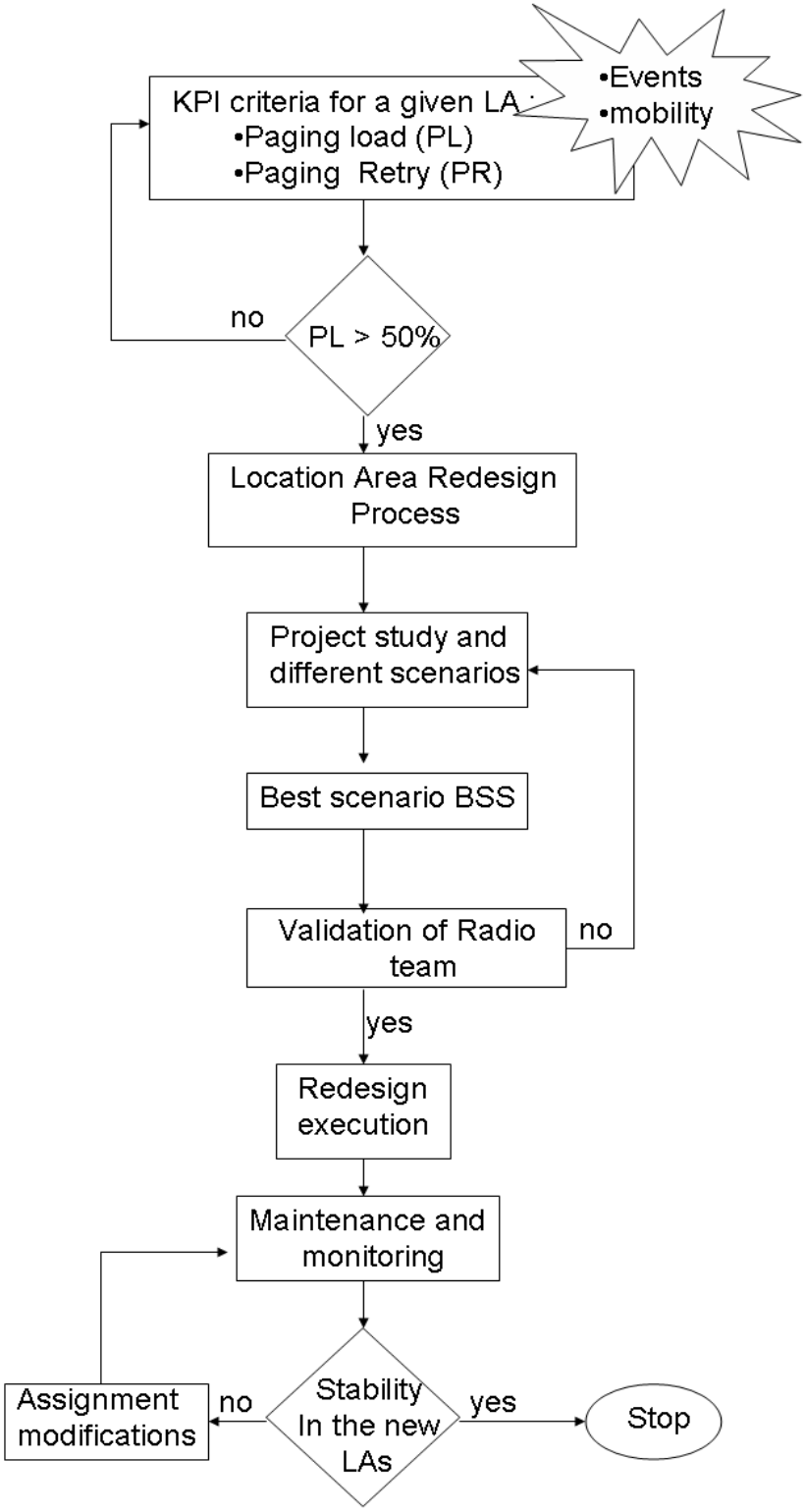

Figure 2.11: Flowchart of TLPM

Chapter 3

A Biobjective Location area planning and connection of antennas in GSM networks: A new solution approach

3.1 Introduction

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 which tends to minimize the LU.

More than the problem of trade off between the amount of Location Update (LU) and paging, and since the allocation of cells is affected by the distance cost between BTS and BSC, we considered a new problem "Biobjective Location Area Planning problem" (BLD) that takes into consideration the distance between BTS and BSC while assigning cells to LAs. So, the simultaneously minimization of LU and distance cost between BTS and BSC can considerably improve the management of the cellular networks and leads to better solutions.

We propose a two-dimensional optimization problem: horizontally, by grouping cells into LAs and vertically, by assigning BTS to BSC.We minimize the location update costs and

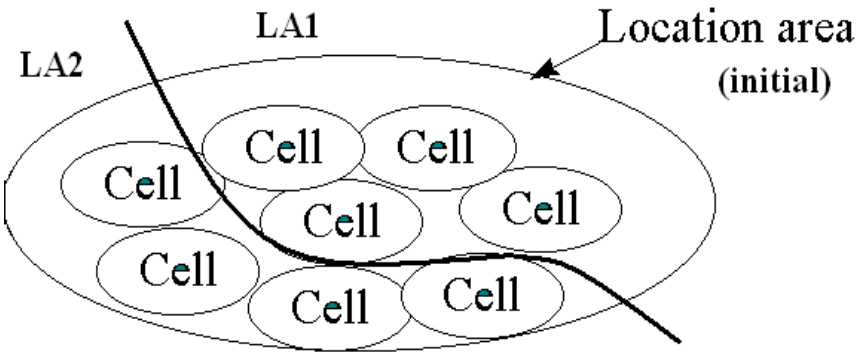

Figure 3.1: Location area division

BTS to BSC distance cost while keeping a reasonable paging rate with adding capacity and proximity constraints.

As the cost of assigning BTS to BSC increases, paging loads in the LAs increase and thus reduces the LU. A high LU traffic is due to small location area zones (The 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, we propose a biobjective model that generates the belongingness of cells to LAs and BTS to BSC.

Being taken into consideration the minimization of both LU and distance cost BTS-BSC to assign cells into LAs, the BLD problem will affect the entire of the BSS subsystem in the GSM networks architecture. This new problem takes into consideration the location area planning (LAP) problem as to minimize the LU under Paging constraints, and the minimization of BTS-BSC distance cost under BSC capacity constraints.

We will consider an existing GSM networks topology, from Orascom Telecom (TUNISIANA), that manage a set of 138 cells, 54 BTS, 3 BSC and 1 MSC in an existing LA. With the increasing number of cells in a LA, we optimize this clustering by redistributing cells into a set of two smallest LAs while satisfying all constraints (see Figure 3.1).

We start by stating the BLD and the network structure that it manages with a mathematical formulation of the new problem as a biobjective nonlinear programming model. As the BLD is an NP-Complete optimization problem, its resolution using an exact method is limited to some size range of the problem. This biobjective model will then

be solved by LINGO for a small examples. Therefore, we propose a resolution strategy accordingly to our problem which satisfies at more our needs. Hence, we Adapted, for the BLD problem, the Vector Evaluated Particle Swarm Optimization (VEPSO) approach which is a multi-objective variant of the PSO algorithm that generates a set of potentially efficient solutions.
It was proven that VEPSO is suitable for multiobjective optimization problems (Hassan.R et al, 2005), produces solutions of high quality with few parameters to adjust and gets better results in a faster and cheaper way compared with other methods.
Besides, it was proven by Venter et al. (2005) that the PSO is more effective than GA in most cases, it may easily cause the swarms to converge to the local area of Pareto Front if the PSO is applied directly to multi objective optimization problems.
Our choice is due especially to:

- the diversification character of this algorithm which is likely to produce solutions of high quality;
- this algorithm manipulates a population of solutions that facilitates the convergence to the required solutions. This is very useful in our case we try to reach more than only one solution;
- this algorithm is likely a method to deal with multi objective optimization problems with few parameters to adjust;
- this algorithm gets better results in a faster and cheaper way compared with other methods.

The rest of the chapter is structured as follows. We state in section 3.2 the mathematical formulation our BLD problem. In section 3.3, we develop the VEPSO heuristic by adapting each component of the model: the structure of a particle, its position and velocity. Finally, we present the implemented algorithm in section 3.4, analyze the results of the computational experiments and investigate the quality of the VEPSO based technique by comparing its results with those of telephone operator.

3.2 Problem formulation

A mobile terminal is informed by an incoming call with a paging message sent over the PAGCH channel of a cell. As a result, a first alternative is to page every cell in the network for each call, which is obviously a waste of radio bandwidth. At the other hand, the mobile could notify the system, via location updating 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 where they affect the load on the signaling subchannels, SDCCH, in the LA border cells, and mobile stations are paged in the cells of their current location area.

A compromise solution used in GSM is to group cells into location areas as to minimize the inter-LA border by reducing the Location update between LAs, and the inter-BSC border by minimizing the cost of assigning BTS to BSC and BTS to LA.

Generally, we subdivide a LA just as it reaches a paging load of 50% in order to optimize the traffic on radio bandwidth and for releasing channels for incoming and outgoing calls. This new allocation must takes into consideration, both proximity constraints of cells, such as cells of the same BTS must be in the same LA, and the constraints of capacity, such as TRX, Traffic and the number of links, because when all cells of a site are connected to a BSC or assigned to a LA, they create that amount of load limit.

The main objective is to cluster the cells into (disjoint) $LAs : LA_1, LA_2,$ (in our case two LAs) such that the paging load represented by our constraints in all LAs does not exceed the thresholds while causing "not too much" location update load between the different LAs (approximated with handover), and not "not too much" distance cost between BTS to BSC and BTS to LA. This task can be formulated as shown in Figure 3.2.

Our BLD problem includes a set of cells C, a set of Base Transceiver Station (BTS), a set of Base Station Controllers (BSC), and one Mobile Switching Center (MSC) with a Two-tier structure hierarchy in the network. While formulating the BLD problem, we only use the available network information and try to include all realistic constraints and objectives.

Note that in our work, 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.

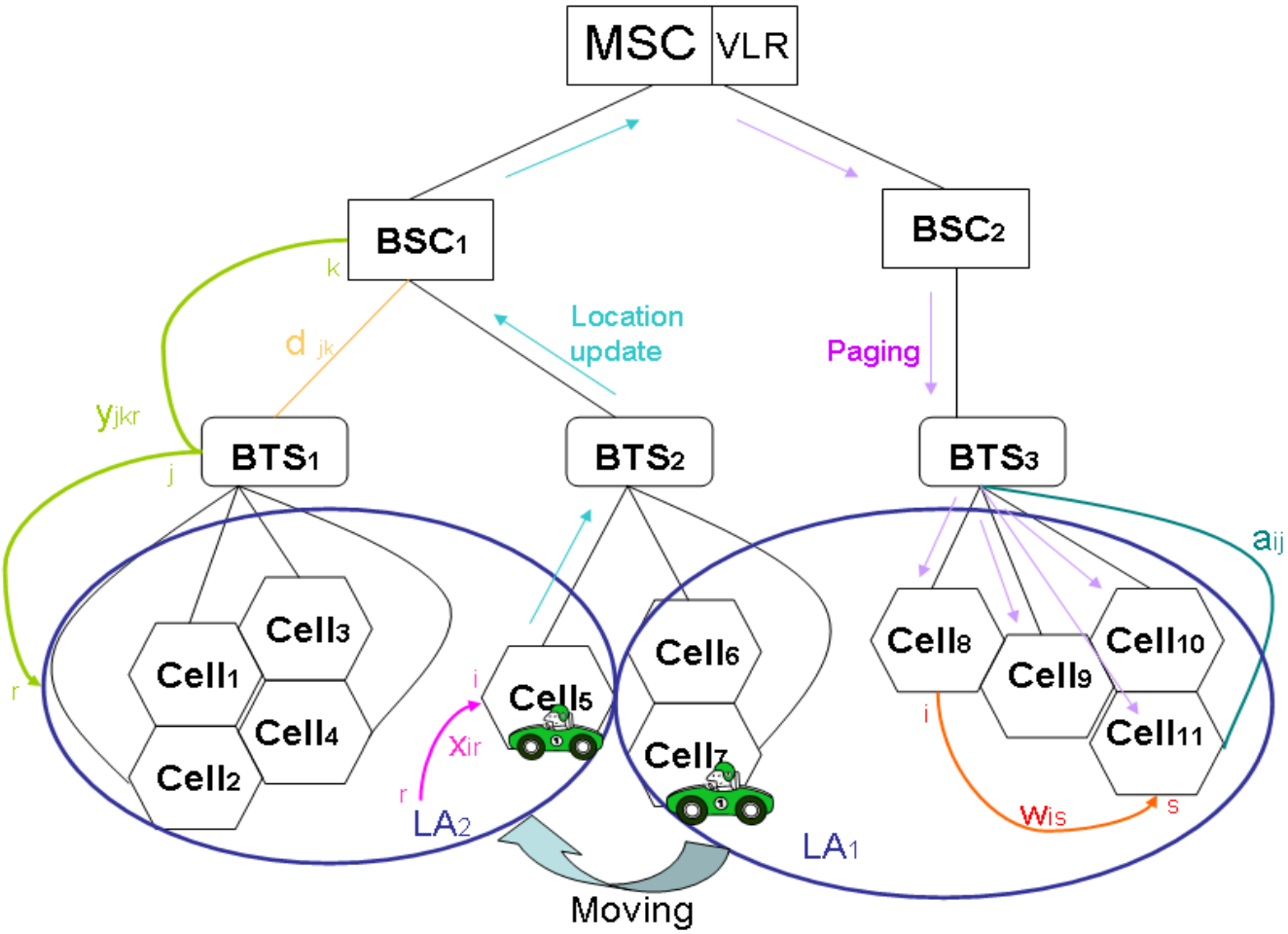

Figure 3.2: BLD problem: "LAs Design"

We presents the mathematical formulation of the BLD problem starting with presenting our variables (inputs and decision variables), then the various constraints and objectives.

❍ Inputs

Note that in our model, each cell should be connected to only one BTS, each BTS to one BSC, each BSC to one MSC and each LA managed by only one MSC. Two cells in the same BTS must be in the same LA. To check those constraints, we use the inputs as follows:

• **Call traffic** and **Number of TRXs** of each cell at busy hour: Which is shown as c_i and r_i for cell i.

• **Cell to BTS connection:** It is modeled with the matrix A with variables a_{ij}, $i = 1, .., N$ and $j = 1, .., Z$ such as:

$$a_{ij} = \begin{cases} 1, & \text{If } i^{th} \text{ cell connected to } j^{th} \text{ BTS} \\ 0, & \text{otherwise.} \end{cases}$$

cell - BTS Topology matrix should have only one entry in the ith row that has value 1 and others must be 0.

• **Handover** (HO) the rate between two cells: to approximate the registration load calculation. The required data is mobile flow rate between each cell pairs, or after call completion. In other words, The HO attempts could be closely correlated with location updates therefore Handover traffic between each cell pair (i,s), is noted by h_{is}. It is important to note that $h_{is} = h_{si}$, i.e. the mobility factor takes into account both the location update traffic generated from mobiles moving from cell i to cell s as well as for mobiles moving from cell s to cell i.

• **Paging loads per cells:** We define λ_i as the average number of incoming requests per unit time to the cell i, and it is equal to the sum of request for paging success divided by the total number of requests for paging load for cell i.

• **Paging capacity per LAs:** Let PB_r define the maximum capacity of paging requests in the rth LA.

• **TRX, Traffic, Links thresholds for each BSC:** Represents the maximum capacity of BSC noted as R_k^{BSC}, C_k^{BSC} and N_k^{BSC} for the kth BSC.

• **Distance cost:** Let d_{jr} be the distance cost between the jth BTS to the kth BSC. Knowing the position of two antennas j and k on a sphere with respective latitudes θ_j and θ_k, and respective longitudes λ_j and λ_k, we compute the curvilinear absciss $S_{(jk)}$. $S_{(jk)}$ is given in radian, then to calculate the distance S in meters, simply multiply it by the mean radius (in our case we multiply it by The radius of the "GRS80" sphere that have R = 6378137m). This value can be calculated as:

$$\begin{aligned} &S_{(jk)}=arccos\ (sin\theta_j.sin\theta_k+cos\theta_j.cos\theta_k.cosd\lambda) \\ &\text{where } d\lambda=\lambda_k\text{-}\lambda_j \\ &\text{Then } S = S_{(jk)}.R \end{aligned} \tag{3.1}$$

❍ Decision variables

The decision variables to be computed by the model are: the assignment of cells to LAs, for the first objective, and the assignment of BTS to BSC and BTS to LAs, for the second objective function.

