Solving channel assignment problem for mobile communication with soft computing

Gu, Wen

2005

Gu, W. (2005). Solving channel assignment problem for mobile communication with soft computing. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4328>

<https://doi.org/10.32657/10356/4328>

Nanyang Technological University

Downloaded on 26 Apr 2023 19:40:30 SGT

Solving Channel Assignment Problem for Mobile Communication with Soft Computing

Gu Wen

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University in
fulfillment of the requirement for the degree of
Master of Engineering

2005

TK
5103.2
G896
2005

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other university or institution.

.....
Date

.....
Gu Wen

Acknowledgements

First, I would like to express my deepest appreciation and sincere gratitude to my supervisor, Associate Professor Lipo Wang, for his professional and invaluable guidance during my research towards master degree. Within these two years, I have learnt much from him, not only knowledge but also the attitude toward science. Without his supervision and help, I can not complete this thesis. I also thank SingTel for the joint research project and Mr. Foo Shih Suan, Mr. Bey Kok Meng and Mr. Yang Chia Lieh from SingTel for their help and discussion.

I would also like to sincerely thank my parents and my girl friend, Zhao Jing, for their support, understanding and encouragement.

Last but not least, I would like to thank all the staff and students in Information System Research Lab and Software Engineering Lab, for their help.

Summary

In industry, agriculture and economy, we often encounter constrained optimization problems. One common way to deal with these problems is to transform the constrained optimization problems into unconstrained ones by introducing a penalty function, which is controlled by a number of penalty coefficients into the objective function. . However, how to set the penalty coefficients turns out to be a difficult problem itself [16]! Stochastic ranking is a novel constraint-handling technique proposed by P.T. Runarsson and Xin Yao. Applying this technique for constrained evolutionary make us do not need to determine the hard-setting penalty coefficients. We demonstrate this novel approach with one difficult combinatorial optimization problem, i.e., the channel assignment problem for cellular mobile communications.

In recent years, there is a steady growth in demand for cellular mobile communications. Since the usable frequency spectrum is limited, optimal assignment of frequency channels is becoming more and more important. Careful design of network can greatly enhance the traffic capacity of a cellular system and decrease interference between calls, thereby improving service quality and customer satisfaction. A competitive communication industry also makes the network design to accommodate

certain performance criteria. Operators of such networks need to assign channels to base stations, in a way that quality of service is guaranteed, i.e., the probability that an unused channel can be used for a new call is sufficiently high and the probability that the signal-to-interference-ratio (S/I) falls below a predefined threshold is sufficiently low. In this thesis, we are concerned with the problem of assigning calls in a cellular mobile network to frequency channels in such a way that interference between calls is minimized, while the demands for channels in each cell are satisfied. It has been shown that the channel assignment problem is a very difficult optimization problem. We propose to solve the channel assignment problem by combining stochastic ranking with genetic algorithm (GA-SR) and compare the results with other algorithms in several sets of benchmarking channel assignment problems. Simulation results showed that our approach could improve on results obtained by many other algorithms.

In recent NTU-SingTel joint research project, which aims at finding an optimal channel assignment for SingTel mobile networks through large-scale computer simulations, we use genetic algorithm, a simple computer model for the evolutionary process in which species evolve to improve themselves. The outcome of this project has been tested by SingTel in a more realistic operating environment.

Contents

Acknowledgements	i
Summary	ii
Table of Contents	iv
List of Figures	vii
List of Tables	viii
List of Abbreviations	x
1 Introduction	1
1.1 Channel Assignment Problem	1
1.2 Motivation	3
1.3 Objective	5
1.4 Major Contributions of the Thesis	6
1.5 Organization of the Thesis	8
2 Channel Assignment Problems (CAPs)	10
2.1 The Review of CAPs	10
2.2 The Model of SCA	13
2.2.1 The Problem Description of SCA	13
2.2.2 The Mathematical Formulation of CAP2	15
2.3 The Benchmarking CAP2s	17
2.4 Summary	23

3 Review of Stochastic Ranking	24
3.1 Introduction	24
3.2 Constraint Handling by Stochastic Ranking	27
3.2.1 Penalty Method	27
3.2.2 Stochastic Ranking	30
3.3 Conclusion	33
4 Basics of Genetic Algorithms (GAs)	35
4.1 Introduction	35
4.2 Basics of Genetic Algorithm	36
4.2.1 Biological Background	36
4.2.2 Principle of Genetic Algorithm	37
4.2.3 Selection	38
4.2.4 Operators of GA	39
4.2.5 Parameters of GA	41
4.3 The Summary	42
5 Solving CAP2s with GA-SR	43
5.1 Introduction	43
5.2 Implementation of Stochastic Ranking to CAP2s	46
5.3 The Simulation Results and Comparisons with Various Algorithms	47
5.4 Conclusions	62
6 Apply Genetic Algorithm on NTU-SingTel Project	63
6.1 Introduction	63
6.2 The DataSet of NTU-SingTel Project	64
6.3 Implementation of GA on NTU-SingTel joint Research Project	68
6.3.1 Choices of GA Parameters	70
6.3.2 Simulation Results	70

6.4 Conclusions71

7 Conclusions and Recommendations 74

7.1 Conclusions74

7.2 Recommendations for Future Research 76

Bibliography 78

Author’s Publications 91

List of Figures

2.1 An interference-free assignment for a 4 cell and 11 channel network. 14

2.2 An assignment for a 4 cell and 10 channel network 15

2.3 The string to solve channel assignment problem 15

2.4 Layout of a 21-cell hexagonal network 18

4.1 A canonical genetic algorithm 36

5.1 Flowchart for overall implementation 45

5.2 P_f versus interference with constant population size, P_c and P_m (HEX) 51

5.3 P_f versus interference with constant population size, P_c and P_m (KUNZ) 52

6.1 A simplified layout of macro layer 66

List of Tables

2.1 Descriptions for hexagonal problems	19
2.2 Descriptions for KUNZ problems	22
4.1 the comparison of natural and GA terminology	37
4.2 Sample of chromosomes	40
4.3 Sample of crossover	40
4.4 Sample of mutation	41
5.1 Parameters used in the simulations	48
5.2 The percentage of feasible individuals in the final population	48
5.3 Comparison of the total interference obtained by different methods	49
5.4 Experimental results on 10 benchmark problems	50
5.5 The minimum interference values and time used	50
5.6 Experimental results on 10 benchmark with fixed $P_f(P_f=0.46)$	52
5.7 Channel assignment for EX1 problem with interference 0	53
5.8 Channel assignment for EX2 problem with interference 0	53
5.9 Channel assignment for HEX1 problem with interference 46	54
5.10 Channel assignment for HEX2 problem with interference 17	55
5.11 Channel assignment for HEX3 problem with interference 76	56
5.12 Channel assignment for HEX4 problem with interference 16	57
5.13 Channel assignment for KUNZ1 problem with interference 19	58
5.14 Channel assignment for KUNZ2 problem with interference 29	59
5.15 Channel assignment for KUNZ3 problem with interference 13	60
5.16 Channel assignment for KUNZ4 problem with interference 0	61

6.1 Descriptions for SingTel data sets 65

6.2 A simplified interference table of cell 1 in macro layer66

6.3 The parameters of genetic algorithm 70

6.4 The number of cells affected by interference. 71

6.5 Channel assignment for macro layer72

6.6 Channel assignment for micro layer73

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
CAP	Channel Assignment Problem
S/I	Signal-to-interference-ratio
SCA	Static Channel Assignment
DCA	Dynamic Channel Assignment
SA	Simulated Annealing
GA	Genetic Algorithm
SR	Stochastic Ranking
GA-SR	Genetic Algorithm with Stochastic Ranking
CCC	Co-channel Constraint
ACC	Adjacent Channel Constraint
CSC	Co-site Constraint
SD	Steepest Descent
SSA	Stochastic Simulated Annealing
HN	Hopfield Network

HCHN	Hill-Climbing Hopfield Network
SONN	Self-Organizing Neural Network
MSO-TCNN	Transiently Chaotic Neural Network
SCSA	Stochastic Chaotic Simulated Annealing

Chapter 1

Introduction

1.1 Channel Assignment Problem (CAP)

Over recent years, there is a steady increase in the popularity of cellular mobile communication systems. However, the electromagnetic spectrum available for this purpose is a limited resource, which places severe limitations on the size and performance of such systems. Careful design of a network can greatly enhance the traffic capacity of a cellular system and decrease interference between calls, thereby improving service quality and customer satisfaction. A competitive communication industry also forces the network design to accommodate certain performance criteria. Operators of such networks need to assign channels (frequencies) to base stations in such a way that the quality of service is guaranteed, i.e., the probability that an idle channel can be found for an new incoming call is sufficiently high and the probability

that the signal-to-interference-ratio (S/I) falls below certain predefined limit is sufficiently low.

The Channel assignment problem (CAP) concerns with assigning the required number of channels to each region in such a way that interference between calls is minimized, while the frequency spectrum is used efficiently. The process of channel assignment can be achieved by two steps. The first step is the design process, in which the future traffic is forecast and the assignments are made to satisfy the requirements for the immediate future. Since channels are assigned only once and remains fixed, this step can be called a static channel assignment (SCA). However, as the traffic changes, it will become more and more difficult to maintain performance standards with this network. The second step, a more dynamic assignment strategy, will be employed to permit rearrangement of assignments in an on-going adaptive environment. Any dynamic channel assignment (DCA) depends on the success or otherwise of the initial SCA to a large extent. That is why a lot of research has focused on solving static channel assignment problems. It is well known that the SCA problem belongs to a class of difficult combinatorial optimization problems named *NP*-hard.