•**LA-cell topology:** Reported in a matrix X in which:

$$x_{ir} = \begin{cases} 1, & \text{If } i^{th} \text{ cell is located in } r^{th} \text{ LA} \\ 0, & \text{otherwise.} \end{cases}$$

LA - Cell topology matrix should have only one entry in the ith row that has value 1 and others must be 0.

$$\begin{array}{c} \\ cell_1 \\ cell_2 \\ \ldots \\ cell_n \end{array} \begin{array}{c} \begin{array}{cccc} LA_1 & LA_2 & .. & LA_m \end{array} \\ \begin{pmatrix} 0 & 0 & .. & 1 \\ 1 & 0 & .. & 0 \\ .. & .. & .. & .. \\ 0 & 1 & .. & 0 \end{pmatrix} \end{array}$$

W matrix is deduced from X matrix mentioned above where we defined:

$$w_{is} = \begin{cases} 1, & \text{If } i^{th} \text{ cell and } s^{th} \text{ cell resides in different LAs} \\ 0, & \text{otherwise.} \end{cases}$$

- **BTS-BSC-LA topology:** Represented by a matrix Y in which:

$$y_{jkr} = \begin{cases} 1, & \text{If } j^{th} \text{ BTS connected to } k^{th} \text{ BSC and to the } r^{th} \text{ LA} \\ 0, & \text{otherwise.} \end{cases}$$

This topology matrix should have only one entry in the jth row that has value 1 otherwise it takes 0.

❍ Constraints

Our system constraints for the BLD problem are:

- **Allocation constraints:**

  - Each cell should be assigned to exactly one LA. So, for the ith cell, cell-LA topology matrix X should have only one entry in the ith row that has value 1 and others must be 0.

$$\sum_r x_{ir} = 1 \quad \forall i = 1, .., N \quad (3.2)$$

  - Each BTS should be assigned to exactly one BSC and one LA. So, for the jth BTS, BTS-BSC-LA topology matrix Y should have only one entry in the jth row and k^{th} column that has value 1 and others must be 0.

$$\sum_r \sum_k y_{jkr} = 1 \quad \forall j = 1, .., Z \quad (3.3)$$

  - We should have $w_{is} = 0$ if cells i and s are in the same LA and 1 otherwise. Note that this constraints is nonlinear.

$$w_{is} = 1 - \sum_r x_{ir} x_{sr} \quad \forall i \;\; and \;\; s = 1, .., N \quad (3.4)$$

  - We assume that the number of BTS j assigned to LA r is equal to the total number of BTS in the LA.

$$\sum_k y_{jkr} = \sum_j \sum_i a_{ji} x_{ir} \quad \forall j, r \quad (3.5)$$

- Presets v_{ij} are constraints that reflect the network designers determination that the cell i must belong to LA r or not.

$$x_{ir} = v_{ir} \quad \forall v_{ir} = 0 \;\; or \;\; 1 \quad (3.6)$$

- **Paging constraints:** In order to have a feasible cellular network managment, we should not exceed the paging load bound for new LAs (PB_r). The paging load is the number of paging requests divided by the maximum capacity of paging allowed per unit time to LA_r. These constraints ensure that the total number of incoming calls to each LA is below the paging boundery for the two new LAs.

$$PB_r - \sum_i x_{ir} \lambda_i \geq 0 \quad \forall r = 1, .., M \quad (3.7)$$

- **Call traffic capacity:** The maximum call traffic load capacities for BSC k (C_k^{BSC}) is an other constraints that should be obeyed. In our problem, we shouldn't exceed 75% of the maximum traffic load of our BSC.

$$\sum_r \sum_j \sum_i a_{ij} y_{jkr} c_i \leq C_k^{BSC} \quad \forall k = 1, .., W \quad (3.8)$$

- **TRX capacity:** Each BTS has a finite number of Transmitters (TRXs) which defines the number of channels used in that cell. TRX capacity constraint is a vendor specific constraint of BSCs. For each BSC K, the sum of the number of TRXs for each cell connected to that BSC has to be less than fixed thresholds denoted as R_k^{BSC}.

$$\sum_r \sum_j \sum_i a_{ij} y_{jkr} r_i \leq R_k^{BSC} \quad \forall k = 1, .., W \quad (3.9)$$

- **Proximity constraints:** Ensure that if cell i and s belongs to the same BTS j, then they must be assigned to the same LA r.

$$[\sum_j a_{ij} a_{sj}] - [\sum_r x_{ir} x_{sr}] \leq 0 \quad \forall i \;\; and \;\; s = 1, .., N \quad (3.10)$$

- **Links constraints:** Number of BTS j connected to each BSC k should not exceed the maximum capacity of links allowed for this BSC.

$$\sum_r \sum_j y_{jkr} \leq N_k^{BSC} \quad \forall k = 1, .., W \quad (3.11)$$

- **Binary constraints:** Represents the results for the decision variables x_{ir} and y_{jkr}.

$$x_{ir} \in \{0, 1\} \quad \forall i = 1, .., N \;\; and \;\; r = 1, .., M \quad (3.12)$$
$$y_{jkr} \in \{0, 1\} \quad \forall j = 1, .., Z \;\; and \;\; k = 1, ..Z \;\; and \;\; r = 1, .., M \quad (3.13)$$

❍ Objective functions

❶ **the total registration signaling:** We approximate this aggregate mobile flow behavior to the handover rate between two cells i and s to minimize the total LU cost. A LU must be performed when a mobile crosses a border between two LAs, it finds the total number of mobiles crossing into LA borders per unit time.

$$\textbf{Min } Z_1 = \sum_i \sum_s h_{is} w_{is} \quad (3.14)$$

❷ **Minimize the total distance cost between BTS to BSC:** The second objective function in our model consists of minimizing the distance between BTS to BSC, based on the assignment of BTS to LAs and to BSC, in order to improve the quality signal transmission.

$$\textbf{Min } Z_2 = \sum_k \sum_j d_{jk} \sum_r y_{jkr} \quad (3.15)$$

The mathematical formulation of the BLD problem represented through equations (3.2)-(3.15) is a difficult optimization problem in which LA-Cell (X matrix) topology, and BTS-BSC-LA topology (Y matrix) have to be decided, can be stated as follows:

$$\textbf{Min } Z_1 = \sum_i \sum_s h_{is} w_{is} \tag{3.14}$$

$$\textbf{Min } Z_2 = \sum_k \sum_j d_{jk} \sum_r y_{jkr} \tag{3.15}$$

S.t.

$$\sum_r x_{ir} = 1 \qquad \forall i \tag{3.2}$$

$$\sum_r \sum_k y_{jkr} = 1 \qquad \forall j \tag{3.3}$$

$$w_{is} = 1 - \sum_r x_{ir} x_{sr} \qquad \forall i, s \tag{3.4}$$

$$\sum_k y_{jkr} = \sum_j \sum_i a_{ji} x_{ir} \qquad \forall j, r \tag{3.5}$$

$$x_{ir} = v_{ir} \qquad \forall v_{ir} = 0 \;\; or \;\; 1 \tag{3.6}$$

$$PB_r - \sum_i x_{ir} \lambda_i \geq 0 \qquad \forall r \tag{3.7}$$

$$\sum_r \sum_j \sum_i a_{ij} y_{jkr} c_i \leq C_k^{BSC} \qquad \forall k \tag{3.8}$$

$$\sum_r \sum_j \sum_i a_{ij} y_{jkr} r_i \leq R_k^{BSC} \qquad \forall k \tag{3.9}$$

$$[\sum_j a_{ij} a_{sj}] - [\sum_r x_{ir} x_{sr}] \leq 0 \qquad \forall i, s \tag{3.10}$$

$$\sum_r \sum_j y_{jkr} \leq N_k^{BSC} \qquad \forall k \tag{3.11}$$

$$x_{ir} \in \{0, 1\} \qquad \forall i, r \tag{3.12}$$

$$y_{jkr} \in \{0, 1\} \qquad \forall j, k, r \tag{3.13}$$

3.2.1 Cardinality of BLD problem

Let S be the cardinality of the problem, in our case the cardinality of the matrix of the decision variables solutions $|S| = i * r + j * k * r$.

The determining factors of the solution space size are the assignment of BTS to BSC and BTS to LA and the assignment of Cells to LA. If we have n Cells, m BTSs, k BSCs and p LAs, the number of possible BTS-BSC assignments is as follows: for each BTSs there are k possible BSCs to be connected, so we have m^k possible connections. Equally, the number of possible Cell-LA assignments is n^p and the number of possible BTS-LA assignments is m^p, because the upper limit of the number of LAs is p (the number of new LAs shall be fixed from the beginning). Consequently the size of the solution space is found to be $(m^k + m^p).n^p$. We should assess all solutions whether they are feasible or not, and because of the solution space size, exhaustive methods would result in exponential time complexity.

3.2.2 Multi-objective optimization problem

Definition 1: *A solution x of the multi-objective problem is said to be Pareto optimal if and only if there does not exist another solution y, such that $f(y)$ dominates $f(x)$.*
The objective functions $f_i(x)$, may be conflicting with each other, thus, most of the time it is impossible to obtain for all objectives the global minimum at the same point. Instead there exists a set of optimal trade-offs which forms the solution set the Pareto set and it is denoted by P^*. The set $PF^* = f(x)|x \in P^*$ is called the Pareto front.[Teghem.J et al, 2000]
Definition 2: *The set of solutions where no solution is dominated by any other solution is the set of non-dominated solutions. The non-dominated set of the entire feasible search space is the Pareto optimal set or Pareto front.[Teghem.J et al, 2000]*
Definition 3: *The efficient set is the set of solutions of the Pareto front in variable space, whereas the Pareto front is defined in objective function space. Identification of the efficient set for a given facility location problem is especially important because the efficient set are the location alternatives for the facility.[Teghem.J et al, 2000]*
Because there is no unique global best solution in the multi objective problem, to find a solution for the multi objective problem is to find a set of solutions (Pareto Optimal Set).
In Pareto preference (Haluk Yapicioglu et al. 2007), a solution, $X_{(2)}$, is dominated by solution, $X_{(1)}$, if $X_{(1)}$ is not worse than $X_{(2)}$in all objectives, and for at least one of the objectives, $X_{(1)}$ is strictly better than $X_{(2)}$. These conditions can be expressed as follows for the case where, like in our problem, all of the objective functions are to be minimized:

$$\begin{aligned} Z_m(X_{(1)}) &\leq Z_m(X_{(2)}) \quad \text{for } m = 1, 2, .., M, \\ Z_m(X_{(1)}) &< Z_m(X_{(2)}) \quad \text{for some } m \end{aligned}$$

Identification of the efficient set for a given assignment problem is especially important because the efficient set are the assignment alternatives for the cells to LAs.
Traditional adopted method for the multi objective optimization is generally to transform the model in a single objective problem through weighted sum. However, this method requires a priori knowledge of the problem itself, so it can not solve real multi objective problems. Therefore, Evolutionary Algorithms are very suitable for solving multi objective optimization problems.

Particle Swarm Optimization (PSO) is an optimization evolutionary algorithm, It is easy to be understood and realized and was applied in many optimization problems. It was demonstrated that the PSO is more effective than traditional algorithms (Venter et al. 2005) in most cases and the application of PSO in the multi objective optimization problems could be very promising.

3.3 VEPSO approach for the BLD problem

Our aim is to propose an algorithm that will take the required constraints described in Section 3.2 as inputs, and find an optimal or near optimal efficient solutions as a network topology which includes the assignment of cells to location areas (LAs) and BTS to BSC and to LAs.
Due to the difficulty of the problem, it is not possible to guarantee the optimal solution. Therefore, techniques that give near optimal solutions are needed.
We solved our problem with LINGO (see Appendix.A) in order to get an optimal solution for a small size of the problem.
The question "Which metaheuristic is the best?" is not a valid one. Each metaheuristic has its benefits and the question to ask when facing a problem is "Which is the appropriate metaheuristic for that kind of problems?". The answer is given by the different parameters that the problem includes. For example, a constructive method is a well adapted method for a problem that can be represented as a shortest path graph. A genetic algorithm is the appropriate method to use when diversification in solutions is required to get close to the optimal.

3.3.1 Particle swarm optimization

Particle swarm optimization (PSO) (Alberto García Villoria and Rafael Pastor, 2009) is an evolutionary stochastic metaheuristic designed by Kennedy and Eberhart in the mid 1990s and was studied by several researchers since its publication. PSO was successfully applied to a variety of problems such as artificial neural network training [Chau (2006); Geethanjali (2007)], combinatorial optimization problems [Andrés (2004); Liao (2007); Secrest (2001) and Tasgetiren (2007)] and multiobjective optimization problems [Hu and

Eberhart (2002); Yin, Yu, Wang, Wang (2007)]. Most of the PSO applications published in the literature were designed to solve continuous optimization problems, but there are few PSO applications for discrete optimization problems.

The purpose of PSO is based on the analogy of the social behavior of flocks of bird when they search for food. As it is a population-based evolutionary metaheuristic, PSO has a population (known as swarm in the PSO ambit) of particles. Each particle has an associated point in the search space (which represents a solution) and an associated velocity (which indicates how the point of the particle is moved in the search space).

Essentially, each particle adjusts its flying based on the flying experiences of both itself and its companions (personal and global best). During the process, it keeps track of its coordinates in hyperspace which are associated with its previous best fitness solution, and also with the overall best value acquired thus far by any other particle in the population. Every swarm updates continuously its own velocity and position through the above best solutions mentioned. Thus, a new generation appear with a better solution and ultimately converge to an optimal solution.

The current velocity of a particle is typically a linear combination of three types of velocity:

1. the inertia velocity (i.e. its previous velocity) which is the former speed of the swarm.

2. the velocity to the best point found by the particle (The cognition model).

3. the velocity to the best point found by the swarm (The social model).

The particle position and velocity update equations in the simplest form for the PSO are given by:

$$v_{t+1,id} = v_{t,id} + c_1 r_1.(P^*_{t,id} - X_{t,id}) + c_2 r_1.(SP^*_{td} - X_{t,id}) \quad (1)$$

$$X_{t+1,id} = X_{t,id} + v_{t+1,id} \quad (2)$$

Where:

- $i = 1, , N$ and N is the population size, $d = 1, , D$ and D is the dimension of the search space;

- $v_{t,id}$ is the inertia velocity of the particle i at iteration t;

- $X_{t,id}$ is the position of the ith particle at iteration t;
- $P^*_{t,id}$ is the best position found by the particle i up to iteration t;
- SP^*_{td} is the best position found by the swarm up to iteration t;
- r_1 and r_2 are random numbers uniformly distributed in [0,1];
- c_1 and c_2 are the coefficients, called acceleration coefficients, that weight the relevance of the last two types of velocity. The values $c_k r_k (k = 1, 2)$ determine the weights of the two parts, where their sum is usually limited to 4.

The two most popular mechanisms that can be used to control explosion of velocities and provide stability are: *inertia*, proposed by Shi and Eberhart (1998), and *the constriction coefficient*; proposed by Clerc (2002).
Using *inertia factor*, the velocity update formula will be as follows:

$$v_{t+1,id} = \omega v_{t,id} + c_1 r_1.(P^*_{t,id} - X_{t,id}) + c_2 r_1.(SP^*_{td} - X_{t,id}) \quad (3.a)$$

Where ω is the coefficient (called inertia weight) that weights the inertia velocity. This coefficient was introduced by Shi and Eberhart to eliminate the need for velocity clamping, but to restrict divergent behavior. It controls the dynamic of the particle by weighting the contribution of the previous velocity, controlling how much memory of the previous flight direction will influence the new velocity.
The selection of coefficients in the velocity update equations affects the convergence and the ability of the swarm to find the optimum. If the inertia weight of the velocity is small, all particles could slow down until they approach zero velocity at the global best. One way to come out of the situation is to reinitialize the particles positions at intervals or when convergence is detected.
Using *the constriction coefficient*, the velocity is updated using the following equation:

$$v_{t+1,id} = K.[v_{t,id} + c_1 r_1.(P^*_{t,id} - X_{t,id}) + c_2 r_1.(SP^*_{td} - X_{t,id})] \quad (3.b)$$

$$\text{Where } K = \frac{2}{|2-(c_1+c_2)-\sqrt{(c_1+c_2)^2-4(c_1+c_2)}|} \quad \text{s.t. } c_1 + c_2 \geq 4 \quad (3.c)$$

Both of these measures are useful to improve the performance of PSO. For the velocity value of each case in a particle, a constant V_{max} is used to limit the range of $v_{t,id}$ and

controls the maximum travel distance during each iteration to avoid this particle to ignore good solutions. In practice, V_{max} is often set at 4, i.e., $v_{t,id} \in [-V_{max}, +V_{max}]$.

If $v_{t,id} \geq V_{max}$, $v_{t,id} = V_{max}$
If $v_{t,id} \leq V_{max}$, $v_{t,id} = -V_{max}$

As the swarm iterates, the fitness of the global best solution improves. It could happen that all particles being influenced by the global best eventually converge to it, and there where the fitness never improves despite many runs the algorithm. The particle also moves in the search space, in close proximity to the global best, without exploring the rest of search space. This phenomenon is called 'convergence'.
$P^*_{t,id}$ and SP^*_{td} are given by the following equations respectively (for minimizing the objective function):

$$P^*_{t,id} = \begin{cases} P^*_{t,id} : F(X_{t,id}) \leq F(P^*_{t,id}) \\ X_{t,id} : F(X_{t,id}) \geq F(P^*_{t,id}) \end{cases} \tag{4}$$
$$SP^*_{td} \in \{X^*_{t,1d}, X^*_{t,2d}, .., X^*_{t,md}\} | Min(F(X^*_{t,1d}), F(X^*_{t,2d}), .., F(X^*_{t,md}))$$

Where F is the *fitness function* and $m \leq M$, M is the total number of particles.
The pseudo-code of the PSO algorithm is:

For each particle $i = 1, .., n$
Randomly initialize the position and velocity in the population.
Calculate the fitness value.
Calculate $P^*_{t,id}$ and SP^*_{td}.
Do
- Update the position and the velocity, at $t+1$, using equation (1) and (2) where respecting the maximum threshold v_{max}.
- Calculate the fitness value.
- If the current fitness value is better than its $P^*_{t,id}$, Update $P^*_{t,id}$.
- Update SP^*_{td} choosing the position with the best fitness value among all its neighbors as the SP^*_{td} for a specific neighborhood topology.

While Stopping criteria

There exists a lot of strategy for solving the multi-objective problems using PSO. The most common approach is to aggregate all the objective functions with appropriate weights into a single objective function. Hu and Eberhat (2003) proposed a dynamic neighborhood PSO.

During this research, we adoptes an algorithm proposed by Persopoulos et al (2004) namely the Vector Evaluated Particle Swarm Optimization (VEPSO) which is inspired from the Vector Evaluated Genetic Algorithm (VEGA). It was proved by Hassan et al(2005) that the Particle Swarm Optimization algorithm (PSO) is more effective than GA in most cases. In addition, the successful application of genetic algorithms in multi objective optimization problems and the similarity between PSO and genetic algorithms reflects that PSO is likely a method to deal with multi objective optimization problems. However, there is a great distinction between PSO and genetic algorithms (GA). In genetic algorithms, the informations are shared by chromosomes, which cause the whole community move gradually into a better area, while in PSO the information is sent out by the best particle which is followed by other individuals to quickly converge to a point. Therefore, it may easily cause the converges of swarms to the local area of Pareto Front if the PSO is applied directly to multi-objective optimization problems.

In past several years, PSO was successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster and cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. Particle swarm optimization was used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

By relying on the above-mentioned reasons, we propose to adopt the Vector Evaluated Particle Swarm Optimization (VEPSO) algorithm, which is a multi-swarm variant of PSO and inspired by the Vector Evaluated Genetic Algorithm (VEGA), for solving our bi-objective LAP problem (or BLD problem).

3.3.2 Vector evaluated PSO - VEPSO

The algorithm of Parsopoulos et al.(2004), Parallel vector evaluated particle swarm optimization (VEPSO), is a multi-swarm variant of PSO. Inspired from the vector evaluated

genetic algorithm (VEGA)(Schaffer, 1984), and similar to the two single objective functions being evaluated separately by separate swarms.

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 experience. Thus, exchanging theses informations 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 the current problem, as it can search multiple efficient solutions in a vast solution space. His principle key issues is that the fitness of an individual in a population depends on the particles of a different swarms.

Considering M swarms $\{S_1, S_2, ..., S_M\}$ each of size N and M objective functions, each swarm is exclusively evaluate according to one of the objective functions. In our case we have two objective functions so we will use two swarms S_1, S_2. Let $X^{[j]}_{(id,t)}$; $V^{[j]}_{(id,t)}$; $P^{[j]}_{(id,t)}$ and $Pg^{[j]}_t$ be the current position, velocity, the best previous position of the ith particle and the global best in the jth swarm, respectively, at a given time t, and $d \in N$ where N is the size of our particles.

VEPSOs swarms should be updated according to the following equations:

$$V^{[j]}_{(id,t+1)} \leftarrow K^{[j]}.[\omega.V^{[j]}_{(id,t)} + c_p r_1.(P^{[j]}_{(id,t)} - X^{[j]}_{(id,t)}) + c_g r_2.(Pg^{[k]}_t - X^{[j]}_{(id,t)})] \quad (5)$$

$$X^{[j]}_{(id,t+1)} \leftarrow X^{[j]}_{(id,t)} + V^{[j]}_{(id,t+1)} \quad (6)$$

where, as in the PSO algorithm, C_p is the Cognitive learning rate and C_g is the Social learning rate. The factors r_1 and r_2 are randomly generated within the range $\{0, 1\}$, ω is the inertia weight and K is the constriction coefficient calculated as showed in Equation (3.c).

A VEPSO algorithm for discrete variables has been developed by Omkar (2007) for the multi-objective optimized design of composites. The problem is formulated with multiple objectives for minimizing the weight and the total cost of the composite component to achieve a specified strength.

Vlachogiannis and Lee (2009) introduced Parallel VEPSO method to solve multi-objectives problem of power systems in a steady-state. They evaluated reactive power control and formulated the problem as a multi-objective optimization problem.

3.3.3 VEPSO for the BLD

The VEPSO method is here incorporated with the necessary modifications to make it able to include non-linear contraints with discrete design variables. Our algorithm attempt to minimize both LU and distance cost by growing regions around an initial seed cell. We then improve the solution by assigning neighborhood cells where fixing a radius size R. Cell i and s are said to be *neighbors* if $(dist(i,s) \leq R)$ were $dist$ is computed as seen in formulae (3.1) and R is a fixed values of neighborhood radius that can be modified.
In this case, since there is two objective functions, two swarms $\{S_1, S_2\}$ of N particle each, are used. S_1 evaluates the *Location update 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.

Construction of a particle

A point consists in a solution represented by an N-length array that contains the sequence of N Cells to be assigned. We define a particle i at iteration t as:
$X^p_{i,t} = \{(x^p_{i1,t}; x^p_{i2,t}; ..; x^p_{in,t}), (y^p_{i1,t}; y^p_{i2,t}; ..; y^p_{im,t})\}$, $x^p_{ij,t} = r$, where $r = \{1, 2, .., m\}$ define the number of location areas. $x^p_{ij,t} = r$ if Cell j of particle i is assigned to LA r for the objective p, $p = \{1, 2\}$. $y^p_{il,t} = f, r$ where $f = \{1, 2, .., z\}$ define the number of BTS and $r = \{1, 2, .., m\}$ the number of location areas. $y^p_{il,t} = f, r$ if BSC l of particle i is assigned to BTS f and to LA r for the objective p, $p = \{1, 2\}$.
For Example let consider the sequence $X^1_{i,t} = \{1, 2, 1, (2, 1), (1, 2)\}$ for the 1st objective, where we have considered the assignment of 3 cells, 2 LA, 2 BTS and 2 BSC. By this definition we have $x^1_{i1,t} = 1$, $x^1_{i2,t} = 2$, $x^1_{i3,t} = 1$ and $y^1_{i1,t} = (2, 1)$, $y^1_{i2,t} = (1, 2)$.

	$Cell_1$	$Cell_2$	$Cell_3$	BSC_1	BSC_2
LA	1	2	1	21	12

Initial population

The population is initialized by growing regions around an initial seed cell and then grouping neighboring cells into LAs while maintaining the paging bound. For **the first**

swarm S_1, each solution $y^p_{il,t}$ created is then evaluated with the fitness function that eliminates the infeasible solutions that not satisfies the capacity constraints.
The initialization operator works as follows:

① Create an initial population $x^p_{ij,t}$. We choose an initial cell to assign in the new LA (choose randomly between LA_1 and LA_2);

② Determine the nearest neighbors of the added 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 paging bound (proximity and paging constrains);

④ Repeat step 2 until all cells have been assigned;

⑤ Built the $y^p_{il,t}$ from the $x^p_{ij,t}$ assignment, were BTS are assigned randomly without respecting BSC capacity constraints;

⑥ Repeat step 5 until all BTS were assigned.

For **the second swarm** S_2, each initial solution created $x^p_{ij,t}$ is evaluated with the fitness function that eliminates the infeasible solutions that not satisfies the paging constraints.
The initialization operator works as follows:

① Create an initial population $x^p_{ij,t}$ where cells are assigned randomly in the new LA;

② Check the paging and proximity contraints. This operator determine the neighbors of the added cell that is also assigned to the same BTS, and 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 Built the $y^p_{il,t}$ from the $x^p_{ij,t}$ repaired. We choose a BTS, determine the nearest neighbors of the added BTS and add them to the BSC and LA, until no more can be added without violating BSC capacity constraints (BSC capacity constraints);

⑤ Repeat step 4 until all BTS were assigned.

For these two swarms, the particle vectors will be iteratively modified based on collective experiences in order to improve their solution quality.
During the built of the two populations, we have considered the neighbors of the added cell and add them to the LA. Our particles compound two main parts: *Assignment of cells to LA* ($x^p_{ij,t}$) and *Assignment of BTS to BSC and BTS to LA* ($y^p_{il,t}$). *Assignment of cells to LA* is the initial population generated by construction, for the first swarm, and randomly generated for the second swarm. Then we apply the Paging and the Proximity constraints to get a feasible assignment of all cells to the fixed LAs. The resulting population enters into the *Assignment of BTS to BSC and BTS to LA* process. This second step consists of taking the population solutions, one by one, get the BTS of the assigned cells, apply the TRX, Traffic and Links of BSC capacity constraints.

Particle vector modification

At each evolutionary iteration, the particle P_i updates its velocity $V^{[j]}_{(id,t)}$ and position $X^{[j]}_{(id,t)}$ referring to the personal best experience ($P^{[j]}_{(id,t)}$) and the swarms best experience ($Pg^{[k]}_t$) using Eqs.(4,5).
Since there is two objective functions, two swarms ($S1, S2$) are used for the two objective. The particles velocity and position update equations for the first swarm S_1:

$$V^{[S_1]}_{(id,t+1)} = K^{[S_1]}.[\omega.V^{[S_1]}_{(id,t)} + c_1 r_1.(P^{[S_1]}_{(id,t)} - X^{[S_1])}_{(id,t)} + c_2 r_2.(Pg^{[S_2]}_t - X^{[S_1]}_{(id,t)})] \quad (5.a)$$
$$X^{[S_1]}_{(id,t+1)} = X^{[S_1]}_{(id,t)} + V^{[S_1]}_{(id,t+1)} \quad (6.a)$$

The particles velocity and position update equations for the second swarm S_2:

$$V^{[S_2]}_{(id,t+1)} = K^{[S_2]}.[\omega.V^{[S_2]}_{(id,t)} + c_1 r_1.(P^{[S_2]}_{(id,t)} - X^{[S_2]}_{(id,t)}) + c_2 r_2.(Pg^{[S_1]}_t - X^{[S_2]}_{(id,t)})] \quad (5.b)$$
$$X^{[S_2]}_{(i,t+1)} = X^{[S_2]}_{(id,t)} + V^{[S_2]}_{(id,t+1)} \quad (6.b)$$

Where c_1 and c_2 are the cognitive and interaction coefficients, r_1 and r_2 are random numbers drawn from $U[0,1]$. Here, the inertia weight parameter ω is adjusted dynamically during the optimization. A starting value of $\omega = 1$ is used to accommodate a more global search and is dynamically reduced to $\omega = 0.4$ referring to the following equation:

$$\omega = \omega_{max} - [\{(\omega_{max} - \omega_{min}).it_{max}\}.it] \quad (7)$$

With ω_{max} the initial weight factor, ω_{min} the final weight factor, it the current iteration number and it_{max} represents the maximum number of iterations.
When the algorithm is terminated with a given maximum number of iterations, the best experienced position by the entire swarm is reported as the final solution.

Fitness function

The fitness function $fitness(X)$ measures to what extent the particle solution X satisfies the objective of the constrained optimization problem.
Peng-Yeng et al. 2006 defined two important criteria that should be concerned: $First$, the solution resulting in a better objective value from the objective function is considered to satisfy more the objective of the problem than the solution with a worse objective value. *Second*, the solutions are divided into several categories according to there feasibility: solutions which satisfy all of the constraints have the highest merit, while the solutions which violate at least one of the constraints have a Fitness less better.
Consequently, we will devise a penalty function $J(X)$ to estimate the infeasibility level of a particle solution. With respect to the above considerations, we define the fitness function as follows:

$$fitness(X) = F(X) = (Z(X) + J(X))^{-1} \quad (8)$$

where $Z(X) = (Z_1(X), Z_2(X))$ are the minimized objective functions of the BLD problem (See Eq. (3.15) and (3.16)) and $J(X)$ is the quantified amount of mismatch if X is infeasible; otherwise, $J(X)$ is set as 0. Therefore, the higher the fitness value of X, the more satisfactory the solution X.
Since the constraints (3.2)-(3.7) and constraints (3.13) and (3.14) are always satisfied, the penalty function is only related to constraints (3.8)-(3.12), and it is given by:

$$\begin{aligned} J(X) &= \textstyle\sum_r max(0, \sum_i l_{ir}\lambda_i - PB_r) + \sum_k max(0, \sum_r \sum_j \sum_i a_{ij} y_{jkr} c_i - C_k^{BSC}) + \\ &+ \textstyle\sum_k max(0, \sum_r \sum_j \sum_i a_{ij} y_{jkr} r_i - R_k^{BSC}) + \sum_k max(0, \sum_r \sum_j y_{jkr} - N_k^{BSC}) \end{aligned} \quad (9)$$

External multiplication of a coefficient by a velocity

The PSO 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 $\omega = 0.4$. The values of the coefficients r_1 and r_2 in Eq.(5) 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 calculated as mentioned in Equation (3.c), will be equal to 1 and then, it will no more affect the velocity update. We fixed a constant $V_{max} = 4$ to limit the range of velocity. The PSO parameters were the same for each swarm and for all simulation runs except for the two last experiments. For all the cases considered, we use the global variant of PSO.

Stopping criterion

The PSO algorithm stops after it has run for a preset maximum number of iteration.

Following is the overview of the pseudo-code of the VEPSO algorithm employed for the current optimization problem.

Inputs:

N: Size of the particle i.e. Number of cells to be assigned.

M: Size of the swarm in S_1 and S_2.

it_{max}: Maximum number of iterations.

$t = 0$;

Outputs:

it: Number of iterations at time t.

$X[i]$: The position of the ith particle in the population; $\forall i = \{1, .., M\}$

$X[i][d]$: LA number to which the dth cells in the ith particle is assigned;$\forall i = \{1, .., M\}$ and $\forall d = \{1, .., N\}$

$F[i]$: Fitness function of the particle i according to the Eq.(8)

$V[i]$: Velocity of the ith particle.

P_{global}: The index to global-best position.

$P_{Local}[i]$: The position of the local-best position of the ith particle.

Initialization

For each swarms S, $S \in \{S_1, S_2\}$

/*the initialization is done according to "Initial population" in section 3.3.3*/

∘ Step 1

- For d cell $d = \{1, .., N\}$; $X[i][d] \leftarrow Random^P()$ from $[1, NbrLA]$.
- All particles are checked to satisfy all the constraints.
- $V[i] \leftarrow Random^v()$ from $[0, 1]$.
- Evaluate $F[i]$.
- $P_{Local}[i] \leftarrow X[i], \forall i = \{1, .., M\}$.
- $P_{global} \leftarrow max(F[1], F[2], .., F[M])$. /*Get the index of the particle*/

Iteration

For each swarm S, $S \in \{S_1, S_2\}$

○ Step 2

Loop ($d = 1$ to N)

- Generate the inertia weight (ω) referring to Eq.(7) using it and it_{max}
- Update the particle velocity $V[i][d]$ referring to Eq.(5)
- Update the particle position $X[i][d]$ referring to Eq.(6)
- After each update check whether the variables of each particle satisfy the constraints.
- Evaluate $F[i]$

End Loop

○ Step 3

Repeat

Loop ($i = 1$ to M)

IF $F[i] >= F(P_{Local}[i])$ THEN /*identify the personal best solution (referring to Eq.(4))*/

Loop ($d = 1$ TO N)

- $P_{Local}[i][d] \leftarrow X[i][d]$
- Next d

End Loop

End IF

○ Step 4

IF $F[j] > F(P_{global})$ THEN /*identify the global best solution (referring to Eq.(4))*/

- $P_{global} \leftarrow j$
- Next j

End IF

Repeat *Step 2*

End **Loop**.