In practice, we may meet many problems, activities and resources that are combinatorial optimization problems. Solving optimization problems in many fields can produce economic benefit. This is the reason why a combinatorial optimization method searches for the best solution out of all possible solutions. Generally, among all the possible solutions, the best one should have the lowest cost or the highest profit and combinatorial optimization is the process of looking for one or more optimal solutions in a well-defined discrete search space [89]. Many combinatorial optimization problems are hard to deal with because they belong to a class of very difficult problems, *NP* (nondeterministic polynomial)-hard class, that is a terminology of computational complexity [43]. If every problem in *NP* is polynomially reducible to a problem *Q*, then

Q is called *NP*-hard. Furthermore, if every problem in *NP* is polynomially transformable to a problem R , R is *NP*-complete, which is a class of the hardest problems in *NP* [43].

Many exact methods have been proposed in order to find an optimal solution for combinatorial optimizations, such as cutting plane methods, dynamic programming based on the bellman's principle of optimality, branch and bound methods, and so on. Unfortunately, no exact polynomial algorithm has been found for any *NP*-hard problem up to now [43]. In the circumstances, most researchers have been trying to find approximate or heuristic algorithms that explore near-optimal solutions at a computational cost without ensuring optimality or feasibility [44]. These algorithms have achieved significant success when applied to deal with CAPs.

Gamst and Rave defined a class of SCA in an arbitrary inhomogeneous cellular radio network [4]. That is to minimize the span of channels for an interference-free assignment while the demand in each cell is satisfied. This class of SCA is denoted as CAP1 in [2]. In another case that the number of available channels is smaller than the lower bound for an interference-free assignment, there is little point trying to find a solution. An alternative and more practical version of the channel assignment problem is CAP2, which is to minimize the severity of interferences while subject to the channel demand constraints.

1.2 Motivation

It is well known that if an *NP*-hard problem can be solved in polynomial time, then its corresponding decision problem can also be solved in polynomial time and as a result, if $P \neq NP$, then any *NP* problem whose corresponding decision problem is *NP*-complete is not solvable in polynomial time [89]. Hence, to deal with CAPs, we have to give up

the optimality and try to find approximate solutions that are computable in polynomial time. Many approximation algorithms have been developed so far to solve channel assignment problems, including simulated annealing (SA), genetic algorithms (GA), Tabu search and neural network. Simulated annealing is a technique inspired from Monte Carlo methods in statistical mechanics [53]. In order to avoid local optima, simulated annealing takes non-locally optimal steps probabilistically in the search space. The probability of taking such steps varies with the “temperature” of the system. The “temperature” reduces with time in file. SA not only can deal with evaluation functions with quite arbitrary degrees of nonlinearities, discontinuities and stochasticity, but also can process quite arbitrary boundary conditions and constraints, which are imposed to those evaluations. It has been shown that with a “large enough” initial temperature and a proper annealing schedule, SA guarantees a globally optimal solution [54]. However, though SA has been systematically used to solve many forms of combinatorial optimization problems as a general algorithm, it has the problems of run time efficiency and solution quality [44], [53]. Hence, it is necessary to design the appropriate neighborhood structures carefully to overcome those drawbacks. Up to now, another algorithm, neural network has shown its power in many fields to solve combinatorial optimizations [55] [56] [57] [58]. The Hopfield neural network is a typical model of artificial neural network and it has symmetric connection weights [59] [60] [55]. It guarantees the convergence to a stable equilibrium point due to its gradient descent dynamics. However, the main shortcoming is that it suffers from local convergence [43].

Genetic algorithms (GAs) have received a great deal of attention because of their ability to search for optimal solutions. They are based on the mechanics of natural selection and genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with

some of the innovative flair of human search [29]. The genetic algorithm was firstly developed by John H. Holland in the 1960's [30]. It allows computers to solve difficult problems [29], [31].

CAP2 can be formulated as a constrained optimization problem, as shown in [2]. One common way often used to deal with constrained optimization problem is to introduce a penalty term into the objective function to penalize constraint violations [17] thereby transforming a constrained optimization problem into an unconstrained one. However, how to set the penalty coefficients becomes a difficult problem itself! Runarsson and Yao [16] proposed a constraint-handling technique, stochastic ranking (*SR*), which can balance the objective and penalty functions conveniently without setting the penalty coefficients. Based on the characteristics of *GA* and *SR*, we combine them together (*GA-SR*). To demonstrate the effectiveness of the *GA-SR*, we apply it to solve the CAPs for cellular mobile communications.

1.3 Objective

Optimal channel assignment can enhance traffic capacity of a cellular mobile network and decrease interference between calls, thereby improving service quality and customer satisfaction. We combine genetic algorithms with stochastic ranking, an approach for balancing objective and penalty functions, proposed by Runarsson and Yao, to solve the problem of assigning calls in a cellular mobile network to frequency channels in such a way that interference between calls is minimized, while demands for channels are satisfied. CAPs can be divided into two classes, Static Channel Assignment (*SCA*) and Dynamic channel assignment (*DCA*). Our research is focused on static channel assignment problems since static assignment is the basis of dynamic assignment to a large extent. There are two types of *SCA*, CAP1 [4] and CAP2 [2].

CAP1 is to minimize the span of channels while the demands and interference-free constraints are satisfied; CAP2 is to minimize the severity of interferences and subject to the demand constraints. In practice, the number of available channels is far smaller than the interference-free lower bound. Hence, we concentrate on CAP2 only in this thesis. It has been shown that CAP is a generalized graph-coloring problem [33]. Therefore, it is NP-hard. By using the simple extension to the genetic algorithm (GA-SR), i.e., the use of the stochastic ranking, we showed that GA-SR is more effective than many other methods such as Hopfield neural network, stochastic simulated annealing.

For the NTU-SingTel joint research project, we proposed novel techniques to achieve practical optimal channel assignment that is suitable for SingTel mobile networks through large-scale computer simulations and incorporates the specifications and requirements from SingTel mobile network. The data set provided by SingTel includes interference tables and traffic information, etc. After we transforming them into the compatibility matrices C (where C_{ij} is the minimum frequency separation between channels assigned to cell i and cell j), we found the matrices C are not symmetric. Hence, we cannot use the algorithms based on Hopfield neural networks in this case. Inspired from evolutionary processes in biological species, genetic algorithms (GA) have shown their power to search for optimal solutions in various optimization problems. We applied GA to this project and good channel assignments were obtained in both macro cell and micro cell layers of the mobile network.

1.4 Major Contributions of the Thesis

The major contributions are summarized as follows:

- **Potential of GA techniques in telecommunication industry:** Stochastic ranking

was proposed by Runarsson and Yao [16], which is a constraint-handling technique. In [16], SR was combined with evolution strategies (ES) to deal with constrained continuous optimization problems. The fact that SR combined with ES can achieve good results on 13 benchmark problems motivates us to apply it in combinatorial optimization. In this thesis, we proposed a novel algorithm, GA-SR, which combines the genetic algorithm (GA) and stochastic ranking (SR). This is the first application of the SR in the CAPs. Since the usual operators, mutation and recombination are 'blind' to constraints, when solving problems with large size, a large number of infeasible solutions are generated, which is inefficient. We introduced a threshold λ to control the number of feasible solutions. That is, when the percentage of feasible solutions in the whole population falls below λ , the crossover site and mutation site will be chosen specially to generate feasible solution. Experimental results showed that our GA-SR could handle the constraints conveniently and could achieve better results than many other existing algorithms. Due to its good performance, GA-SR shown in this thesis is of great importance to the telecommunication industry, not limited to only the CAP for cellular mobile networks, but also in satellite and other systems where the resource is finite. In addition, the optimization techniques investigated here can also be applied to many other combinatorial optimization problems, such as optimal CDMA multi-user detection, optimal broadcast scheduling in packet radio networks, and optimal beam forming and equalization.

- **Implementation of GA on NTU-SingTel research project:** In practical cellular mobile network, the interferences between two cells are not exactly reciprocal, i.e., the compatibility matrix is not symmetric. Genetic algorithm is used to solve the channel assignment problem for SingTel network. It has shown that GA is very useful in finding feasible frequency assignments for practical cellular mobile. We

also proposed a method to transform interference tables to compatibility matrices. The experimental results showed that our methods could reduce the number of cells, in which interference exists. The outcomes of this project have been tested by SingTel in a realistic environment.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows.

In chapter 2, the concept of channel assignment problem is introduced and the existing techniques for solving CAP are surveyed briefly. Three potential interferences, the co-channel interference, the adjacent-interference and the co-site interference are considered. Then, we use the model of the static channel assignment problem, which is reformulated as a generalized quadratic assignment problem [2]. The interference free constraint is treated as the soft constraints in the objective function. The demand constraint is treated as hard constraints in the penalty function. We map each potential solution of static CAP2 onto a chromosome with $N \times M$ genes. We adopt a cost tensor proposed by Smith and Palniswami to measure the degree of interference between cells [2].

Chapter 3 reviews stochastic ranking, a constraint-handling technique proposed Runarsson and Yao [16] to balance the dominance of objective and penalty function. Common approaches are discussed at the beginning of this chapter and the disadvantages of common methods are analyzed. Then, the efficiency of the new constraint-handling technique is stated and the theoretical explanation is given.

In chapter 4, we briefly review the genetic algorithms. The principles of genetic algorithms are reviewed. Three parameters (crossover probability, mutation probability and population size) and three operators (crossover, mutation and selection) are also

introduced in this part.

The simulation results of GA-SR are described in chapter 5. Three main sets of test data are used to compare the performances of GA-SR with other existing techniques.

In chapter 6, the NTU-SingTel joint research project is described. In this chapter, we introduce the problems on different layers of SingTel mobile network. We use genetic algorithm to solve channel assignment problems on macro and micro layer because it does not have the nonsymmetrical limitation and has a high probability for leading to good solutions. We transform the original data set, i.e., practical interference and the area of each cell, into the compatibility matrices. The results of this project have been tested by SingTel company in a more realistic operating environment.