Until ($it = it_{max}$)

3.4 Computational Experiments

PSO 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, at 3.20 GHz processor with 512 Mb of RAM and under Microsoft Windows XP.
To better manage our memory, we use "threads" that are a portion of code to run in parallel to other treatments. Threads do not run at the same time, but time-share, this is why it is important for a Thread to leave a chance for others to run. Other than the threads, it was assigned the objective "Synchronized" for some methods because all these are mutually blocked when applied to the same object.
As our algorithm must handle a number N of particle in each swarms, the "threads" with the object "Synchronized" will be a perfect choice for the sharing of tasks the optimization in CPU time of our algorithm.
For the implementation of our VEGA algorithm, we inspired from the application of Dr Mark.C Sinclair who used the PSO JAVA applet for solving the two of De Jong's test functions in two dimensions problem.
The execution time of our program depends on the size of the particles in each swam and for each VEPSO parameters used. Experiments are repeated 1000 time for each parameter set. The size of the problem is defined by data that are collecting from TUNISIANA at different times in a week.
All the experiments use the same data with varying VEPSO parameters. In the first group of experiments, we vary the size of the problem by taking different number of particles in each swarm. In the second group, we take different values of seed cell trying to find the optimal location of the initial cell. Then, we will vary in the radius of neighborhood to fixe the border shape. In the last two experiments, we will vary in the velocity and the inertia factor, in order to optimize the results founded in the previous experiments.
The performance of the VEPSO algorithms is compared with the results of TUNISIANA using the TLPM for LA Redefinition (see section 2.5.3).

3.4.1 Implementation of the Algorithm

The GSM network elements (cells, BTSs, and BSCs) have a number of shared attributes such as call traffic load, Paging load and TRX load. Therefore a common database is defined, namely 'lap ', and all network element classes are using this database. We used

a database to store information and to avoid misuse of the tables, thus better manage computer memory.
'lap 'database consists of 7 tables, including the last one for management efficient solutions, managed as following:

- ✧ **'bsc'** table has the name of the BSC,the LA name in which it is assigned and TRX, traffic and number of links for each BSC:
  - NumBSC *(Integer)*: Distinguishing id number of the BSC in the network and is the name of the BSC.
  - TRXBSC *(Integer)*: TRX capacity of each BSC.
  - TRAFFBSC *(Integer)*: Call traffic capacity (in Erlang) of each BSC.
  - LINKBSC *(Integer)*: Number of available link of each BSC.
  - NumLAC *(Integer)*: Number of the LA in which is located the BSC.
  - NumBSC1 *(Integer)*: Distinguishing number of the BSC between [0,2].
- ✧ **'bts'** table has the name of the BTS and the LA name in which it is assigned:
  - NumBTS *(Integer)*: Distinguishing id number of the BTS in the network and is the name of the BTS.
  - NumLAC *(Integer)*: Number of the LA in which is located the BTS.
  - NumBTS1 *(Integer)*: Distinguishing number of the BTS between [0,53].
- ✧ **'cells'** table has the name of the Cell; the LA, BTS and BSC name in which it is assigned; TRX, Traffic and paging values for each cells:
  - NumCell *(Integer)*: Distinguishing id number of the Cell in the network and is the name of the Cell.
  - NumLAC *(Integer)*: Number of the LA in which is located the Cell.
  - NumBTS *(Integer)*: Number of the BTS in which is located the Cell.
  - NumBSC *(Integer)*: Number of the BSC in which is located the Cell.
  - TRXCell *(Integer)*: TRX capacity of each Cell.

- TRAFFCell *(Double)*: Call traffic capacity (in Erlang) of each Cell.
- PagingCell *(Integer)*: Number of paging load request for each Cells.
- NumCell1 *(Integer)*: Distinguishing number of the Cell between [0,137].

✧ **'distance'** table, contains the distance cost between all BTS and BSC for a given LA, and has the following attributes:

- NumBTS *(Integer)*: Distinguishing id number of the BTS.
- NumBSC *(Integer)*: Distinguishing id number of the BSC.
- DISTANCE *(Double)*: distance cost between BTS an BSC.

✧ **'distancebts'** table, contains the distance cost between all BTS for a given LA, and has the following attributes:

- NumBTS *(Integer)*: Distinguishing id number of the BTS1.
- NumBTS1 *(Integer)*: Distinguishing id number of the BTS2.
- distance *(Double)*: distance cost between BTS1 an BTS2.

✧ **'handover'** table, contains the Handover between all Cells for a given LA, and has the following attributes:

- NumCell1 *(Integer)*: Distinguishing id number of the Cell1.
- NumCell2 *(Integer)*: Distinguishing id number of the Cell2.
- Handover *(Integer)*: Handover value between Cells.

✧ **'fitness'** table, contains the efficient fitness value useful to have the best assignment, and has the following attributes:

- Fitness1 *(Double)*: First objective value function (Z_1).
- Fitness2 *(Double)*: Second objective value function (Z_2).

While executing algorithms, our program updates all objective function values in the *'fitness'* database table, leaving only effecient solutions, as mentionned above in Section

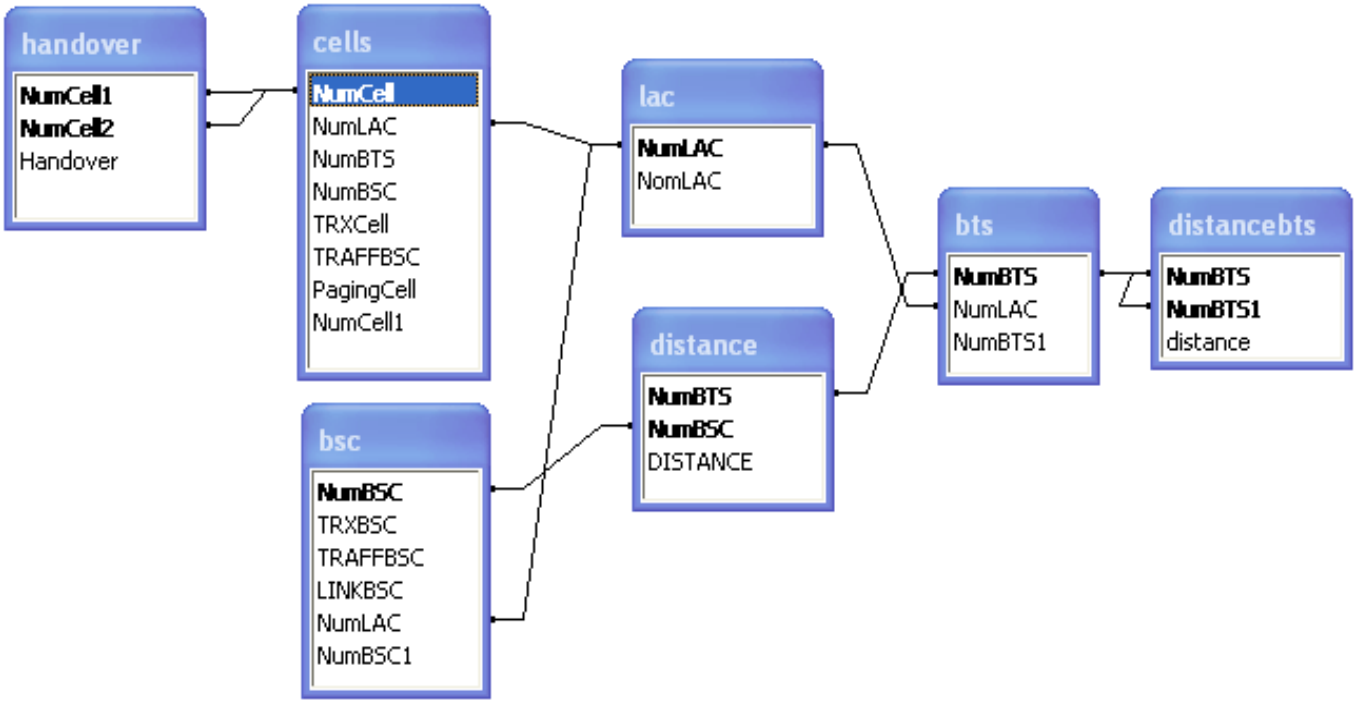

Figure 3.3: BLD Database

3.2.2. Figure 3.2 shows the Database design with instances of these table.

After the algorithm finiched its execution, we collect all the efficient solutions in a text file namely "Efficient Solutions.txt", then, the contents of the table is automatically deleted making it ready for the next running.

An other text file named "Solutions.txt" will collect a trace of all the solutions in each iteration, during the running of the algorithm. We write in only if there is an improvement (minimization) of one of the objective function Z_1 and Z_2, of our problem .

We defined five classes to represent the PSO for the BLD problem in the JAVA project. The first one namely, *Interface*, is defined to collect the entered parameters of the algorithm, useful for the initialization and the update of swarms. The second class, namely *Swarm1*, is defined to initialize two swarms, with N Particles in each swarm. Every swarm will compute the value of its fitness with the only one objective function addet to the som of penalty constraints (as mentioned in Section 3.3.3). These fitness functions will be useful to compute the *Global-best* of every swarms and then update the velocity

and the position of every particle by exchanging their best experiences.
Classes number three and four, namely *Particle* and *Particle1*, are used for the initialization and the update of each particle in both $swarm_1$ and $swarm_2$. The last class, namely *Fitness*, is used to compute the fitness function and constraints for each particle in both swarms.
I did a training in a telecommunications company TUNISIANA to access their personal documents and integrate their workplace, in order to better understand the concept of LA and the process of Paging and LU.
For this case study elaborated in TUNISIANA telecommunication company, it was 3 rows entered in the **'bsc'** table, 54 rows in the **'bts'** table, 138 rows in the **'cells'** table, 162 rows in the **'distance'** table, 2862 rows for the **'distancebts'** table and 2496 rows for the **'handover'** table.

3.4.2 Run of the VEPSO Algorithm

Our problem manage 138 cells, 54 BTS, 3 BSC and 1 MSC. Indeed, the databases performed are scaled about 2862 records. Before the experiments, we will begin with a small example managing 20 particles in each swarm for 50 iterations, in order to have an overview on the execution program with the resulting outputs.
In this example, the VEPSO parameters entered are: 2 for the acceleration parameters c_1 and c_2, a starting value of $\omega = 1$ that is dynamically reduced to $\omega = 0.4$, the maximum velocity $V_{max} = 4$, the maximum number of iteration $It_{max} = 50$, and the number of population $N = 20$ particles to each swarm.
For this algorithm, the radius of neighborhood is fixed to 115000_m and the cutting procedure begin from an initial cell C_16 from BTS_16, in order to ensure a cutting that minimizes the number of border constraints and area specification. The PSO parameters used in the current case are listed in Table 3.1.

As we can see in Figure 3.6, the VEPSO settings are changed by the users. We can also stop the running to evaluate the progressing at every stage. Figure 3.4 shows the split of the LA into 2 new LAs. This cutting minimizes the LU and the distance cost. This Figure shows the allocation of BTS associated to different cells in LAs. The result of the optimization process are listed in Table 3.2.

Table 3.1: VEPSO parameters

Cognitive learning rate	$c_1 = 2$
Social learning rate	$c_2 = 2$
Inertia factor	$\omega = [1, \ldots, 0.4]$ dynamically reduced from 1 to 0.4 with each iterations.
Maximum velocity	$V_{max} = 4$ to control the maximum travel distance during each iteration.
Maximum number of iterations	$It_{max} = 1000$
Number of swarm particles	$N = 30$
Default radius of neighborhood\m	$R = 115000m$
Thresholds of paging (%)	20% for LA_1
	25% for LA_2
Default Seed cell	C_{16} from BTS_{16}

Table 3.2: Results for problem size 20 and 50 iterations

Number of particles in each swarm	**CPU time**	**Number of efficient solutions**	**Paging /LA**	**Objective Function (Z_1)**	**Objective Function (Z_2)**
20	4.8125 sec	3	($\simeq$ 15%;$\simeq$ 30%)	33749	1.0393926709399998E8

The first column refers to the number of particles in each swarm, the second column refers to the CPU time where the third accounts the number of efficient solutions.The fourth column refers to the rate of paging load in each LAs. The two lasts columns, refers to the best value of objective function respectively for Z_1 and for Z_2. The paging bound was set to 20% and 25% from the paging load of the old LA, for the new LAs.
By comparing this result by the one of TUNISIANA, we found the same shape of border for the parameters listed above in table 3.1. Figure 3.5 shows the assignment of BTS to BSC as it was defined by Orascom Telecom, this assignment was not based on the minimization of the distance between the two antennas and BTS to BTS.

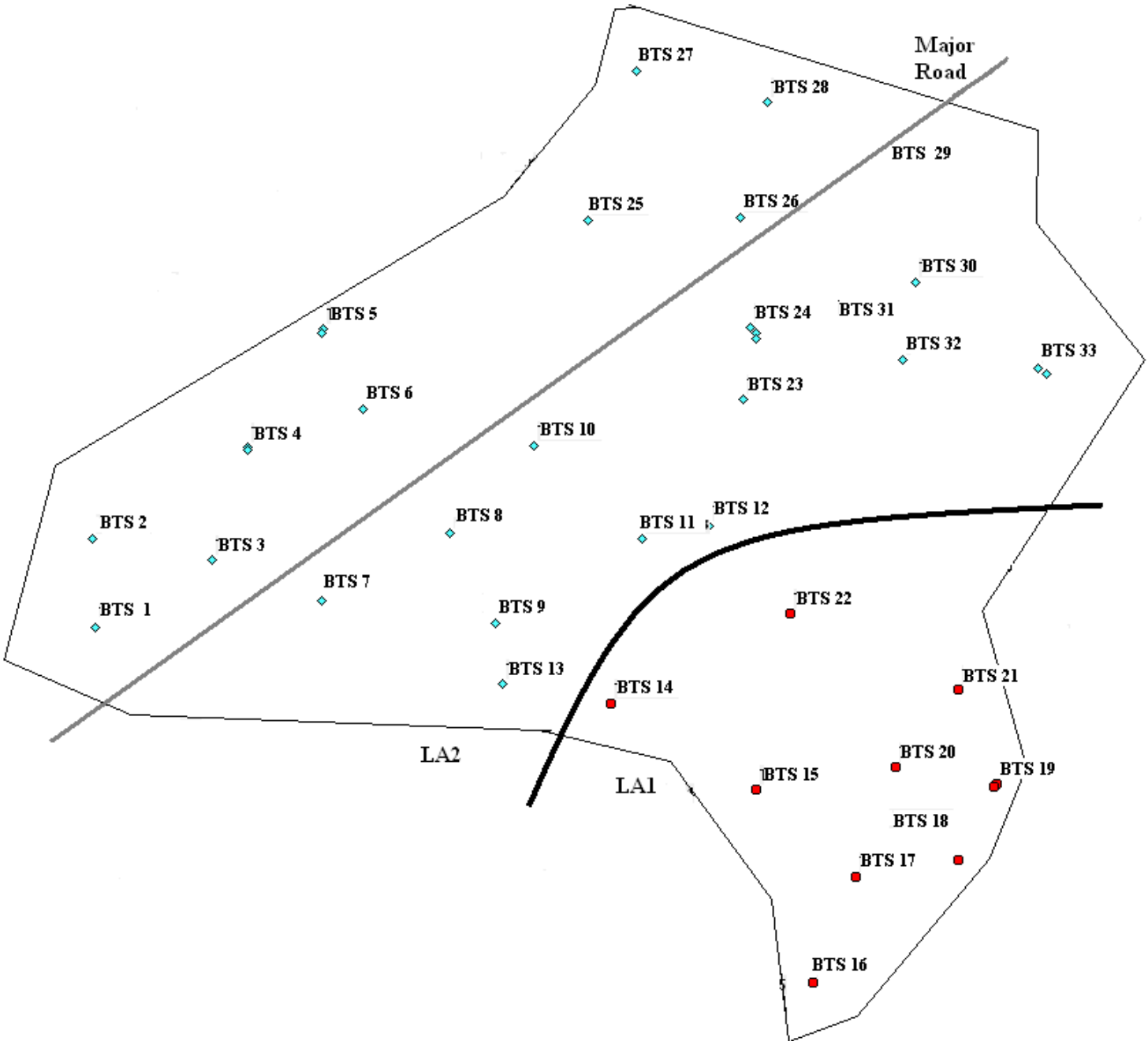

Figure 3.4: LA border in TUNISIANA

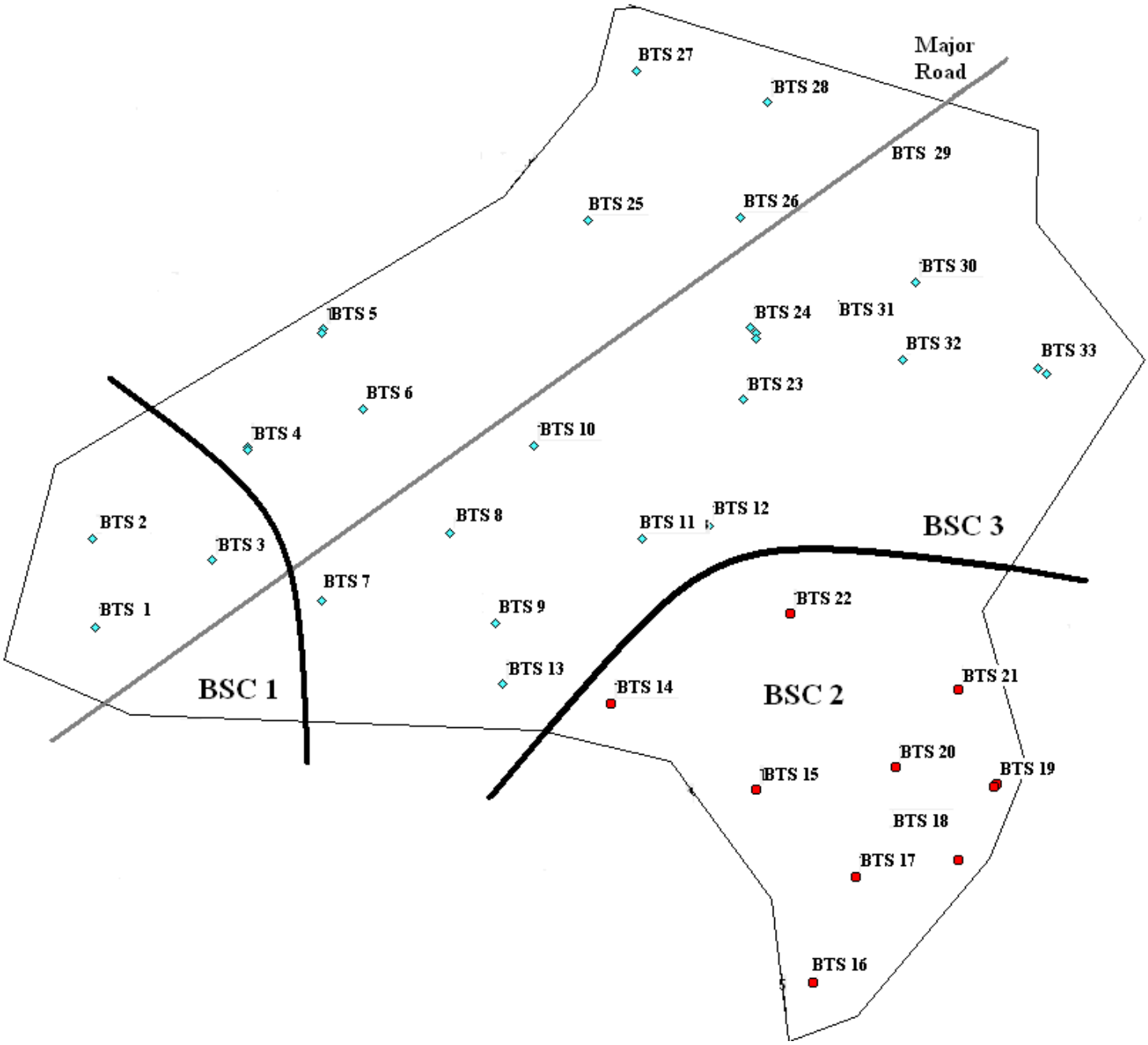

Figure 3.5: Assignment of BTS to BSC in TUNISIANA

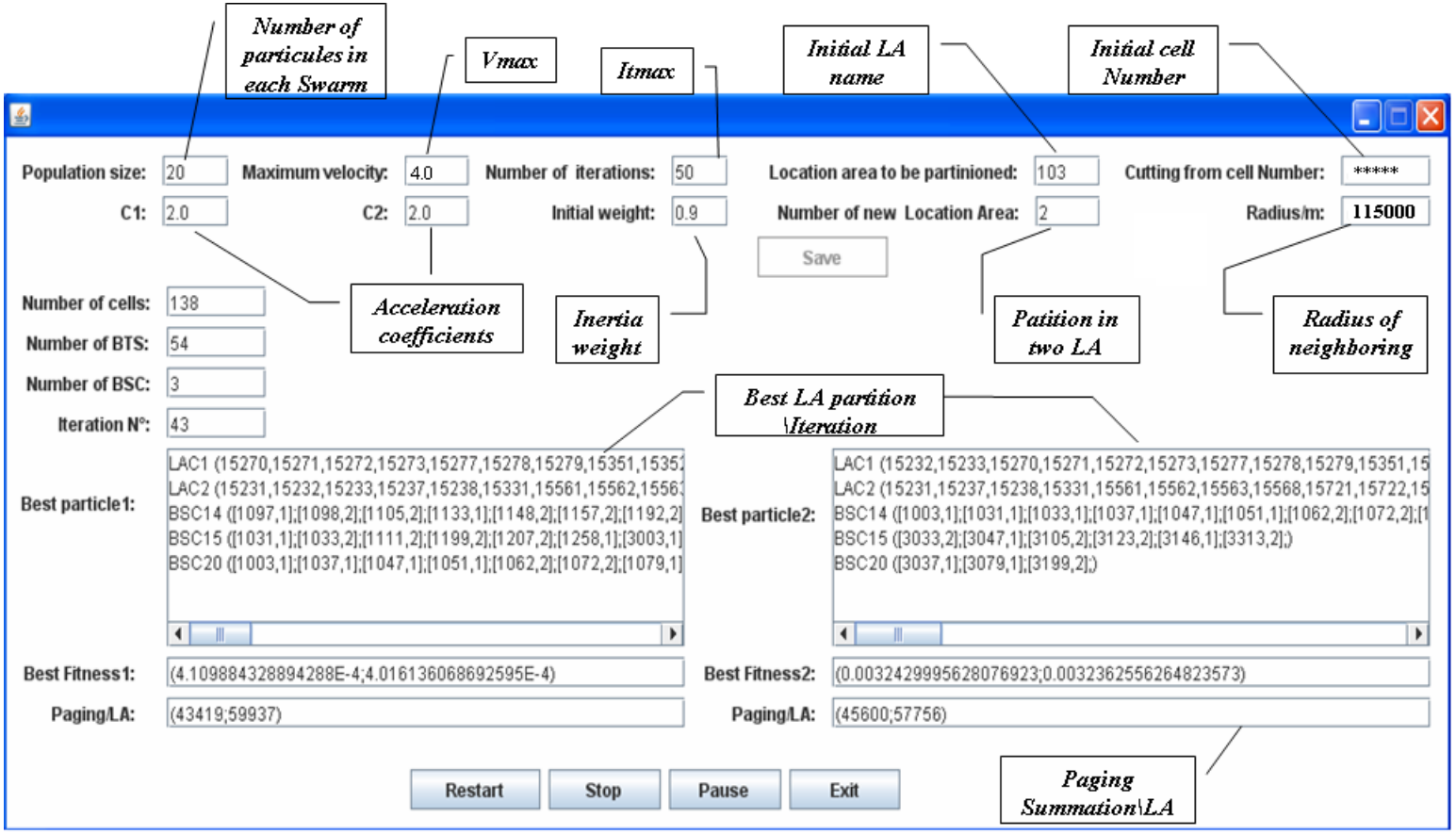

Figure 3.6: One hundred thirty eight cell configurations in VEPSO application

3.4.3 Sensitivity analysis

To show the effectiveness of our approach, we proceed by managing 138 cells, 54 BTS, 3 BSC and 1 MSC. Using these data, we will carry out the creation of a new border in this LA that minimizes both the location update and the distance cost function.
In order to deal with different scenarios of our problem, we will take different values for the number of particles in each swarm N and the Radius of neighborhood R, in addition to the variation of the seed cell around which the regions growth, and some parameters of the VEPSO algorithm as the maximum velocity V_{max} and the inertia factor W.
In each script perform, we will compare the results obtained by comparing our algorithm to the one adopted in the company proofing that our method generates better results and allows the user to choose between a range of effective solutions. This section is devoted to the resolution of BLD problem. We will summarize and perform the results obtained and made comparision with the existent real case.
The number of swarm particles is decided empirically as the current problem is fixed. Because initialization itself may take a longer time if the population size is too large, a lower population size significantly lowers the computational time.
All the following experiments will run for $It_{max} = 1000$.

Variations in the number of particles in each swarm

It can be observed from the results in Table 3.3, that the different optimal design evolved for the three different sizes of swarms are quite comparable.
The first column of Table 3.3 denote the number of particles in each swarm, the second column refers to the computing time of the algorithm fot these fixed parameters. The third column, counts the number of efficient for each population size. The fourth column refers to the rate of paging load in each LA. The two last column represents the best value of objective function for Z_1 and the best value of objective function for Z_2.
We can deduce that, for the a same seed cell and the same radius of neighborhood, we have similar paging rate per LAs. As we increase the number of particles N in the VEPSO algorithm for each swarm, the number of efficient solutions(see Figure 3.7) and the computing processing time (see Figure 3.8) increases.
The optimal assignment for the first and the second objective function is reached by all

our simulations for different size of swarms.
As lower values of swarm size are favored, causing a more focused exploration of the search space, we will fixe the number of population to 30 in the next experiments.

Table 3.3: Results for variation in the number of particles in each swarm

Number of Particles	**CPU time**	**Efficient solutions**	**Paging /LA**	**Objective Function/Z_1**	**Objective Function/Z_2**
30	73.453125 sec	6	($\simeq$ 15%;$\simeq$ 30%)	33749	1.03401020429E8
40	80.109375 sec	7	($\simeq$ 15%;$\simeq$ 30%)	33749	1.03908421726E8
50	82.34375 sec	6	($\simeq$ 15%;$\simeq$ 30%)	33749	1.02895858333E8
60	82.734375 sec	8	($\simeq$ 15%;$\simeq$ 30%)	33749	1.0289797035E8
70	90.46875 sec	9	($\simeq$ 15%;$\simeq$ 30%)	33749	1.03382038586E8)
75	90.46875 sec	9	($\simeq$ 15%;$\simeq$ 30%)	33749	1.03906587527E8

Variations of the seed cell

It can be observed from the results, summarized in Table 3.4, that different optimal LA designs are got for different seed cells by comparing the percentages of paging load in each LAs. We selected five different cells spread into different locations of the our LA (see figure 3.4), in order to have different cutting angle of the LA.
The first column of Table 3.4 refers to the initial cell number from which we will begin 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 paging load in each LA, while the fourth counts the number of

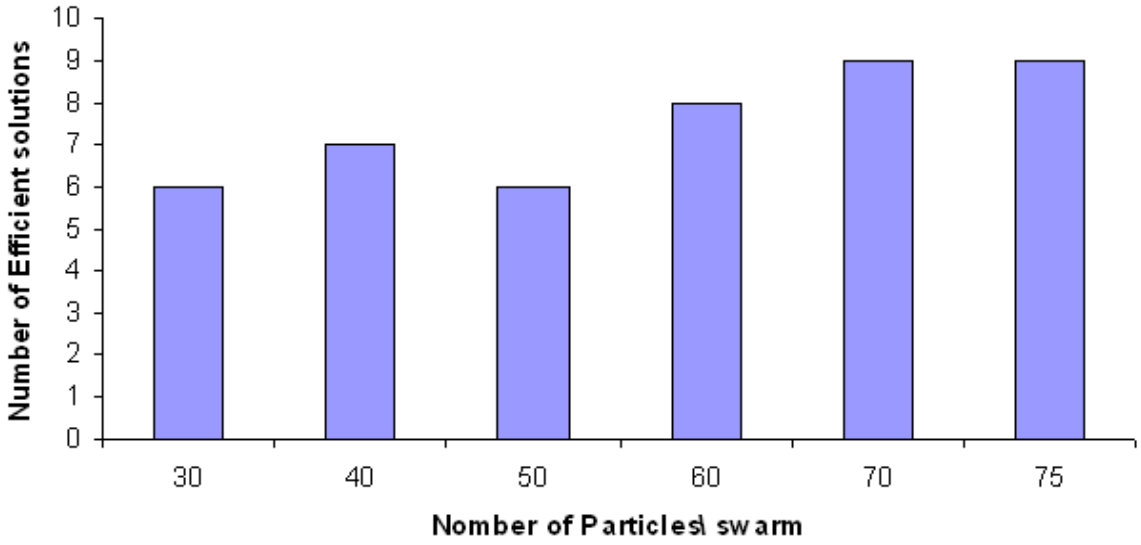

Figure 3.7: Number of efficient solutions for different sizes of swarms

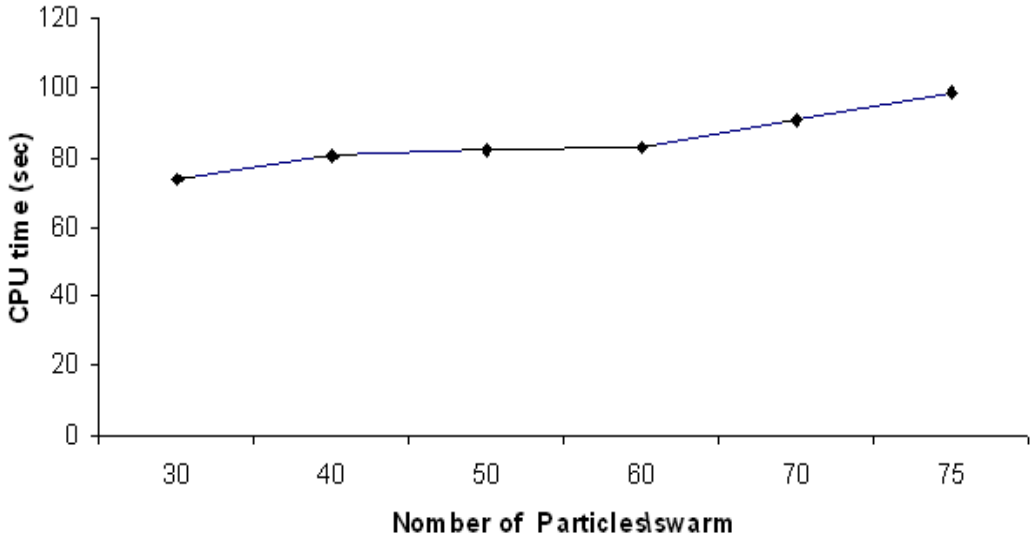

Figure 3.8: The CPU time for different sizes of swarms

efficient solutions for each initial cell. The two last columns represents the best objective functions for Z_1 and for Z_2.
We selected the following cells: cell C_5 belong to BTS_5, cell C_{33} belong to BTS_{33}, cell C_{16} belong to BTS_{16}, cell C_2 belong to BTS_2 and cell C_{11} belong to BTS_{11}.
Referring to Table 3.4, we can notice the change in the value of the CPU time with the

Table 3.4: Results of varying the seed cell

Number of seed Cell	**CPU time**	**Paging /LA**	**Number of efficient solutions**	**Objective Function/Z_1**	**Objective Function/Z_2**
C_5	75.78125 sec	($\simeq$ 19%;$\simeq$ 21%)	9	106587	1.0287413809E8
C_{33}	93.96875 sec	($\simeq$ 10%;$\simeq$ 35%)	7	139621	1.0287413809E8
C_{16}	90.71875 sec	($\simeq$ 15%;$\simeq$ 30%)	5	33749	1.03909450237E8
C_2	92.4375 sec	($\simeq$ 16%;$\simeq$ 29%)	7	52381	1.0439589065E8
C_{11}	97.234375 sec	($\simeq$ 11%;$\simeq$ 34%)	8	173216	1.03382038586E8

change in the choice of initial cell, taking the lowest value for cell C_5 and the higher one for cell C_{11} (see Figure 3.10).
By comparing paging rates, the assignment that satisfies at more constraints of paging is the one that starts from seed cell C_{16}, with a paging load of ($\simeq$ 15%;$\simeq$ 30%) for thresholds of 20% and 25% respectively for the LA_1 and LA_2.
Moreover, referring to the objective functions, by choosing the initial cell C_{16} we get the lowest LU cost value that minimizes at most the first objective and than give the better border shape to cut our LA (see Figure 3.11)
The best assignment of BTS to BSC minimizing the second objective costs, is the one starting form cell C_5 and C_{33} (see Figure 3.12). However, it exists other geographical

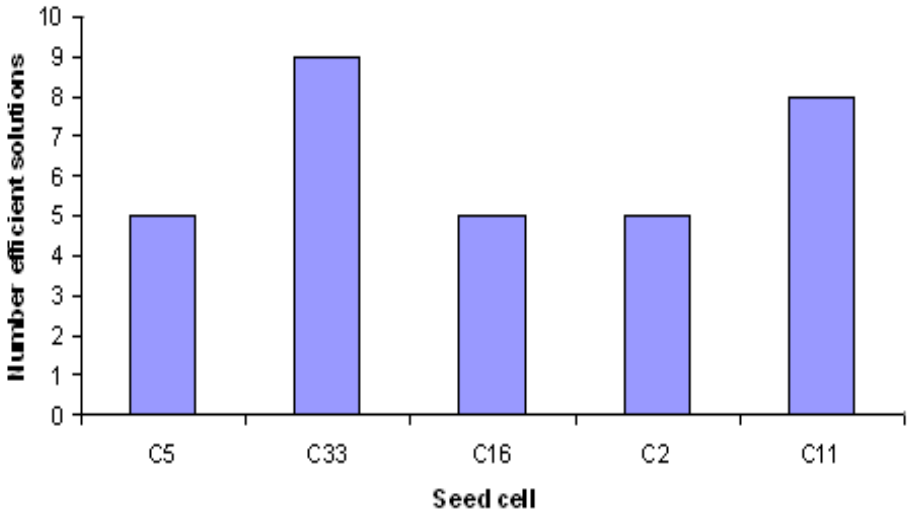

Figure 3.9: Number of efficient solutions for different seed cells

constraints (crossed by a highway for example) which decides on the way how we will made our devision on real case. So, we have to check on the feasibility of the proposed split, due to the geographical location of different cells (see Major road in Figure 3.5). Based on these geographical constraints, we chose the seed cell number C_{16}. This scenario is the one that satisfy at most to all our constraints, therefore we adopt this alternative with a radius size of 115000_m.
Figure 3.9 shows the variation in the number of efficent solutions for a given value of cell C_i.