Conclusions and recommendations for further research work are discussed in Chapter 7.

Chapter 2

Channel Assignment Problems (CAPs)

2.1 The Review of CAPs

The number of users in mobile communications continues to grow; however, the number of usable channels (frequencies), which are necessary for the communication between mobile users and the base stations of cellular radio networks, is very limited. Hence, it is necessary to design a network carefully in order to ensure an acceptable signal-to-interference (S/I) ratio. The channel assignment problem (CAP) concerns with assigning calls in a cellular mobile network to frequency channels in such a way that interference between calls is minimized, while demands for channels are satisfied [1] – [15]. Optimal channel assignment can enhance traffic capacity of a cellular mobile network and decrease interference between calls, thereby improving service quality and customer satisfaction.

There are basically three sources of interference, namely:

- *The Co-channel Constraint (CCC)*: The interference is caused by another caller within some geographical range using the same channel. That means the same frequency cannot be assigned to certain pairs of radio cells simultaneously;

- *The Adjacent Channel Constraint (ACC)*: The interference is caused by another caller in an adjacent geographical region using an adjacent channel. That means channels adjacent in the frequency domain can not be assigned to adjacent cells simultaneously;
- *The Co-Site Constraint (CSC)*: The interference is caused by another caller within the same geographical region using another channel. That means any pair of frequencies assigned to a radio cell must have certain distance in the frequency domain.

The radio frequency propagation (obtained from the regional topography and morph structure) and the spatial density of the expected traffic can determine the cells and channels, which may cause interference. Specialized program have been developed for the purpose of calculating these relationships [32]. The demand for channel in a certain region can also be calculated by the expected traffic.

Gamst and Rave defined a class of SCA in an arbitrary inhomogeneous cellular radio network [4]. That is to minimize the span of channels for an interference-free assignment while the demand in each cell is satisfied. This class of SCA is denoted as CAP1 in [2]. Graph coloring algorithms [7], [33] are usually used to solve CAP1. A vertex in the graph is used to represent a certain call in the mobile network and an edge connects two vertices if interference exists between corresponding calls. Each edge is tagged with the minimum separation C_{ij} by which two channels assigned to cell i and cell j must be separated. Thus CAP1 can be transformed to assign positive integer to vertices of the graph such that if two vertices are connected by an edge, the absolute value of the difference of the integers used is at least equal to the edge label. The maximum integer used should be as small as possible. If no adjacent constraints are considered, i.e. $C_{ij} \leq 1$, the CAP1 can be simplified to the classical graph coloring problem, which is NP-complete [34]. Up to the present, a lot of heuristics have been

devised to give near-optimal solutions to CAP1. Many of them can arrive at a value for the number of channels, which is above the minimum number needed. However, most of them do not indicate of how far from optimal their solutions are. Gamst [1] defined a lower bound of channels required for an interference free assignment, which can serve as a supplement to these heuristics.

In another case that the number of available channels is smaller than the lower bound, there is little point trying to find a solution. An alternative and more practical version of the channel assignment problem is CAP2. Kunz used the Hopfield neural network to solve a different CAP [5], which is to minimize the severity of interferences while subject to the channel demand constraints. Since in practice, the number of channels available is much lower than the lower bound, we are concerned with CAP2 only.

Various techniques have been applied to solve CAPs [1]-[15], [31]-[42]. KUNZ [5] implemented the continuous Hopfield network with an energy function containing the objective function and individual terms for all constraints of the problem. Funabiki *et al.* used a parallel algorithm to solve CAP [8], which does not require a rigorous synchronization procedure. In [13], a heuristic was used to solve CAPs. The channels were ranked according to how difficult it would be to assign that channel to a base station. This approach was also adopted by [7], namely a sequential cell ordering algorithm. M. Duque-Anton *et al.* applied simulated annealing to solve CAPs in [4]. There have been several attempts in using GA to deal with CAPs. For example, [42], [93]-[95] used GA for CAP1. [31], [37] and [86] used GA for CAP2.

Generally, two steps are needed to achieve the proper assignment in a cellular mobile network [2]. The first step is the design process, in which the future traffic is forecast and the assignments are made to satisfy the requirements for the immediate future. This process is called a static channel assignment (SCA) since the assignment of

channels to regions is made only once and remains fixed. As the traffic varies, however, it will become more difficult to maintain the network performance. A more dynamic assignment strategy will need to be employed which permits rearrangement of assignments in an on-going adaptive environment [2]. The task of any dynamic channel assignment (DCA) procedure is dependent, to a large extent, on the success or otherwise of the initial SCA [2]. That is why we focus our research on the static design problem in this thesis.

2.2 The Model of SCA

In this thesis, we use the same format of SCA problem as in [2], which is a generalized quadratic assignment problem. The noninterference constraints are treated as soft constraints in the objective function and the demand satisfaction constraints are treated as hard constraints in the penalty function. This approach is particularly useful if the demands and constraints are such that no interference-free solution exists.

2.2.1 The Problem Description of SCA

Let us consider a network of N cells and M channels. The channels required (expected traffic) for cell i is given by D_i ($i = 1, 2, \dots, N$). $D^T = (D_1, D_2, \dots, D_N)$ is called the demand vector. We suppose that the RF propagation and the spatial density of the expected traffic have already been calculated and the non-interference constraints can thus be determined (perhaps using a program like GRAND [32]). The electromagnetic compatibility (EMC) constraints specify the minimum distance in the frequency domain by which two channels must be separated so that no interference exists within the regions to which the channels have been assigned. These minimum distances are stored in a symmetric compatibility matrix C that has a dimension of $N \times N$. Each element

C_{ij} represents the minimum frequency separation between two channels assigned to cell i and cell j .

We use EX1 data set as an example. The number of cells is $N=4$ and the demand for channels in each of cells is $D^T = (1,1,1,3)$. The compatibility matrix is given as [2]:

$$C = \begin{pmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{pmatrix}$$

The diagonal term $C_{ii} = 5$ means that any two channels assigned to cell i must be at least five channels apart in order that no co-site interference occurs. Channels assigned to cells 1 and 2 must be four frequencies apart. Off-diagonal terms of $C_{ij} = 1$ and $C_{ij} = 2$ correspond to CCC and ACC [2]. We can obtain the interference-free assignments for $M = 11$. The channel assignment problems, as demonstrated by this example, are CAP1 [2]:

Minimize span of channels
Subject to demand and noninterference constraints

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											
4											

Figure 2.1 an interference-free assignment for a 4 cell and 11 channel network

We can see that $M=11$ is the lower bound for this example, as shown in Figure 2.1. If the number of available channel M is less than 11, we cannot find any interference-free assignments. Let us suppose there are only ten channels available. It is impossible to find an interference-free assignment. Figure 2.2 shows an assignment for a 4 cell and 10

channel network.

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										

Figure 2.2 an assignment for a 4 cell and 10 channel network

In this case, we need solve CAP2 [2]:

Minimize

severity of interferences

Subject to

demand constraints

The purpose of CAP2 is to reduce the severity of the interference for the network. This model is much more useful in practical cases where the number of channels available is not enough for an interference-free assignment. In the case that the number of available channels is greater than the lower bound, CAP1 should be considered to make efficient use of the spectrum.

2.2.2 The Mathematical Formulation of CAP2

One solution to CAP2 is mapped onto a string with $N \times M$ bits as shown in Figure 2.3.

Figure 2.3 the string to solve channel assignment problem

The mathematical formulation of CAP2 given by Smith and Palaniswami [2] is reviewed below.

$$U_{j,k} = \begin{cases} 1, & \text{if cell } j \text{ is assigned with channel } k, \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

for $j = 1, \dots, N$ and $k = 1, \dots, M$. For example, if $U_{j,k} = U_{i,l} = 1$, that means channels k and l are assigned to cells j and i , respectively. The degree of interference caused by such assignments can be measured by weighting each assignment by an element in a cost tensor $Q_{j,i,m+1}$ where $m = |k - l|$ represents the distance (in the channel domain) between channels k and l [2]. When $k = l$, the interference cost should be at its maximum [2]. The cost decreases until the two channels are far enough apart that no interference exists [2]. Now, the problem becomes how to minimize the total cost of all the assignments in the network.

Minimize

$$\text{Interf}(U) = \sum_{j=1}^N \sum_{k=1}^M U_{j,k} \sum_{i=1}^N \sum_{l=1}^M U_{i,l} Q_{j,i,|k-l|+1}, \quad (2.2)$$

Subject to

$$\sum_{k=1}^M U_{j,k} = D_j, \quad \forall j = 1, \dots, N, \quad (2.3)$$

A recursive relation, (2.4)-(2.6) can be used to generate the proximity factor tensor Q described above

$$Q_{j,i,m+1} = \max\{0, Q_{j,i,m} - 1\}, \text{ for } m = 1, \dots, M-1, \quad (2.4)$$

$$Q_{j,i,1} = C_{ji}, \quad \text{for all } j, i \neq j, \quad (2.5)$$

$$Q_{j,j,1} = 0, \quad \text{for all } j. \quad (2.6)$$

From eqn. (2.4-2.6), we can see the proximity factor tensor Q is a three dimensional matrix. The front square of the cuboid is simply the matrix C and the diagonal terms are overwritten to zero [2]. The reason is that if $j=i$ and $k=l$, there should be no penalty incurred (there is effectively only a single call and no interference is possible) [2]. $Interf(U)$ is the total interference in the network and (2.2) is the objective function to be minimized. Equation (2.3) indicates that the number of channels assigned to cell j must equal to D_j . The demand vector D is the constraint, which can be transformed to penalty function as shown below:

$$p(U) = \sum_{j=1}^N \left[\left(\sum_{k=1}^M U_{j,k} \right) - D_j \right]^2. \quad (2.7)$$

We can find that if the number of channels assigned to each cell is equal to D_j , the value of the penalty function is zero. The decrease in the value of (2.7) represents an approach to the feasible regions.