Variations of the Radius R

Based on experimentations above to determine the seed cell, we choose cell C_{16} belonging to BTS_{16}. We will vary, during these experiments, the radius of neighborhood to get the best border for our LA that will minimize both the LU and BTS-BSC distance costs.
The first column of Table 3.5 refers the radius of neighborhood, where the second refers to the computing time of our algorithm. The third columns, refers to the rate of paging load in each LA. The fourth column, represents the number of efficient solution for each size of radius. The two last columns represents the best value of objective functions for

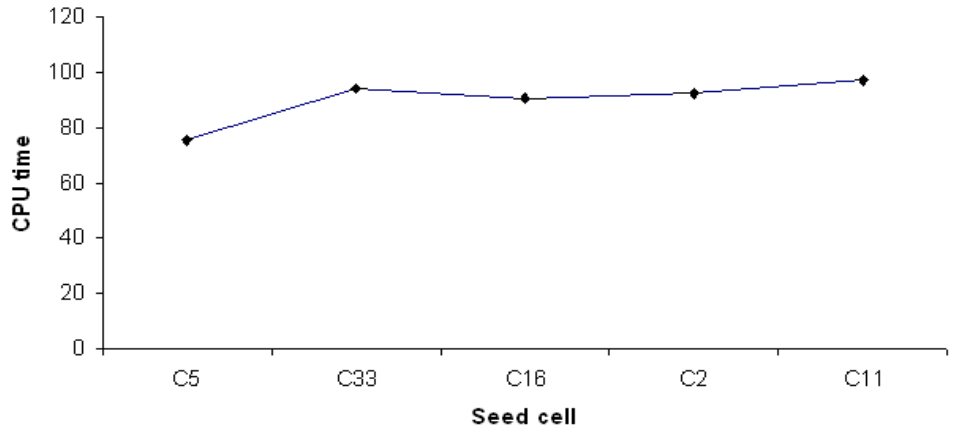

Figure 3.10: CPU times for different seed cells

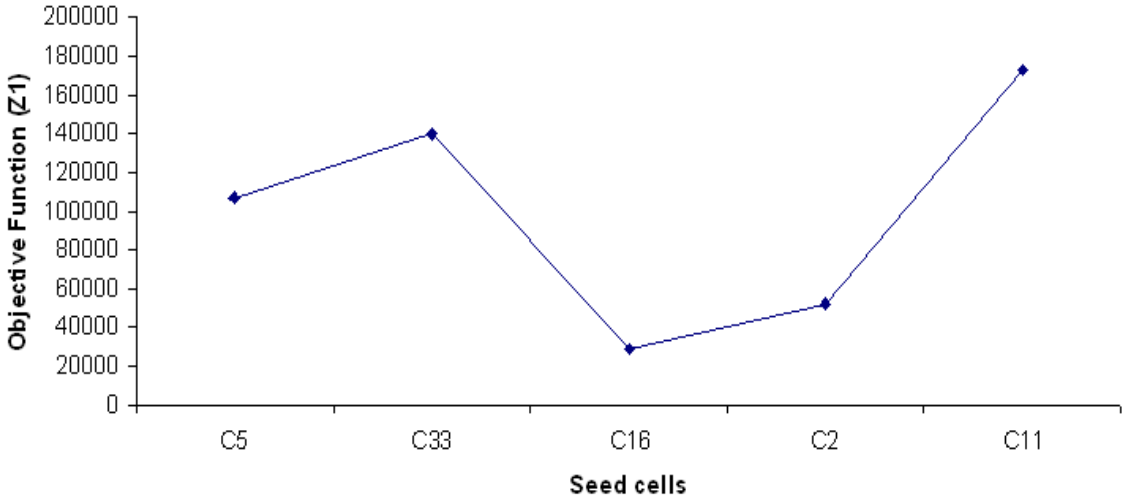

Figure 3.11: The variation of the first objective function for different seed cells

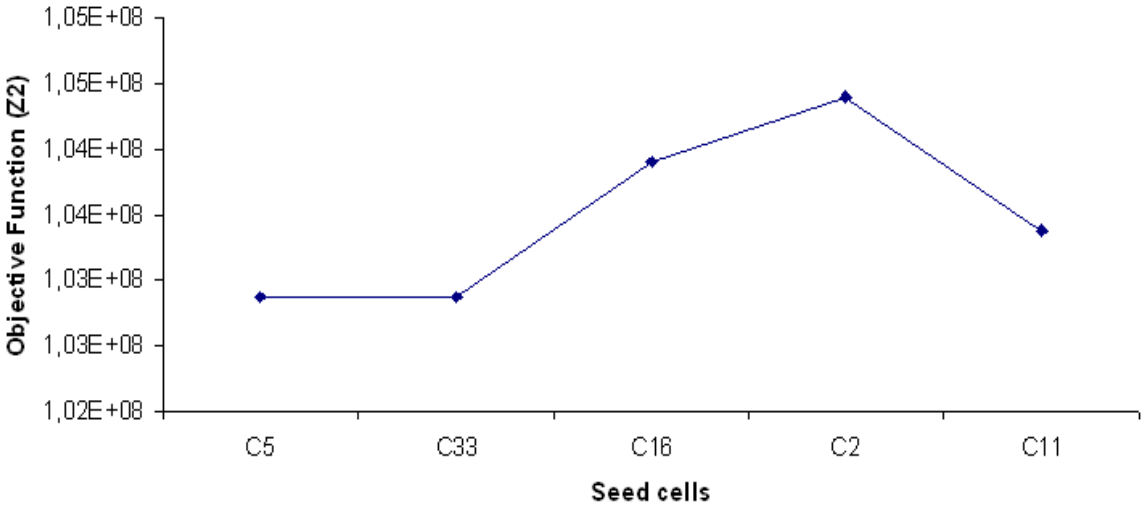

Figure 3.12: The variation of the second objective function for different seed cells

Z_1 and the best value of fitness function for Z_2.
We observed from the results in Table 3.5, that by increasing the radius' size the number of efficient solutions decreases (See Figure 3.14) while the objective function increase. Therefore, a hight fitness function satisfies at most the constraints for the VEPSO algorithm and is closer to the optimal solution.
For a radius size equal to 120000_m, we have a split that give the lowest values of both objective functions, satisfying thus to more constraints, mainly to the constraints of paging. 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 paging load in each LAs (see Figure 3.13).
We conclude then that the best radius of neighborhood, is a radius size of 120000 satisfying both objective functions Z_1 and Z_2 (see Figure 3.15 and 3.16).

Variations of the velocity V

Starting from cell C_{16} with a radius of 120000_m, we will develop a set of experimentations for the variation of the maximum velocity in order to find more satisfactory results.
The first column of Table 3.6 refers to the gap of velocity in which it will vary during the running from V_{max} to V_{min}, the second denotes CPU time for each velocity. The third column, refers to the paging load percentage for each LAs. The fourth column, counts

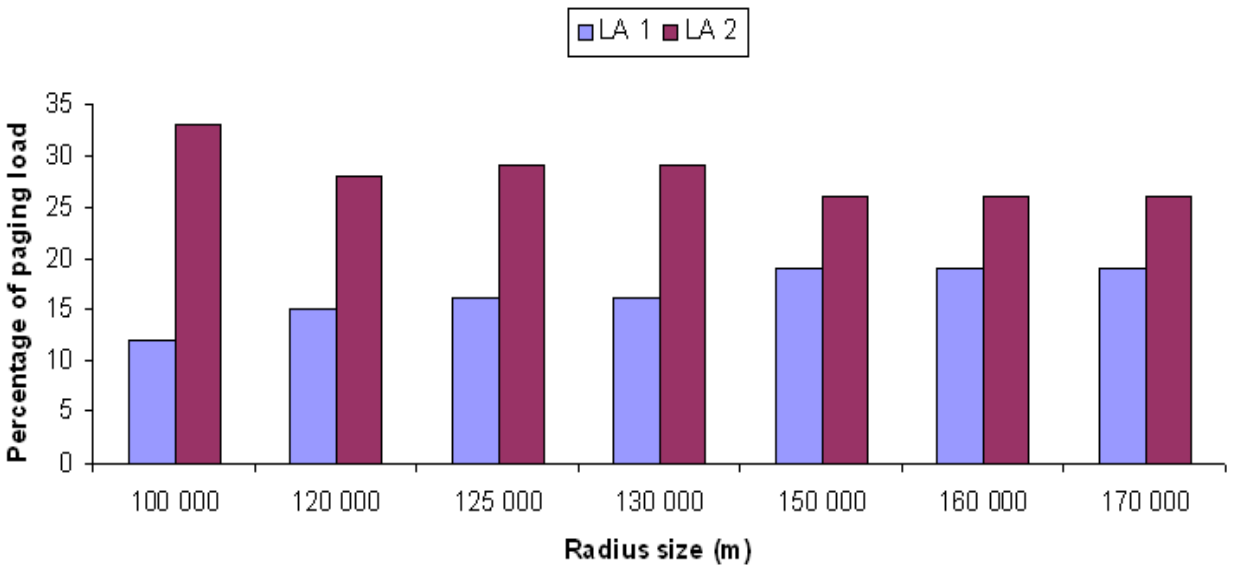

Figure 3.13: Percentage of paging load/LA for different radius of neighborhood

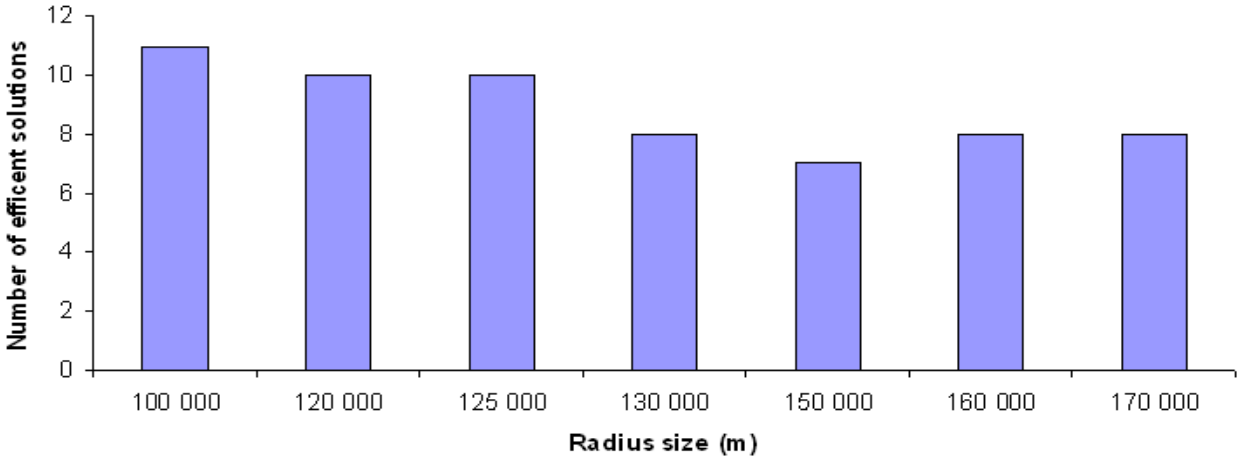

Figure 3.14: Number of efficient solutions for different neighboring radius

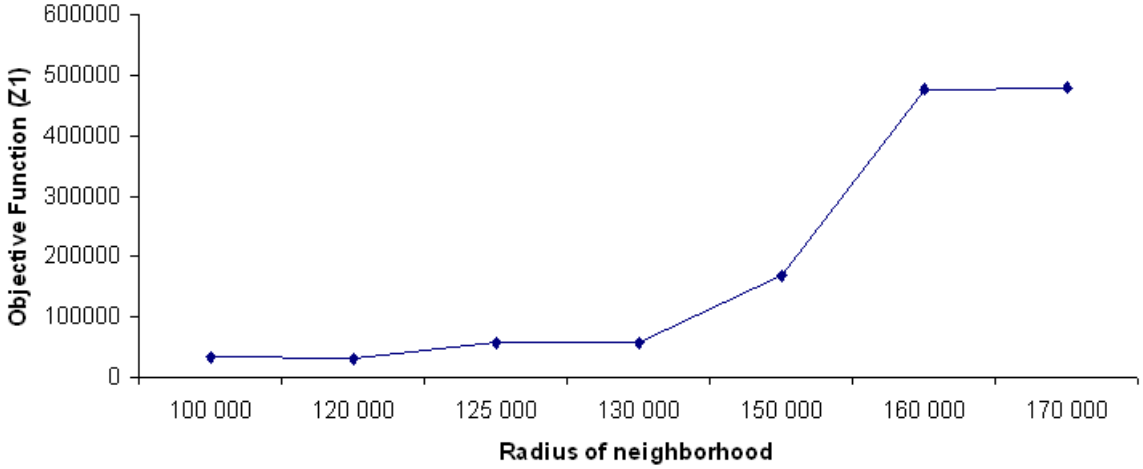

Figure 3.15: Variation of the objective function Z_1 for different radius of neighborhood

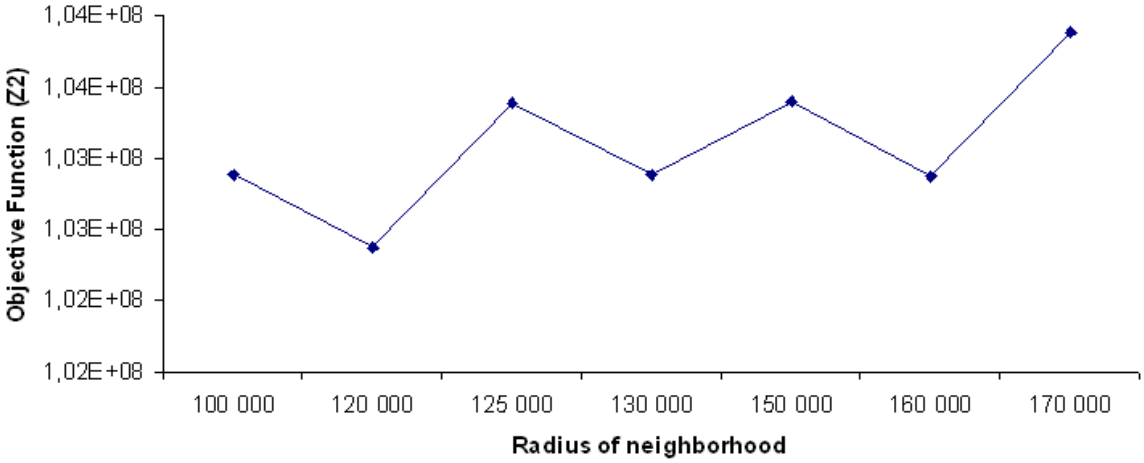

Figure 3.16: Variation of the objective function Z_2 for different radius of neighborhood

Table 3.5: Results for different radius of neighborhood

radius size/m	CPU time	Paging /LA	Number of efficient solutions	Objective Function/Z_1	Objective Function/Z_2
100000	75.109375 sec	($\simeq$ 12%;$\simeq$ 33%)	11	33749	1.02881466747E8
120000	75.178105 sec	($\simeq$ 15%;$\simeq$ 28%)	10	28873	1.02371309392E8
125000	75.078125 sec	($\simeq$ 16%;$\simeq$ 29%)	10	56156	1.03390933732E8
130000	90.359375 sec	($\simeq$ 16%;$\simeq$ 29%)	8	56156	1.02884369629E8
150000	95.296875 sec	($\simeq$ 19%;$\simeq$ 26%)	7	166661	1.0339821455E8
160000	98.515625 sec	($\simeq$ 19%;$\simeq$ 26%)	8	477642	1.0287580014E8
170000	90 sec	($\simeq$ 19%;$\simeq$ 26%)	8	480508	1.03884589466E8

the number of efficient solution. The two last columns denote the best objective function for Z_1 and the best objective function for Z_2.