2.3 The Benchmarking CAP2s

The data set used in this thesis can be divided into three classes.

1. The first class consists of two small example networks. One is EX1 introduced in this chapter. The other is a larger one, called EX2 [2]. For EX2, $N = 5$, $M = 17$ and $D^T = (2, 2, 2, 4, 3)$. The compatibility matrix is shown below:

$$C = \begin{pmatrix} 5 & 4 & 0 & 0 & 1 \\ 4 & 5 & 0 & 1 & 0 \\ 0 & 0 & 5 & 2 & 1 \\ 0 & 1 & 2 & 5 & 2 \\ 1 & 0 & 1 & 2 & 5 \end{pmatrix}$$

2. The second class of data sets is based on the 21-cell regular hexagonal network used

by Sivarajan *et al.* [7], as shown in Figure 2.4. Two sets of demands used for the 21-cell problems are as follow:

$$D_1^T = (2, 6, 2, 2, 2, 4, 4, 13, 19, 7, 4, 4, 7, 4, 9, 14, 7, 2, 2, 4, 2);$$

$$D_2^T = (1, 1, 1, 2, 3, 6, 7, 6, 10, 10, 11, 5, 7, 6, 4, 4, 7, 5, 5, 5, 6).$$

Table 2.1 shows the details of these hexagonal test problems (from HEX1 to HEX4). The number of available channel M is the lower bound obtained by applying the lemma of Gamst [1] to those problems. Since ACC is not considered in HEX1 and HEX3, two different matrices are used for HEX problems. HEX1 and HEX3 are concern with only the co-channel (CCC) and the co-site constraint (CSC), they have the same matrix $C^{(1)}$. HEX2 and HEX4 include not only the co-channel (CCC) and the co-site constraint (CSC), but also the adjacent channel constraint (ACC), and they have the same matrix $C^{(2)}$.

Figure 2.4 Layout of a 21-cell hexagonal network

Table 2.1 Descriptions for hexagonal problems

Problem	N	M	D	CCC	ACC	C_{ii}	C
HEX1	21	37	D_1	Yes.	No	2	$C^{(1)}$
HEX2	21	91	D_1	Yes	Yes	3	$C^{(2)}$
HEX3	21	21	D_2	Yes	No	2	$C^{(1)}$
HEX4	21	56	D_2	Yes	Yes	3	$C^{(2)}$

The two compatibility matrices $C^{(1)}$ and $C^{(2)}$ used for the four HEX problems are calculated by considering the first two rings of cells around a particular cell as the sources of interference (See Figure 2.4) [2]. According to Gamst and Rave’s definition [4], the two matrices are both 21×21 and symmetric. Each diagonal term C_{ii} represents the minimum distance by which any two frequencies assigned to cell i must be separated corresponding to CSC. The CCC is represented by off-diagonal term $C_{ij} = 1$, and ACC is represented by $C_{ij} = 2$ [2]. The meaning of $C_{ij} = 0$ is that cells i and j are able to use the same frequency simultaneously without interference. From table 2.1, we can see that $C^{(1)}$ used by HEX1 and HEX3 includes CCC and CSC, while ACC is not considered. Hence, the off-diagonal terms in $C^{(1)}$ are 1 and 0, which correspond to CCC and no interference respectively. For instance, in Figure 2.4, cell 2 and cell 3 are in the first two rings of cells around cell 1, so there is co-channel interference between cell 1 and cell 2 as well as cell 1 and cell 3. Thus $C_{12}^{(1)} = C_{21}^{(1)} = 1$ and $C_{13}^{(1)} = C_{31}^{(1)} = 1$. Cell 3 and Cell 5 are not among the first two rings of cells around cell 1, so there is not co-channel interference between cell 1 and cell 4 or between cell 1 and cell 5. Hence $C_{14}^{(1)} = C_{41}^{(1)} = 0$ and $C_{15}^{(1)} = C_{51}^{(1)} = 0$. HEX2 and HEX4 use $C^{(2)}$, which considers CCC, CSC and ACC. The diagonal elements $C_{ii}^{(2)}$ in $C^{(2)}$ are all 3 as shown in table 2.1, and the off-diagonal terms of $C_{ij}^{(2)}$ are 1 or 2, which corresponds to CCC and ACC respectively. The two compatibility matrices are produced and shown as follows:

[illegible]

$$C^{(2)} = \begin{pmatrix} 3 & 2 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 2 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 2 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 2 & 3 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 2 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 2 & 3 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 0 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 3 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 2 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 1 & 2 & 3 \end{pmatrix}$$

3. The third class of data set is generated from the topographical data of an actual 24×21 km area around Helsinki, Finland [35]. Kunz calculated the traffic demand and interference relationships between the 25 regions around the base stations of this area. The compatibility matrix $C^{(3)}$ is obtained from Kunz data [2] [8], which we reproduce in as follows. The demand vector is:

$$D_3^T = (10, 11, 9, 5, 9, 4, 5, 7, 4, 8, 8, 9, 10, 7, 7, 6, 4, 5, 5, 7, 6, 4, 5, 7, 5).$$

This benchmarking data set is divided into four classes by considering only the first 10 regions (KUNZ1), 15 regions (KUNZ2), 20 regions (KUNZ3), and the entire area (KUNZ4) as shown in table 2.2.

2.4 Summary

In this chapter, we review the channel assignment problems briefly. Three classes of benchmarking CAP2s are presented. In Chapter 5, we will use stochastic ranking combined with genetic algorithm to solve these benchmarking problems and prove that our approach can improve the results and balance the dominance of objective and penalty functions.

Chapter 3

Review of Stochastic Ranking

3.1 Introduction

In constrained optimization problems, penalty functions are often used. However, it is difficult to strike the right balance between objective and penalty function. Runarsson and Yao [16] proposed a novel technique, stochastic ranking which can balance the objective and penalty functions conveniently.

The general nonlinear programming problem can be formulated as: Find x to

$$\text{minimize } y(x), \quad x = (x_1, \dots, x_n) \in R^n \quad (3.1)$$

In (3.1), $y(x)$ is the objective function. x is an n -dimensional vector and $x \in S \cap F$, where S defines the search space and F defines the feasible region. S is an

n -dimensional space whose boundary is:

$$x_{\min} \leq x_i \leq x_{\max}, \quad i \in \{1, \dots, n\}, \quad (3.2)$$

where $x_{\min}$ is the lower bound of x_i and $x_{\max}$ is the upper bound of x_i . The feasible region F is given by:

$$F = \{x \in R^n \mid c_j(x) \leq 0 \quad \forall j \in \{1, \dots, m\}\} \quad (3.3)$$

where $c_j(x)$, $j \in \{1, \dots, m\}$ are constraints.

One way often used to deal with constrained optimization problems is to introduce a penalty term into the objective function to penalize constraint violations [17] thereby transforming a constrained optimization problem into an unconstrained one:

$$f(x) = y(x) + \omega_g p[c_j(x)]; \quad j = 1, \dots, m \quad (3.4)$$

In this equation, $p \geq 0$ is a real-valued function and represents a “penalty”. The “strength” of the penalty is determined by penalty coefficient ω_g , where g is the generation counter. The introduction of the term, i.e., eqn (3.4), has been used widely in evolutionary optimization [18], [19] where penalty coefficient ω_g may vary from generation to generation. The penalty function may take various forms. One often used is the following quadratic loss function (3.5) [20], [21]. The decrease in the value of (3.5) represents an approach to the feasible regions.

$$p(c_j(x); \quad j = 1, \dots, m) = \sum_{j=1}^m \max\{0, c_j^2(x)\}. \quad (3.5)$$

What we should notice is that this constraint-handling technique is also applicable to

any other forms of penalty functions.

We will use this function to solve channel assignment problems. The cost function given by eqn (2.2) is the objective function to be minimized and the demand vector D is the constraint, which can be transformed to penalty function as shown below:

$$p(x) = \sum_{j=1}^N \left[\left(\sum_{k=1}^M X_{jk} \right) - D_j \right]^2 \quad (3.6)$$

Although for many problems, this penalty function method may work well, another difficult problem, i.e., how to determine the penalty coefficients, arises. If ω_g is chosen to be too small, the constraints may not be satisfied in the end. If ω_g is too large, the quality of solution could be very poor. Even for the same problem, different phases of search may require different ω_g value. For some problem whose feasible regions in the whole search space are disjoint, it is very inefficient to use a large ω_g . Since a large ω_g hinders the search in the infeasible regions and it may be difficult for an evolutionary algorithm to move from one feasible region to another unless they are very close to each other [16]. Hence, it is necessary to set an appropriate ω_g value in order that two or more different feasible regions can be connected by exploration of infeasible regions. How large ω_g should be? The answer to this question is problem dependent.

Some work has used the idea of dynamic setting of ω_g values in evolutionary constrained optimization [18], [20], [21]. A predefined monotonically nondecreasing sequence of ω_g values is used in such work. Although this idea worked for some simple problems, unfortunately, it failed when met more difficult ones because the choice of ω_g values is problem dependent [22]. In some work [23], [24], the concept of multiobjective optimization is used in order to avoid setting ω_g , i.e., constraints are considered as additional objective functions. However, this method does not work very well, because it is not easy to balance different objectives in optimization.

According to eqn. (3.4), different ω_g values define different fitness functions and an individual that is fit under one fitness function may not be fit under a different fitness function [16]. Finding a near optimal ω_g adaptively is equivalent to ranking individuals in a population adaptively [16]. Hence, for evolutionary constrained optimization problems, the issue becomes how to rank individuals according to their objective and penalty values [16]. Stochastic ranking is such a constraint-handling technique that ranks individuals in one population without specifying a ω_g value.