Referring to the results from Table 3.6, we can observe that the different objective functions for the three different velocity range values are quite similar, there is a small variation with the second objective function, while the assignment of cells to LA does not change in the three scenarios.

In addition, we observe a decrease in the number of efficient solutions (see Figure 3.17). The more we narrowed the interval V_{max}, the more the number of efficient solutions decreases. In addition to that, we can observe a variation in the CPU time by varying V_{max}.

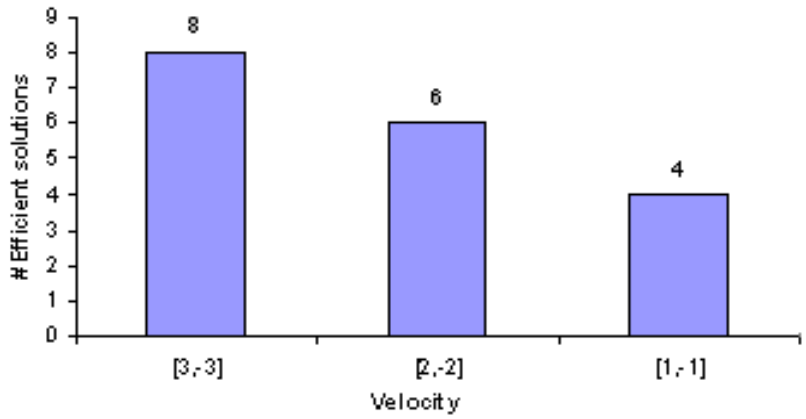

Figure 3.17: Number of efficient solutions for different velocity V

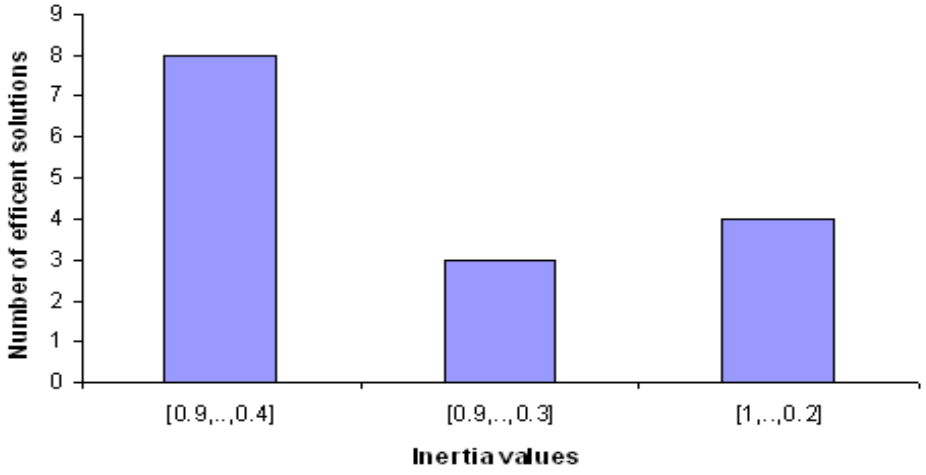

Figure 3.18: Number of efficient solutions for different inertia factor W

Table 3.6: Results of varying velocity V

Maximum velocity $[V_{max}..V_{min}]$	CPU time	Paging /LA	Number of efficient solutions	Objective Function/Z_1	Objective Function/Z_2
$[3, -3]$	93.546875 sec	($\simeq$ 19%;$\simeq$ 26%)	8	28873	1.02903416969E8
$[2, -2]$	86.96875 sec	($\simeq$ 19%;$\simeq$ 26%)	6	28873	1.0287892301E8)
$[1, -1]$	74.3125 sec	($\simeq$ 19%;$\simeq$ 26%)	4	28873	1.0289797035E8

Variations of the inertia factor W

As we concluded that the velocity V had no effect on variation of the objective function, we varies on the inertia factor W trying to find better results. The first column of Table 3.7 refers to the gap range of the inertia factor, the second denotes CPU time for each inertia value. The third column, refers to the paging load percentage for each LAs. The fourth column, represents the number of efficient for each range of inertia value. The two last columns represent the best objective function for Z_1 and the best objective function for Z_2.

As we can see from Table 3.7, The variation on the gap of the Inertia factor has no effect on the paging rate in the new LAs, however, as we increase the size of the interval W, the number of efficient solutions increases (see Figure 3.18).

It can be observed from the results, that the objective functions resulting from the three different inertia gap are quite comparable.

It differs just a little variation in the values of the second objective, which is no more significant in practical case and for the number of generated efficient solutions.

In Table 3.8, we compared between the results of the VEPSO algorithm and those of the operator. We can clearly conclude that the VEPSO algorithm is better on all our comparisons' criteria such as; CPU time, Objective functions, paging load and number

of efficient solutions.

Based on TLPM, the action of dividing a LA into new two smaller LAs decrese the paging load, but the number of location update and SDCCH traffic increased. However, the VEPSO method minimizes both LU and distance cost functions.

Figure 3.19 and 3.20 illustrate the shape of the map before and after applying our resolution new model. It summarizes all the above experimental results defining simultaneously the border for the LAs and the BTS to BSC.

VEPSO is proved to be better than TLPM, producing good quality of results that satisfies all the constraints of our model, and than converge faster to the Pareto-optimal front.

Table 3.7: Results of varying the inertia factor

Inertia gap $[W_{max}..W_{min}]$	**CPU time**	**Paging /LA**	**Number of efficient solutions**	**Objective Function/Z_1**	**Objective Function/Z_2**
$[0.9, .., 0.4]$	98.984375 sec	($\simeq$ 19%;$\simeq$ 26%)	8	28873	1.02900921071E8
$[0.9, .., 0.3]$	79.921875 sec	($\simeq$ 19%;$\simeq$ 26%)	3	28873	1.03895318811E8
$[1, .., 0.2]$	81.5625 sec	($\simeq$ 19%;$\simeq$ 26%)	4	28873	1.03398149053E8

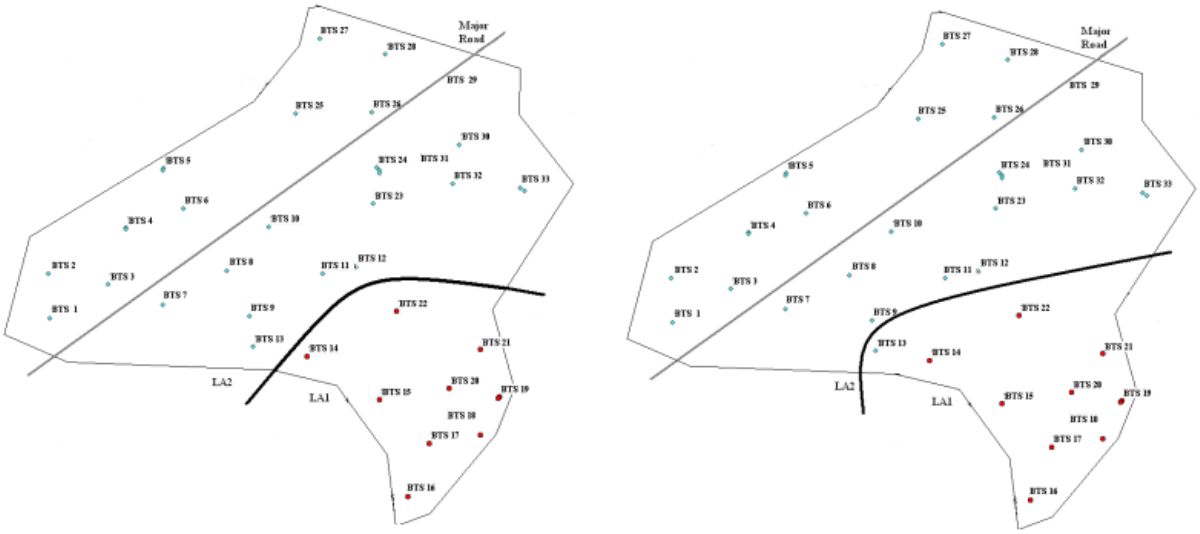

Figure 3.19: Comparision between TLPM and VEPSO of Cells to LA assignment

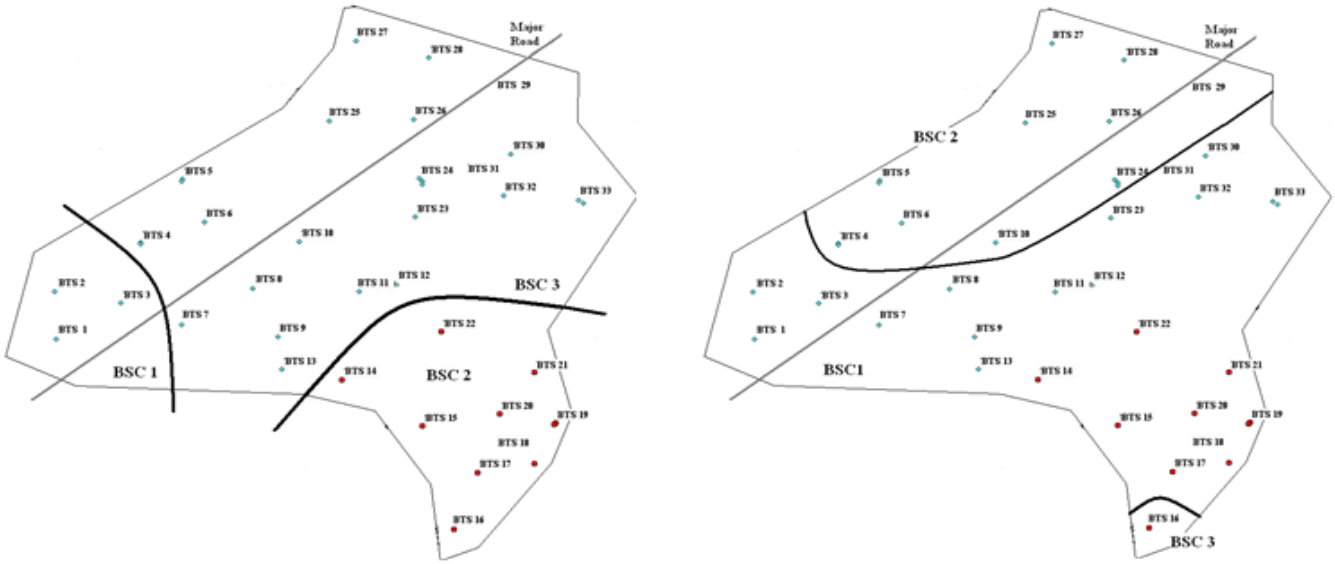

Figure 3.20: Comparision between TLPM and VEPSO of BTS to BSC assignment

Table 3.8: Comparison between TLPM and VEPSO algorithm

Comparing Parameters	**TLPM results**	**VEPSO results**
Resolution time	$\simeq$ 1 week	75.1 sec
Number of efficient solutions (scenarios)	3 or 4	10
$\begin{matrix} Paging\ \ Load\ \ in\ \ LA_1 \\ Paging\ \ Load\ \ in\ \ LA_2 \end{matrix}$	$\begin{pmatrix} 32\% \\ 18\% \end{pmatrix}$	$\begin{pmatrix} \simeq 15\% \\ \simeq 28\% \end{pmatrix}$
Objective function Z_1 (Handover)	331681	28873
Objective function Z_2 (Distance)	1.49E8	1.024E8

3.5 Conclusion

We developed the Bi-objective Location Area Planning problem with a BTS-BSC connexions. This problem minimizes simultaneously the LU and the distance cost between BTS and BSC. We implemented a VEPSO algorithm that operates in two main steps, *Assignment of cells to LA* and *Assignment of BTS to BSC and BTS to LA*. *Assignment of cells to LA* is the initial population generated by construction, for the first swarm and generated randomly for the second swarm. Its resulting population is the input of the *Assignment of BTS to BSC and BTS to LA* process. This second step consists of taking the population solutions, one by one, get the BTS of the assigned cells, and then apply the TRX, Traffic and Links capacity constraints.
Our experimental analysis can be summarized into the following points:

- We deduced that the best scenario is to choose the initial cell C_{16} from BTS_{16} with a radius average size of 120000_m and with objective costs equal to 28873 and 1.02371309392E8 simultaneously for objective function Z_1 and Z_2. These two criteria present too much influence on solutions' quality.

- After implementing and testing our results of LA design with that of TUNISIANA represented in Figure 3.4 and 3.5, we note that the border of the new LA_1 would be expanded to include more BTS, giving a rate of paging load of ($\simeq$ 15%;$\simeq$ 28%) for LA_1 and LA_2 while they was fixed at ($\simeq$ 18%;$\simeq$ 32%). We are convinced that, if we increase the radius size, we will get better redefinition of the LAs that is more

suitable to our real case study.

- Also we see that the algorithm behavior depends mainly on the LA characteristic function. By behavior, we mean, number of iterations run, number of efficient solutions found, seed cells, radius of neighborhood etc.
- Neither the variation in the size of swarms nor the variation in the velocity V and in the inertia factor W affects the convergence to the set of efficient solutions. However, they have an impact on the number of efficient solutions and on the assignment in the second objective function.
- The seed cell and radius of neighborhood, have the more affect the behavior of our VEPSO algorithm, because they change the value of objective functions and the rate of paging load of each LA.

We can point out that our solution perform better than the one adopted by TUNISIANA. Therefore, our design of the subsystem will be adopted by Orascom Telecom in order to redistribute the set of cells into LAs with respects to system constrains.
The Pareto-optimal front of the proposed VEPSO is evident in terms of its diversity and quality. The results show the potential and efficiency of the proposed algorithm to solve BLD problem and produce multiple Pareto-optimal solutions in one simulation run.

General conclusion

A new model for the "Bi-objective Location area Design (BLD)" problem was constructed to determine the LA's division within two objective functions. This problem minimizes simultaneously LU and BTS-BSC distance cost with respect to the system constraints. Since radio resources are crucial for communication between customers, avoiding a lack in radio resources is an important benefit at our researches to reduce the network construction costs.

We formulated the BLD problem as a biobjective nonlinear programming model. This model was then solved by LINGO for a limited size range of the problem in order to get exact solutions.

As our problem is Bi-objective, we adopted the Vector Evaluated Particle Swarm Optimization (VEPSO) because it easily allows swarms to converge to the local area of Pareto Font with high quality solutions.

The VEPSO takes as input an initial seed cell around which region growth with a radius size, attempting to minimize LU and distance cost between BTS and BSC. Next, the algorithm improves the solution by performing global-best exchanges between the two swarms, allowing faster convergence to the Pareto-optimal solutions.

This algorithm operates with two swarms. Each swarm is exclusively evaluated with one of the objective functions and information obtained from other swarms are used to influence its movement in the solution space.

Our algorithm behavior depends mainly on the LA characteristic function. By behavior, we mean, number of iterations, number of efficient solutions found, seed cells, radius defining neighbor cells etc. After the simulation and variation of its components, we found that the BLD problem depends essentially on the location of the seed cell and the size of the radius.

Our model, approved by TUNISIANA, performs better than the existing solution. By increasing the radius size to 120000 with the seed cell C_{16}, we get better results; thus better assignment that are more suitable to the behavior of our LA. Therefore, we can conclude that we have formulated a general case of BLD problem that includes the majority of practical constrains.

Our work can be extended by comparing our approach to other metaheuristics such as tabu search or simulated annealing. Other constraints can also be included in our problem, such as engineering rules. Finally, as the 3G wireless network is also based on cell concept, we can apply our resolution approach to such a problem.

Appendix A

Computational experiment of the BLD

In this section, we will develop an example for our model trying to assign 3 cells into 2 LAs. We propose to solve this example by LINGO 9.0 (EXACT) in order to get exact solutions because we can not resolve it with a large number of variables. In that follows, we choose some numerical value for our inputs that are: the topology matrix A (Cell-BTS), Handover and λ_i^* for 3 cells, 2 LAs, 2 BTS, 2 BSC and one MSC.
Our developed model will be as follows:

$$\textbf{Min } Z_1 = w_{12}h_{12} + w_{13}h_{13} + w_{21}h_{21} + w_{23}h_{23} + w_{31}h_{31} + w_{32}h_{32}$$

$$\textbf{Min } Z_2 = y_{111}d_{11} + y_{211}d_{21} + y_{121}d_{12} + y_{221}d_{22} + y_{112}d_{11} + y_{212}d_{21}$$
$$+y_{122}d_{12} + y_{222}d_{22}$$

S.t.