3.2 Constraint Handling by Stochastic Ranking

3.2.1 Penalty Method

We suppose that there are n individuals in one population. For a given penalty coefficient $\omega_g > 0$, we rank the n individuals according to eqn. (3.4)

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \quad (3.7)$$

Let us check the adjacent pair i and $i+1$ in the ranked order (3.7)

$$y(x_i) + \omega_g p(x_i) \leq y(x_{i+1}) + \omega_g p(x_{i+1}), \quad i \in \{1, \dots, n-1\} \quad (3.8)$$

A parameter $\tilde{\omega}_i$ is denoted as the critical penalty coefficient in [16] for the adjacent pair i and $i+1$.

$$\tilde{\omega}_i = (y(x_{i+1}) - y(x_i)) / (p(x_i) - p(x_{i+1})), \quad \text{for } p(x_i) \neq p(x_{i+1}) \quad (3.9)$$

For convenience, we let $y_i = y(x_i)$ and $p_i = p(c_j(x_i))$, $j = 1, \dots, m$

For the given $\omega_g \geq 0$, there are three different cases [16]:

- 1) $y_i \leq y_{i+1}$ and $p_i \geq p_{i+1}$: This comparison is dominated by the objective function because the objective function y plays the main role in determining the inequality. In this case, $0 < \omega_g < \tilde{\omega}_i$. When both individuals are feasible, $p_i = p_{i+1} = 0$ and $\tilde{\omega}_i \rightarrow \infty$.
- 2) $y_i \geq y_{i+1}$ and $p_i < p_{i+1}$: This comparison is dominated by the penalty function because the penalty function p plays the main role in determining the inequality. In this case, $0 < \tilde{\omega}_i < \omega_g$.
- 3) $y_i < y_{i+1}$ and $p_i < p_{i+1}$: This comparison is *nondominated* because neither objective function nor penalty function can determine the inequality by itself. In this case, $\tilde{\omega}_i < 0$.

When comparing nondominated or feasible individuals, the value of ω_g has no effect on the inequality (3.8), which means that ω_g does not change the order of ranking of the two individuals. However, the ω_g value is very important in the first two cases where $\tilde{\omega}_i$ is the flipping point that will decide whether the comparison is objective or penalty function dominated. For example, if we decrease ω_g to a value less than $\tilde{\omega}_i$ in the second case, individual i will change from a less fit one into a fitter one. For the whole population, the fraction of individuals dominated by the objective and penalty functions will be determined by the ω_g value used for comparisons [16].

Although ω_g value is very important in ranking, not all possible ω_g values can affect the ranking of individuals. Only those within a certain range, i.e., $\omega_{g \min} < \omega_g < \omega_{g \max}$ can influence the ranking. The lower bound $\omega_{g \min}$ is the minimum critical penalty coefficient and the upper bound $\omega_{g \max}$ is the maximum critical penalty coefficient. The lower bound $\omega_{g \min}$ is calculated from adjacent individuals ranked only according to the objective function and the upper bound $\omega_{g \max}$ is computed from adjacent individuals ranked only according the penalty function. Generally, the ω_g values can be categorized into three categories [16]:

- $\omega_g < \omega_{g \min}$: In this case, ω_g is too small to influence the ranking of individuals

so that all comparisons are determined by the objective function only. This case can be called as under penalization.

- $\omega_g > \omega_{g \max}$: In this case, all comparisons are determined only by the penalty function. The influence of objective function can be ignored because the ω_g value is too large. This case is called as over penalization.
- $\omega_{g \min} < \omega_g < \omega_{g \max}$: In this case, all comparisons are based on both objective and penalty function.

All penalty methods can be categorized into one of the above three categories and some methods may fall into different categories during different stages in search [16]. It is important to understand the difference among these three classes because they indicate which function (both functions) is driving the search process and how search progresses [16]. For example, many dynamic methods use a low ω_g value at the beginning and a good region can be found which may contain both feasible and infeasible individuals. To the end of the search, a large ω_g is used to find a good feasible individual. Such method is efficient if the unconstrained global optimum of the problem is close to its constrained one. However, when deals with problems whose constrained global optimum is far away from its unconstrained one, the initial low ω_g value would drive the search toward the unconstrained one and thus further away from the constrained global optimum. Hence this method is unlikely to work well for such problems.

The traditional constraint-handling technique used in evolution strategies belongs to the category of overpenalization because all infeasible individuals are regarded worse than feasible ones [25], [26], [27], [86]. Neither under- nor over- penalization is a good constraint-handling technique, and there should be a balance between preserving feasible individuals and rejecting infeasible ones [28]. That means ranking should be dominated by a combination of objective and penalty functions, and thus the penalty

coefficient ω_g should be within the bounds $\omega_{g\min} < \omega_g < \omega_{g\max}$ [16]. It should be noticed that the upper bound $\omega_{g\max}$ and lower bound $\omega_{g\min}$ are problem dependent, and may change from generation to generation.

We can measure the balance of dominance of objective and penalty functions by counting how many comparisons of adjacent pairs are dominated by the objective and penalty function, respectively [16]. This number of comparisons can be calculated for any given ω_g by counting the number of critical penalty coefficients given by eqn. (3.9), which are greater than ω_g [16]. From the analysis above, we can see what a penalty method tries to do is to obtain the right balance between objective and penalty function in order to make the search move to the optimum in the feasible space.

3.2.2 Stochastic Ranking

Because how to find an optimal or near optimal ω_g is a difficult problem, another approach, stochastic ranking was proposed [16] to balance the dominance of the objective and penalty functions. In [16], a probability P_f was introduced. It is the probability of using only the objective function for comparisons in ranking in the infeasible regions of the search space [16]. That is, given any pair of two adjacent individuals, if both individuals are feasible, the probability of comparing them (in order to determine which one is fitter) according to the objective function is 1; otherwise, the probability of comparing them according to the objective function is P_f . This method looks like the use of a probability by Surry and Radcliffe [23] in deciding the outcome of competitions between two individuals in tournament selection. The stochastic ranking is, however, quite different because rank-based selection is used and there is not any extra computational cost for self-adapting P_f [16]. More importantly, the motivation of stochastic ranking comes from the need for balancing objective and penalty functions directly and explicitly in optimization [16]. Surry and Radcliffe's

method [23] does not try to balance the dominance of penalty and objective functions in a population. Stochastic ranking is achieved by a bubble-sort-like procedure as shown below [16]:

```

1  Initialize  $n$  individuals  $x_1, \dots, x_n$ 
2  for  $i = 1$  to  $N$ 
3  { for  $j = 1$  to  $n - 1$  do
4    {  $u = \text{rand}(0, 1)$ 
5      if ( $p(x_j) = p(x_{j+1}) = 0$ ) or ( $u < P_f$ )
6        { if ( $y(x_j) > y(x_{j+1})$ )
7          Exchange the position of  $x_j$  and  $x_{j+1}$  in the ranking list
8        }
9      else if ( $p(x_j) > p(x_{j+1})$ )
10     {
11       Exchange the position of  $x_j$  and  $x_{j+1}$  in the ranking list
12     }
13   else
14     no exchange
15 }
16 }
```

$\text{rand}(0, 1)$ is a uniform random number generator. When $P_f = 0$, the ranking is an over penalization, and for $P_f = 1$, the ranking is an under penalization. N is the number of sweeps going through the whole population. The initial ranking is always generated at random.

Given that at least one individual is infeasible, in the ranking procedure, the probability of an adjacent individual winning a comparison, i.e., holding the higher rank, is [16]

$$P_w = P_{fw}P_f + P_{\phi w}(1 - P_f) \quad (3.10)$$

where P_{fw} is the probability of the individual winning according to the objective

function, and P_{ϕ_w} is the probability of the individual winning according to the penalty function. If both adjacent individuals are feasible, $P_w = P_{fw}$. Let us calculate what is the probability of winning k more comparisons than losses. Then the total number of wins must be $k' = (N + k)/2$, where N is the total number of comparisons made. The probability of winning k' comparisons out of N is give by the binomial distribution [16]

$$P_w(y = k') = \binom{N}{k'} P_w^{k'} (1 - P_w)^{N-k'} \quad (3.11)$$

The probability of winning at least k' comparisons is [16]

$$P_w'(y \geq k') = 1 - \sum_{j=0}^{k'-1} \binom{N}{j} P_w^j (1 - P_w)^{N-j} \quad (3.12)$$

From eqn. (3.11) and (3.12), we can see that if the number of comparisons N is large enough, the initial ranking will have little influence [16]. Also, from the procedure of stochastic ranking described above, we can find that if N is sufficiently large, the ranking is decided by P_f , not by the initial ranking. It is worth noting that the probability P_w is usually different for different individuals in different stages of ranking. We can also find if $P_f = 1/2$, then $P_w = 1/2$. That means the chance for a comparison to be made based on the objective or penalty function will be equal. Because we only need feasible individuals to be the final solutions, P_f should be chosen less than $1/2$ in order that there is a bias against infeasible solutions. It is easy to adjust the strength of the bias by adjusting P_f only [16]. When the number of sweeps N approaches ∞ , the ranking will be determined by the bias P_f , which means, if $P_f > 1/2$, the ranking is dominated by the objective function, and when $P_f < 1/2$, the ranking is the over penalty ranking [16].

In [16], stochastic ranking is achieved by the evolution strategy and some benchmark continuous optimization problems are solved. The fact that stochastic

ranking can balance the objective and penalty function efficiently and the combination of stochastic ranking and evolution strategy can achieved better results compared with many other traditional algorithms enlightens us to apply the new constraint-handling technique to combinatorial optimization problems. Since combinatorial optimization problems are “discrete”, we use genetic algorithm instead of evolution strategy to achieve stochastic ranking.