Constraints (1)

$$x_{11} + x_{12} = 1 \qquad i = 1$$
$$x_{21} + x_{22} = 1 \qquad i = 2$$
$$x_{31} + x_{32} = 1 \qquad i = 3$$

Constraints (2)

$$y_{111} + y_{121} + y_{112} + y_{122} = 1 \qquad j = 1$$
$$y_{211} + y_{221} + y_{212} + y_{222} = 1 \qquad j = 1$$

Constraints (3)

$$w_{12} = 1 - (x_{11}x_{21} + x_{12}x_{22}) \qquad i = 1 \ s = 2$$
$$w_{13} = 1 - (x_{11}x_{31} + x_{12}x_{32}) \qquad i = 1 \ s = 3$$
$$w_{21} = 1 - (x_{21}x_{11} + x_{22}x_{12}) \qquad i = 2 \ s = 1$$
$$w_{23} = 1 - (x_{21}x_{31} + x_{22}x_{32}) \qquad i = 2 \ s = 3$$
$$w_{31} = 1 - (x_{31}x_{11} + x_{32}x_{12}) \qquad i = 3 \ s = 1$$
$$w_{32} = 1 - (x_{31}x_{21} + x_{32}x_{22}) \qquad i = 3 \ s = 2$$

Constraints (4)

$$(a^t_{11}.x_{11} + a^t_{21}.x_{21} + a^t_{31}.x_{31}) = y_{111} + y_{121} \qquad j = 1 \ r = 1$$
$$(a^t_{12}.x_{11} + a^t_{22}.x_{21} + a^t_{32}.x_{31}) = y_{211} + y_{221} \qquad j = 2 \ r = 1$$
$$(a^t_{11}.x_{12} + a^t_{21}.x_{22} + a^t_{31}.x_{32}) = y_{112} + y_{122} \qquad j = 1 \ r = 2$$
$$(a^t_{12}.x_{12} + a^t_{22}.x_{22} + a^t_{32}.x_{32}) = y_{212} + y_{222} \qquad j = 2 \ r = 2$$

Constraints (5)

$x_{i1} = 1 \quad or \quad 0$ $\qquad \forall i$

$x_{i2} = 1 \quad or \quad 0$ $\qquad \forall i$

Constraints (6)

$PB_1 - [x_{11}\lambda_1 + x_{21}\lambda_2 + x_{31}\lambda_3] \geq 0$ $\qquad r = 1$

$PB_2 - [x_{12}\lambda_1 + x_{22}\lambda_2 + x_{32}\lambda_3] \geq 0$ $\qquad r = 2$

Constraints (7)

$[a_{11}y_{111}c_1 + a_{21}y_{111}c_2 + a_{31}y_{111}c_3 + a_{12}y_{211}c_1 + a_{22}y_{211}c_2 + a_{32}y_{211}c_3]$
$+[a_{11}y_{112}c_1 + a_{21}y_{112}c_2 + a_{31}y_{112}c_3 + a_{12}y_{212}c_1 + a_{22}y_{212}c_2$
$+a_{32}y_{212}c_3] \leq C_1^{BSC}$ $\qquad k = 1$

$[a_{11}y_{121}c_1 + a_{21}y_{121}c_2 + a_{31}y_{121}c_3 + a_{12}y_{221}c_1 + a_{22}y_{221}c_2 + a_{32}y_{221}c_3]$
$+[a_{11}y_{122}c_1 + a_{21}y_{122}c_2 + a_{31}y_{122}c_3 + a_{12}y_{222}c_1 + a_{22}y_{222}c_2$
$+a_{32}y_{222}c_3] \leq C_1^{BSC}$ $\qquad k = 2$

Constraints (8)

$[a_{11}y_{111}r_1 + a_{21}y_{111}r_2 + a_{31}y_{111}r_3 + a_{12}y_{211}r_1 + a_{22}y_{211}r_2 + a_{32}y_{211}r_3]$
$+[a_{11}y_{112}r_1 + a_{21}y_{112}r_2 + a_{31}y_{112}r_3 + a_{12}y_{212}r_1 + a_{22}y_{212}r_2$
$+a_{32}y_{212}r_3] \leq R_1^{BSC}$ $\qquad k = 1$

$[a_{11}y_{121}r_1 + a_{21}y_{121}r_2 + a_{31}y_{121}r_3 + a_{12}y_{221}r_1 + a_{22}y_{221}r_2 + a_{32}y_{221}r_3]$
$+[a_{11}y_{122}r_1 + a_{21}y_{122}r_2 + a_{31}y_{122}r_3 + a_{12}y_{222}r_1 + a_{22}y_{222}r_2$
$+a_{32}y_{222}r_3] \leq R_1^{BSC}$ $\qquad k = 2$

Constraints (9)

$[a_{11}a_{21} + a_{12}a_{22}] - [x_{11}x_{21} + x_{12}x_{22}] \leq 0$ $\qquad i = 1 \ s = 2$

$[a_{11}a_{31} + a_{12}a_{32}] - [x_{11}x_{31} + x_{12}x_{32}] \leq 0$ $\qquad i = 1 \ s = 3$

$[a_{21}a_{11} + a_{22}a_{12}] - [x_{21}x_{11} + x_{22}x_{12}] \leq 0$ $\qquad i = 2 \ s = 1$

$[a_{21}a_{31} + a_{22}a_{32}] - [x_{21}x_{31} + x_{22}x_{32}] \leq 0$ $\qquad i = 2 \ s = 3$

$[a_{31}a_{11} + a_{32}a_{12}] - [x_{31}x_{11} + x_{32}x_{12}] \leq 0$ $\qquad i = 3 \ s = 1$

$[a_{31}a_{21} + a_{32}a_{22}] - [x_{31}x_{21} + x_{32}x_{22}] \leq 0$ $\qquad i = 3 \ s = 2$

Constraints (10)

$y_{111} + y_{211} + y_{112} + y_{212} \leq N_1^{BSC}$ $\quad k = 1$

$y_{121} + y_{221} + y_{122} + y_{222} \leq N_2^{BSC}$ $\quad k = 1$

Constraints (11 and 12)

$x_{11}, x_{21}, x_{31}, x_{12}, x_{22}, x_{32} \in \{0, 1\}$

$y_{111}, y_{121}, y_{112}, y_{122}, y_{211}, y_{221}, y_{212}, y_{222} \in \{0, 1\}$

Our inputs concerning cells are:

Cell	BTS	λ_i	TRX	Call traffic
C_1	1	13	3	13
C_2	2	15	2	24
C_3	1	8	2	15

We have for two different cells i and s, $h_{is} = h_{si}$.

Handover	Value
h_{12}	35
h_{13}	22
h_{23}	37

These are paging thresholds for the two new LAs:

Paging	Value
PB_1	20
PB_2	23

BSC maximum capacity are:

BSC	Maximum of Links	call traffic Capacity	TRX
BSC_1	2	75	12
BSC_2	3	70	15

Costs Values D_{jk}:

$$\begin{array}{c} \\ BTS_1 \\ BTS_2 \end{array} \begin{array}{c} \begin{array}{cc} BSC_1 & BSC_2 \end{array} \\ \begin{pmatrix} 20 & 31 \\ 36 & 10 \end{pmatrix} \end{array}$$

As inputs we have The matrix A as follows:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{cc} BTS_1 & BTS_2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$$

and the matrix C as follows:

$$\begin{array}{c} \\ BSC_1 \\ BSC_2 \end{array} \begin{array}{c} MSC \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{array}$$

As output, we will have the matrix X:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{cc} LA_1 & LA_2 \end{array} \\ \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{pmatrix} \end{array}$$

Then we derive matrix W:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \end{array}$$

And The Matrix Y That assigns BTS j to BSC k and BTS j to LA r.

In our example, for the first objective we have $S_1 \begin{pmatrix} Z_1 = 114 \\ Z_2 = 56 \end{pmatrix}$
a topology matrix X:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{cc} LA_1 & LA_2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

Then we derive matrix W:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

The decision variables for the matrix Y is: $Y_{1,1,2} = 1$ and $Y_{2,1,2} = 1$ and 0 otherwise.

For the second objective $S_2 \begin{pmatrix} Z_1 = 144 \\ Z_2 = 30 \end{pmatrix}$
a topology matrix X:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{cc} LA_1 & LA_2 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

Then we derive matrix W:

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

The decision variables for the matrix Y is:
$Y_{1,1,2} = 1$ and $Y_{2,2,1} = 1$ and 0 otherwise.

For this small example We remarque that the two objective are conflictual. If we minimize one, the other increase and vice versa.

Bibliography

[1] A. Merchant, B. Sengupta, 2005. Assignment of cells to switches in PCS networks, IEEE/ACM Transactions on Networking.

[2] Alberto García Villoria, Rafael Pastor, 2009. Introducing dynamic diversity into a discrete particle swarm optimization. Computers & Operations Research 36,951-966.

[3] Alejandro Quintero, Oscar Garcia, Samuel Pierre, 2004. An alternative strategy for location update and paging in mobile networks. Computer Communications 27, 1509-1523.

[4] Caselli.A, Righini.G, 2006. A Branch-and-Price Algorithm for the Multilevel Generalized Assignment Problem.

[5] Ching-Jong Liao, Chao-Tang Tseng, Pin Luarn, 2007. A discrete version of particle swarm optimization for flowshop scheduling problems. Computers & Operations Research 34, 3099-3111.

[6] Chun-hyun Paik, Samit Soni, 2006.A simulated annealing based solution approach for the two-layered location registration and paging areas partitioning problem in cellular mobile networks. European Journal of Operational Research 178, 579-594.

[7] Debashis Saha, Partha S. Bhattacharjee, Amitava Mukherjee, 2007. Time efficient heuristics for cell-to-switch assignment in quasi-static/dynamic location area planning of mobile cellular networks. Computer Communications 30, 326-340.

[8] G.-E. Amoussou, G. Pesant, S. Pierre, 2001. Affectation de cellules des commutateurs par programmation par contraintes, IEEE CCECE2001.

[9] G.-E. Amoussou, M. André, G. Pesant, S. Pierre, 2003. Une approche basée sur la programmation par contraintes pour affecter des cellules des commutateurs dans les réseaux cellulaires pour mobiles, Annales des Télé communications.

[10] Global System for Mobile Communication (GSM). The International Engineering Consortium. Web ProForum Tutorials
http://www.iec.orghttp.

[11] Geir Dahl, Njal Foldnes, 2006. LP based heuristics for the multiple knapsack problem with assignment restrictions.

[12] Haluk Yapicioglu, Alice E. Smith, Gerry DozierSolving, 2007. Solving the semi-desirable facility location problem using bi-objective particle swarm. European Journal of Operational Research 177, 733-749.

[13] Hassan.R, Cohanim.B, Weck.O.D, Venter.G, 2005. A comparision of Particle Swarm Optimization and the Genetic algorithm. American Institute of Aeronautics and Astronautics.

[14] Ílker Demirkol, Ufuk cağlayan, Hakan Deli, Cem Ersoy, 2004. Location Area Planning And Cell To Switch Assignment In Cellular Networks. IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 3, NO. 3, MAY 2004.

[15] John Scourias, 1996. Overview of GSM: The Global System for Mobile Communications. March 13 1996.

[16] Joseph R.L. Fournier, Samuel Pierre, 2005. Assigning cells to switches in mobile networks using an ant colony optimization heuristic. Computer Communications 28, 65-73.

[17] J.G. Markoulidakis, E.D. Sykas, 1993. Method for efficient location area planning in mobile telecommunications. ELECTRONICS LETERS 9th December 7993 Vol. 29 No. 25.

[18] John G. Vlachogiannis, Kwang Y. Lee, 2009. Multi-objective based on parallel vector evaluated particle swarm optimization for optimal steady-state performance of

power systems. Expert Systems with Applications, ESWA 3576, No. of Pages 7, Model 5G, 17 March 2009.

[19] Kyung Hee Kim, Jang Hyun Baek, Che Soong Kim, 2007. Modeling and optimization of zone-based registration considering cell-by-cell location area for mobile communication networks. Computer Networks 52, 667-674.

[20] K.W. Chau, 2006. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. Journal of Hydrology 329, 363-367.

[21] Location Areas Planning and Optimisation, MobiNetS (Mobile Networks Solutions-Alcatel), Online's TUNISIANA document, Orascom Telecom Tunisie (OTT).

[22] Mark Vroblefski, Evelyn C.Brown, 2006. A grouping genetic algorithm for registration area planning. Omega: The International Journal of Management Science 34, 220-230.

[23] M. André, G. Pesant, S. et Pierre, 2002. Affectation de cellules des commutateurs de réseaux cellulaires mobiles par programmation par contraintes, IEEE Canadian Conference on Electrical and Computer Engineering.

[24] Mireille Palpant, Cristian Oliva, Christian Artigues, Philippe Michelon and Mohamed Didi Biha, 2008. Models and methods for frequency assignment with cumulative interference constraints. International Transactions in Operational Research.

[25] Öncan Temel, 2007. The Survey of the Generalized Assignment Problem and its applications. INFOR; Aug 2007; 45, 3; ABI/INFORM Global pg. 123.

[26] Paging process, MobiNetS (Mobile Networks Solutions-Alcatel), Online's TUNISIANA Document, Orascom Telecom Tunisie (OTT).

[27] Peng-Yeng Yin, Shiuh-Sheng Yu, Pei-Pei Wang, Yi-Te Wang, 2007. Multi-objective task allocation in distributed computing systems by hybrid particle swarm optimization. Applied Mathematics and Computation 184, 407-420.

[28] Peng-Yeng YinT, Shiuh-Sheng Yu, Pei-Pei Wang, Yi-Te Wang, 2006. A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems. Computer Standards & Interfaces 28, 441-450.

[29] P. Demestichas, N. Georgantas, E. Tzifa, V. Demesticha, M. Striki, M. Kilanioti, M. Theologou, 2000. Computationally efficient algorithms for location area planning in future cellular systems. Computer Communications 23, 1263-1280.

[30] Pierre DuPont, 1998. An Optimized Paging Procedure For GSM Packet Data. Motorola, Technical Developments, P 76-77.

[31] Przybylski.A, Gandibleux.X, Ehrgott.M, 2007. Two phase algorithms for the bi-objective assignment problem. European Journal of Operational Research.

[32] Roberto Battiti, Alan A. Bertossi, Mauro Brunato, 2001. Cellular Channel Assignment: A new Localized and Distributed Strategy. Mobile Networks and Applications.

[33] Shi-Wu Lo, Tei-Wei Kuo, Kam-Yiu Lam, Guo-Hui Li, 2004. Efficient location area planning for cellular networks with hierarchical location databases. Computer Networks 45, 715-730.

[34] S.N. Omkar, Dheevatsa Mudigere, G. Narayana Naik, S. Gopalakrishnan, 2008. Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures. Computers and Structures 86, 1-14.

[35] Swarup Mandal, Debashis Saha, Ambuj Mahanti, Parag C. Pendharkar, 2007. Cell-to-switch level planning in mobile wireless networks for efficient management of radio resource. Omega 35, 697-705.

[36] Shyong Jian Shyu, B.M.T. Lin, Tsung-Shen Hsiao, 2005. Ant colony optimization for the cell assignment problem in PCS networks. Computers & Operations Research 33, 1713-1740.

[37] Souheyl Touhami, Jean-Marie Bourjolly et Gilbert Laporte, 2009. Partial integration of frequency allocation within antenna positioning in GSM mobile networks. European Journal of Operational Research, 541551.

[38] T.M. Chan, S. Kwong, K.F. Man, K.S. Tang, 2002. Hard handoff minimization using genetic algorithms. Signal Processing 82, 1047 1058.

[39] Tabitha James, Mark Vroblefski, Quinton Nottingham, 2007. A hybrid grouping genetic algorithm for the registration area planning problem. Computer Communications 30, 2180-2190.

[40] Teghem.J, Tuyttens.D, Ulungu.E.L, 2000. An interactive heuristic method for multi-objective combinatorial optimization. Computers & Operations Research 27, 621-634.

[41] Tsan-Pin Wang, Shu-Yuen Hwang, and Chien-Chao Tseng, 1998. Registration Area Planning for PCS Networks Using Genetic Algorithms. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 47, NO. 3, AUGUST 1998.

[42] Yujia Wang, Yupu Yang, 2009. Particle swarm with equilibrium strategy of selection for multi-objective optimization. European Journal of Operational Research.

[43] Y.H. Chew, B.S. Yeo, D.C.M. Kuan, 2007. Sensitivity study of location management area partitioning in cellular communication systems. Computer Networks 51, 1555-1573. Research.