3.3 Conclusions

To solve the problem of how to set the hard-setting penalty coefficient r_g (which is problem dependent), a novel approach stochastic ranking was proposed by Runarsson and Yao [16]. The dominance of objective and penalty function can be adjusted by a probability P_f . It does not require a prior knowledge about a problem because it does not use any penalty coefficient in a penalty function [16]. In [16], some benchmark continuous optimization problems are solved to test the efficiency of stochastic ranking. The good results obtained show that stochastic ranking is very convenient to handle constraints. To cope with combinatorial optimization problems, we choose genetic algorithm to achieve stochastic ranking. One reason for choosing GA is that it does not introduce any specialized constraint-handling variation operators. It is worth noting that the constraint-handling technique based on the stochastic ranking scheme can be used in any evolutionary algorithm. The features of this method are summarized as follows:

- ✓ Stochastic ranking can balance the dominance of objective and penalty function conveniently.
- ✓ The dominance can be adjusted easily by adjusting P_f only.
- ✓ Rank-based selection is used.
- ✓ It does not require a priori knowledge about a problem.

- ✓ The technique does not introduce any specialized variation operators.

Based on this new constraint-handling technique, we combine it with genetic algorithm to solve difficult combinatorial optimization problems, i.e., the channel assignment problems.

Chapter 4

Basics of Genetic Algorithms (GAs)

4.1 Introduction

Genetic algorithms (GAs) have received a great deal of attention because of their ability to search for optimal solutions. They are based on the mechanics of natural selection and genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [29]. The genetic algorithm was firstly developed by John H. Holland in the 1960's [30]. It allows computers to solve difficult problems [29], [31]. The basic operation of genetic algorithm is simple as shown in fig. 4.1. Firstly, a certain number of possible solutions are initialized randomly. Next, calculate the fitness value for each individual. The better solutions are recombined with each other to form some new solutions. Finally the new solutions are used to replace the poorer of the original solutions and the process is repeated.

Figure 4.1 A canonical genetic algorithm

4.2 Basics of Genetic Algorithm

4.2.1 Biological Background

Genetic algorithms are inspired by Darwin’s theory about evolution. All living organisms consist of cells and in each cell, there is the same set of chromosomes, which are made of genes and serve as a model of the whole organism. A gene consists of blocks of DNA and each gene encodes a particular protein. Basically, each gene encodes a trait such as the color of skin, eyes, etc. Alleles are possible settings for a trait. Each gene has its own position in the chromosome called locus. Table 4.1 shows the comparison of natural and GA terminology [30].

Natural	Genetic Algorithm
Chromosome	string
Gene	feature, character, or detector
Allele	feature value
Locus	string position
Genotype	structure
Phenotype	parameter set, alternative solution, A decoded structure
Epistasis	nonlinearity

Table 4.1 the comparison of natural and GA terminology

During the process of reproduction, crossover occurs firstly. Genes from parents generate a whole new chromosome. After crossover, mutation takes place. The elements in the DNA are changed.

4.2.2 Principle of Genetic Algorithm

The vocabulary used by genetic algorithms is nearly the same as those used by natural genetics. Each chromosome represents a possible solution to a problem. An evolution process runs on a population of chromosomes corresponding to a search through a space of potential solutions [31] and such process has two objectives: exploiting the best solutions and exploring the search space. GAs have many variants. Their basic mechanisms are similar as illustrated in the following [29]:

- **[Initialization]** Generate the first generation of n chromosomes randomly (suitable solutions for the problem). n is called the population size.
- **[Calculating the Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population;
- **[New population]** Create a new population by repeating the following steps

until the new population is complete;

- **[Selection]** Select two chromosomes to be parents from the current population according to their fitness (the best one has the biggest chance to be selected);
 - **[Crossover]** Cross over the parents to form new offspring (child) with a crossover probability. If no crossover was performed, the offspring is an exact copy of parents;
 - **[Mutation]** With a mutation probability mutate new offspring at certain locus (position in chromosome);
 - **[Accepting]** The offspring is accepted as an individual in a new population.
- **[Replace]** New population is used for a further run of the algorithm;
 - **[Test]** If the end condition is satisfied, return the best solution in the current population and the algorithm is stopped;
 - **[Loop]** Go to **[Calculating the Fitness]**.

4.2.3 Selection

Chromosomes are selected from the population to be parents to crossover. The issue of selecting the chromosomes is important in Genetic Algorithm. In Darwin's evolution theory the fittest ones should survive and create new offspring. However, If only the fittest individuals are chosen every time, a convergence to local minima is inevitable. Basically, selection can be performed by 3 steps. Firstly, the good solutions (usually above-average) in a population are identified. Then, make multiple copies of good solution. Finally, the bad solutions are get rid of so that the multiple copies of good solutions can be placed in the population.

There exit many ways to select the parents. One of the simplest methods is based

on the *Tournament Selection* [29] used in our model. In the tournament selection, tournaments are played between two solutions and the better solution is chosen and placed in the mating pool [29]. Two other solutions are picked again and the better one is placed into another slot in the mating pool. This process goes on until the mating pool is full. If the tournaments are carried systematically, each solution can participate in exactly two tournaments. The best solution in a population will win both times, thereby making two copies of it in the new population. On the contrary, the worst solution will lose both tournaments and will be eliminated from the population. Hence, any solution in a population will have zero, one or two copies in the new population. It has been shown in [29] that the Tournament Selection has better or equivalent convergence and computational time complexity properties when compared to any other reproduction operator that exists in the literature. An interesting aspect of the tournament selection operator is that just by changing the comparison operator, the minimization and maximization problems can be handled easily [29].

4.2.4 Operators of GA

From the genetic described in 4.2.2, we can see that the most important parts of the genetic algorithm are crossover and mutation and these two operators influence the optimization performance to a large extent.

The information about solution is stored in the chromosome. A binary string is the most used way of encoding. Thus, the chromosome could look like the one shown in table 4.2:

Table 4.2: Sample of chromosomes

Chromosome 1	111000110010
Chromosome 2	000111001101

Each chromosome is a string bit of 0 or 1 and each bit in this string represents certain characteristic of the solution. It is worth noting that there exist many other ways of encoding and it is problem-dependent.

Crossover

Crossover is the process of generating new offspring by using the genes selected from parent chromosomes. The simplest form of crossover is to choose some crossover points randomly. Before this point, every bit of the first parent is duplicated and after the crossover point every bit of the second parent is duplicated. The process of crossover is simply shown in Table 4.3.

Table 4.3 Sample of crossover

Chromosome 1	100100 1010100
Chromosome 2	001010 1101001
Offspring 1	1001001101001
Offspring 2	0010101010100

Crossover can be rather complicated. It is dependent on the way of encoding. Using specific crossover for a specific problem can enhance the performance of the genetic algorithm to some extent.

Mutation

After crossover, mutation will be performed. It is mainly to prevent all solutions in population from falling into a local optimum of the solved problem. Mutation can change the new offspring randomly. In the case of binary encoding, we can change

some “1” to “0” or “0” to “1” randomly. Table 4.4 shows a sample of mutation.

Table 4.4 Sample of mutation

Offspring 1	1001001101001
Offspring 2	0010101010100
Mutated offspring 1	1000001101001
Mutated offspring 2	0011101010100

The mutation also depends on the way of encoding.

4.2.5 Parameters of GA

There are three basic parameters of GA: crossover probability, mutation probability and population size.

Population size indicates how many individuals (chromosomes) are there in one generation. If there are too many chromosomes in one generation, the speed of GA is very slow. However, if there are too few chromosomes in one generation, only a small part of search space is explored. Research denotes that under some limits, which depend mainly on encoding and the problem itself, it is no use increasing population size, because it does not make problem solving faster.

Crossover probability indicates how often crossover will be performed. Crossover exchanges some parts of chromosome between parents to generate new offspring. Hence, without crossover, offspring is same as the exact copies of parents. If crossover probability is 100%, then no individual will remain in the next generation. If it is 0%, no new offspring will appear in the next generation (It does not mean that the new generation is the exactly same). The purpose of crossover is that new chromosomes will have good parts of old chromosomes and maybe the new chromosomes will be better

than the old ones. However, it is useful to leave some part of population survive to next generation.

Mutation probability indicates how often parts of chromosome will be mutated. Without mutation, offspring is generated only by crossover. Mutation changes part of chromosome. If mutation probability is 100%, all chromosomes are changed. If it is 0%, nothing is changed after crossover. Mutation is to prevent GA from falling into local extreme, but it should not be performed very often. Otherwise, GA will become to a random search.

4.3 Summary

In this chapter, we review the basics of genetic algorithms. Genetic algorithms are search algorithms based on the mechanics of natural selection and genetics. One reason for choosing GA to combine with stochastic ranking is that it does not introduce any specialized constraint-handling variation operators. We will show that the simple extension to the GA, i.e., the use of the stochastic ranking scheme shown in the next chapter, can achieve significantly better results than other more complicated techniques.

Chapter 5

Solving CAP2s with GA-SR

5.1 Introduction

Many methods have been proposed to handle constraints for combinatorial optimization problems. Most of them are based on the penalty functions, which penalize unfeasible solutions, i.e.

$$f(X) = \begin{cases} y(X) & \text{if } X \in F \\ f(X) + \text{penalty}(X) & \text{if } X \in S - F \end{cases} \quad (5.1)$$

where S defines the search space and $F \subseteq S$ defines the feasible regions. Some methods apply death penalty, which is the most severe penalty. In this case, infeasible individuals are rejected. Homaifar, Lai, and Qi (1994) proposed a method which uses static penalties [91]. It works by setting several levels of violation for each constraint. Penalty coefficients R_{ij} is created for each level of violation and for each constraint.

Higher levels of violation require larger values of this coefficient. The evaluation function used is:

$$f(X) = y(X) + \sum_{j=1}^m R_j c_j^2(X) \quad (5.2)$$

In (5.2), $c_j(X)$ measures the violation of the j -th constraint. In [92], Joines and Houck proposed a method which applies dynamic penalites. Individuals are evaluated (at the iteration i) by the following formula:

$$f(X) = y(X) + (C \times i)^\alpha \sum_{j=1}^m f_j^\beta(X) \quad (5.3)$$

where C , α and β are constants. A reasonable choice for these parameters is $C = 0.5$, $\alpha = \beta = 2$ [90].

In this chapter, we combine genetic algorithm with stochastic ranking to solve CAP2s. One reason for choosing GA is that it does not introduce any specialized constraint-handling variation operator (such as formula (5.2), (5.3)), although these operators may be quite useful for particular types of problems. (for example, see [90]).

Firstly we initialize n individuals randomly, $X_1, X_2, \dots, X_n$. The way we use to represent a solution was proposed by K. Smith [40]. X_i is a matrix which represents an individuals i within total population n of solutions. Each individual has an objective value $y(X_i)$ and a penalty value $p(X_i)$, $i = 1, 2, \dots, n$. Then, the bubble-sort-like ranking described in chapter 3 is used to rank these n individuals. m individuals which occupy the higher positions in the ranking list may have the chance to generate offspring, while others are abandoned. $m < n$ and m/n is called the truncation level. A new generation is created by applying evolutionary operators to these m individuals (selection, crossover and mutation). Each one of the m individuals creates n/m offspring on average so that a total of n offspring are generated. The individuals in the new generation are ranked

again and this procedure will go on until the stop criteria are satisfied. Figure 5.1 shows the process:

Figure 5.1 Flowchart for overall implementation

5.2 Implementation of Stochastic Ranking to CAP2s

In this section, we apply a new constraint balance technique proposed by Runarsson and Yao [16], i.e., stochastic ranking (SR), together with genetic algorithms (GA), to solve CAP2 and to balance the constraints. The length of a chromosome is $M \times N$ and each bit of the chromosome is given by eq. (2.1). Each chromosome can be regarded as an $M \times N$ matrix. In [40] and [86], for the first generation of chromosomes, the number of “1” assigned to row i (representing cell i) is equal to D_i (the call demand in cell i) so that each individual is feasible. During crossover, entire rows are swapped so that after crossover, the feasibility of the two individuals in the new generation is guaranteed. Mutation takes place by swapping a “1” and a “0” within the same row (cell). Thus, all the individuals generated are feasible and infeasible solutions are excluded. In our work, we initialize the first generation randomly and keep a certain number of infeasible solutions in each generation. Both the crossover site and the genes mutated are selected randomly. When the problem is very large, such as KUNZ4 and HEX4, most of the search space is infeasible. In order to save searching time, we define a ratio γ between the number of feasible solutions to the population size. If γ is less than a certain value, we use the same way of crossover and mutation as used in [40] and [86], in order to generate fewer infeasible solutions. If the ratio is larger than this value, we randomly choose the crossover site and the genes mutated so that there can be enough infeasible solutions. During simulations, several parameters, such as P_f , γ , the crossover probability, the mutation probability, and the population size need to be set. These values were set by trial and error. Table 5.1 shows the parameters for various CAP2s. If no better results have been found within 20000 generations or an interference-free assignment has been found, the search process is stopped. Table 5.2 lists the percentage of feasible individuals in the final populations.

5.3 The Simulation Results and Comparisons with Various Algorithms

The program was coded in C language using the Microsoft Visual C++ 6.0 Integrated Development Environment (IDE). The simulation was done on a SUN SPARC workstation running Sun OS 5.7.

Table 5.2 compares the interference values obtained by using GA-SR with many other algorithms. Those algorithms are genetic algorithms without stochastic ranking (GA) [86], GAMS/MINOS-5 (labeled GAMS) [2], steepest descent (SD) [2], stochastic simulated annealing (SSA) [2], the original Hopfield network (HN) (without hill-climbing) [2], the hill-climbing Hopfield network (HCHN) [2], the self-organizing neural network (SONN) [2], the multistage self-organizing algorithm combined with the transiently chaotic neural network (MSO-TCNN) [88], and stochastic chaotic simulated annealing (SCSA) [87]. The values in bold indicate improved solutions over all other algorithms. The minimum values and average values obtained by other algorithms listed in that table are calculated from 10 runs' results. In order to compare with them, we also run the program 10 times and calculate the average values. We note that standard deviations are not available in most methods shown in Table 5.3; however, we list the standard deviations for our method for future references and comparisons. "Min", "Ave" and "Sd" are the minimum, average, and standard deviation of the total interference in the mobile network, respectively. Table 5.4 gives the experimental results on 10 benchmark problems using GA-SR; 30 independent runs were carried out.

Table 5.1. Parameters used in the simulations. (P_s is the population size; P_c is the crossover probability; P_m is the mutation probability)

Problem	P_s	P_c	P_m	γ	P_f
EX1	40	0.75	0.3	--	0.45
EX2	60	0.85	0.2	--	0.46
HEX1	100	0.7	0.4	--	0.46
HEX2	120	0.65	0.35	--	0.465
HEX3	140	0.8	0.4	0.15	0.455
HEX4	140	0.85	0.35	0.15	0.46
KUNZ1	80	0.75	0.25	--	0.45
KUNZ2	80	0.7	0.2	--	0.445
KUNZ3	120	0.8	0.3	0.2	0.455
KUNZ4	140	0.7	0.35	0.15	0.465

Table 5.2 The percentage of feasible individuals in the final population. (Parameters used are listed inTable 5.1.)

Problem	EX1	EX2	HEX1	HEX2	HEX3
Percentage(%)	100	100	85	77.5	27.8
Problem	HEX4	KUNZ1	KUNZ2	KUNZ3	KUNZ4
Percentage(%)	15	100	40	25.8	15

Table 5.3 Comparison of the total interference obtained by different methods

	GA-SR		GA		GAMS	SD		MSO-TCNN	
Problem	Min	Ave±Sd.	Min	Ave	Min	Min	Ave	Min	Ave
EX1	0	0±0.0	0	0	2	0	0.6		
EX2	0	0±0.0	0	0	3	0	1.1		
HEX1	46	47.7±0.4	48	48.1	54	55	56.8		
HEX2	17	18.4±0.2	19	19.3	27	25	28.9		
HEX3	76	76.5±0.5	76	76.4	89	84	88.6		
HEX4	16	17.5±0.6	17	17.2	31	26	28.2		
KUNZ1	19	19.8±1.0	20	20.1	28	22	24.4	20	20.6
KUNZ2	29	29.4±0.3	29	29.4	39	26	28.1	30	31.2
KUNZ3	13	13.0±0.0	13	13.0	13	15	17.9	13	13.0
KUNZ4	0	0±0.0	0	0	7	3	5.5	0	0.0

	SSA		HN		HCNN		SONN		SCSA	
Problem	Min	Ave	Min	Ave	Min	Ave	Min	Ave	Min	Ave±Sd.
EX1	0	0.0	0	0.2	0	0.0	0	0.4	0	0.0±0.0
EX2	0	0.1	0	1.8	0	0.8	0	2.4	0	0.0±0.0
HEX1	49	50.7	48	49.0	48	48.7	52	53.0	47	47.7±0.3
HEX2	19	20.4	19	21.2	19	19.8	24	28.5	18	18.5±0.2
HEX3	79	82.9	79	81.6	78	80.3	84	87.2	76	77.3±0.3
HEX4	17	20.1	20	21.6	17	18.9	22	29.1	16	17.2±0.4
KUNZ1	21	21.6	21	22.1	20	21.1	21	22.0	19	20.0±2.1
KUNZ2	32	33.2	32	32.8	30	31.5	33	33.4	30	30.3±0.2
KUNZ3	13	13.9	13	13.2	13	13.0	14	14.4	13	13.0±0.0
KUNZ4	1	1.8	0	0.4	0	0.1	1	2.2	0	0.0±0.0

Table 5.4 Experimental results on 10 benchmark problems; 30 independent runs were carried out.

Data Sets	P _s		P _c		P _m		P _f		Interference		
Problem	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Ave.
EX1	40	80	0.6	0.8	0.2	0.4	0.4	0.6	0	0	0
EX2	40	80	0.65	0.85	0.1	0.3	0.4	0.7	0	0	0
HEX1	80	120	0.6	0.8	0.2	0.4	0.36	0.65	46	48	47.73
HEX2	90	130	0.6	0.8	0.2	0.4	0.4	0.7	17	20	19.03
HEX3	100	140	0.65	0.85	0.3	0.5	0.3	0.6	76	80	77.3
HEX4	100	140	0.65	0.85	0.2	0.4	0.4	0.7	16	18	17.67
KUNZ1	60	100	0.6	0.8	0.15	0.35	0.45	0.75	19	22	20.96
KUNZ2	70	110	0.65	0.85	0.1	0.3	0.3	0.6	29	32	29.97
KUNZ3	90	130	0.65	0.85	0.2	0.4	0.4	0.7	13	15	13.4
KUNZ4	100	140	0.6	0.8	0.21	0.41	0.4	0.7	0	1	0.67

Table 5.5 The minimum interference values and the time used.

Problem		HEX1	HEX2	HEX3	HEX4	KUNZ1	KUNZ2	KUNZ3	KUNZ4
GA	Min	48	19	76	17	20	29	13	0
	Time (s)	30973	155714	7943	70932	6706	39160	18729	219988
GA-SR	Min	46	17	76	16	19	29	13	0
	Time (s)	129381	606039	102416	309372	59531	167027	196421	1209731

Table 5.5 gives the time needed to achieve the minimum interference using GA-SR. In comparison with GA, we have to acknowlege that GA-SR needs much more time to search the minimum values. That is because GA-SR searches not only the feasible regions but also the infeasible regions. However, in the cases where channels need to

be assigned only once and remain fixed, GA-SR is a powerful tool.

Figure 5.2 and 5.3 shows P_f versus interference with constant population size, P_c and P_m . The values of population size, P_c and P_m used are listed in table 5.1 and the time limits are those given in table 5.5. We can find that when P_f is set between 0.4 and 0.5, we have a high probability of finding good results. If P_f is larger than certain value, feasible solutions are difficult to be found or the quality is not good.

Figure 5.2 P_f versus interference with constant population size, P_c and P_m (HEX)

Figure 5.3 P_f versus interference with constant population size, P_c and P_m (KUNZ)

Table 5.6 gives the experimental results obtained by using a fixed P_f . We can see that for EX1-2, HEX1, HEX4 and KUNZ1, 3, 4, minimum interference values are found. The actual channel assignment solutions obtained by GA-SR with minimum interference for different CAP2s are listed in Table 5.7-Table 5.16

Table 5.6 Experimental results on 10 benchmark problems with fixed P_f ($P_f=0.46$); other parameters are those in Table 5.1.

Problem	EX1	EX2	HEX1	HEX2	HEX3
Min. Interference	0	0	46	18	77
Problem	HEX4	KUNZ1	KUNZ2	KUNZ3	KUNZ4
Min. Interference	16	19	30	13	0

Table 5.7 Channel assignment for EX1 problem with interference 0

Base Station No.	#Channels	Assigned channels
1	1	10
2	1	2
3	1	3
4	3	1, 6, 11

Generation number for which minimum value is achieved = 10

Table 5.8 Channel assignment for EX2 problem with interference 0

Base Station No.	#Channels	Assigned channels
1	2	1, 17
2	2	5, 13
3	2	3, 7
4	4	1, 6, 12, 17
5	3	4, 8, 15

Generation number for which minimum value is achieved = 365.

Table 5.9 Channel assignment for HEX1 problem with interference 46

Base Station No.	#Channels	Assignment channels
1	2	13, 19
2	6	4, 11, 15,20,24,27
3	2	21, 23
4	2	8, 29
5	2	17, 20
6	4	3, 18, 20, 31
7	4	5, 19, 22, 25
8	13	4,6,8,10,12,14,20,24,26,28,31,33,38
9	19	1,2,4,6,8,10,12,14,16,18,20,22,25,27,29,31,33,35,37
10	7	7, 13, 21,30,32,36
11	4	9, 15, 19, 34
12	4	5, 17, 25, 33
13	7	11,21,27,29,31,35,37
14	4	2.14,17,19
15	9	3, 7, 9,11,19,24,28,34,36
16	14	1,3,5,8,10,12,16,20,23,25,27,30,32,35
17	7	2, 4,6,17,21,31,33
18	2	5, 15
19	2	7, 17
20	4	10, 19, 27, 35
21	2	13, 29

Generation number for which minimum value is achieved = 50892.

Table 5.10 Channel assignment for HEX2 problem with interference 17

Base Station No.	#Channels	Assignment channels
1	2	20, 83
2	6	11, 22, 30,35,43,74
3	2	37, 59
4	2	6, 16
5	2	26, 87
6	4	6,18,51,75
7	4	16,29,53,79
8	13	5,9,25,33,38,45,50,55,58,65,69,72,87
9	19	3,7,15,18,24,27,41,49,52,57,61,63,66,70,78,81,84,89,90
10	7	12,20,29,32,46,73,76
11	4	38,69,80,91
12	4	24,44,51,82
13	7	12,20,30,47,60,69,80
14	4	14,32,35,43
15	9	19, 23, 26,40,62,67,77,82,85
16	14	1,13,17,21,28,31,36,42,47,60,64,75,80,91
17	7	17,34,39,44,54,68,86
18	2	4, 14
19	2	58,79
20	4	10, 19, 27, 35
21	2	13, 29

Generation number for which minimum value is achieved = 41351.

Table 5.11 Channel assignment for HEX3 problem with interference 76

Base Station No.	#Channels	Assignment channels
1	1	19
2	1	8
3	1	2
4	2	4, 6
5	3	5, 16, 20
6	6	1, 5, 7,10,13,18
7	7	2, 6,9,11,15,17,21
8	6	4, 6, 8,12,14,16
9	10	1,3,5,7,10,12,14,16,18,20
10	10	1,3,7,9,11,,13,15,17,19,21
11	11	1,4,6,8,9,11,13,15,17,19,21
12	5	2, 10,12,14,18
13	7	3,8,10,12,14,18
14	6	3, 7, 9,15,18,20
15	4	2, 4, 11, 17
16	4	5,10,13,16
17	7	3, 5,7,10,14,18,20
18	5	8, 11,17,19,21
19	5	1, 13,17,19,21
20	5	2,6,9,15,18
21	6	2, 4, 8,12,14,20

Generation number for which minimum value is achieved = 63762.

Table 5.12 Channel assignment for HEX4 problem with interference 16

Base Station No.	#Channels	Assignment channels
1	1	53
2	1	4
3	1	48
4	2	24, 36
5	3	17, 34, 53
6	6	16, 21, 25,31,35,42
7	7	2, 6,9,13,28,40,44
8	6	17, 20, 24,32,47,51
9	10	1, 7, 10, 14,26,29,35,39,43,54
10	10	3, 6, 11, 16,21,26,33,41,46,52
11	11	5,9,13,19,22,27,30,38,42,50,55
12	5	2, 15,32,40,47
13	7	7,14,27,39,48,52,56
14	6	4, 11, 18,46,50,54
15	4	15,30,34,38
16	4	5,12,42,45
17	7	12,18,23,31,37,49,56
18	5	1, 8,25,28,44
19	5	22, 25,36,48,55
20	5	2, 9,27,40,53
21	6	4, 15, 20,34,47,51

Generation number for which minimum value is achieved = 56431.

Table 5.13 Channel assignment for KUNZ1 problem with interference 19

Base Station No.	#Channels	Assignment channels
1	10	5, 7,9,11,13,19,21,25,27,29
2	11	2,5,7,11,15,17,19,21,23,27,29
3	9	1, 3,6,9,16,20,25,28,30
4	5	7, 11,19,27,29
5	9	4,8,10,12,14,18,22,24,26
6	4	5,19,21,29
7	5	8, 10,12,22,24
8	7	4, 8, 10, 14,18,22,26
9	4	2,15,17,23
10	8	1,3,6,13,16,20,28,30

Generation number for which minimum value is achieved = 70468.

Table 5.14 Channel assignment for KUNZ2 problem with interference 29

Base Station No.	#Channels	Assignment channels
1	10	3,11,13,19,21,24,33,37,42,44
2	11	3,7,9,11,14,19,21,25,33,37,40
3	9	1, 15, 18,26,28,29,32,34,43
4	5	3, 11,19,37,42
5	9	2, 4,6,8,12,16,20,31,38
6	4	9,11,14,24
7	5	2, 12,16,20,31
8	7	4, 6,8,20,22,36,38
9	4	5,17,25,27
10	8	10, 23,26,29,35,39,41,43
11	8	7, 9,13,21,24,33,40,44
12	9	1,7,13,24,28,30,32,34,40
13	10	5,10,14,17,23,25,27,29,35,41
14	7	2, 4,8,20,22,36,38
15	7	1,15,18,28,30,32,34

Generation number for which minimum value is achieved =70153.

Table 5.15 Channel assignment for KUNZ3 problem with interference 13

Base Station No.	#Channels	Assignment channels
1	10	10,15,20,22,33,35,40,43,50,55
2	11	3,7,12,14,17,25,38,48,52,57,60
3	9	5,8,25,29,36,38,41,46,51
4	5	17, 20,55,57,60
5	9	6, 11, 13,19,21,31,34,42,47
6	4	35,39,43,52
7	5	4, 11,31,37,54
8	7	6,13,19,21,23,34,47
9	4	12,16,30,45
10	8	15, 18,29,44,49,51,53,58
11	8	1,3,7,10,14,33,35,52
12	9	1,7,14,24,26,28,39,50,59
13	10	2, 9, 16, 22,30,32,39,45,49,56
14	7	4, 9, 23, 27,32,37,54
15	7	8,24,26,36,41,46,59
16	6	12, 17, 20,28,57,60
17	4	3, 7, 13, 50
18	5	19, 21,42,51,58
19	5	6, 29,34,38,53
20	7	15,18,25,31,40,44,47

Generation number for which minimum value is achieved = 42019.

Table 5.16 Channel assignment for KUNZ4 problem with interference 0

Base Station No.	#Channels	Assignment channels
1	10	9,13,17,25,32,46,50,56,65,69
2	11	7,11,15,21,30,34,36,48,52,58,70
3	9	8, 29, 31,35,40,44,53,62,73
4	5	13, 25,32,46,52
5	9	2, 12, 28,37,41,47,51,59,66
6	4	1,50,58,63
7	5	18, 37,43,51,66
8	7	2,12,14,20,28,41,59
9	4	16,26,45,67
10	8	4, 10,23,49,55,61,68,72
11	8	17, 19,36,42,50,54,58,69
12	9	6, 19, 27,33,38,42,54,60,64
13	10	1, 3,5,16,24,26,45,63,67,71
14	7	14,18,20,22,39,43,57
15	7	8,29,33,40,62,64,73
16	6	11, 15, 30,35,46,65
17	4	54,59,61,69
18	5	12, 31,41,49,53
19	5	2,4,7,34,68
20	7	23,25,38,52,59,61,72
21	6	12, 28, 37,41,46,53
22	4	9,13,20,24
23	5	3, 16,43,47,55
24	7	27,33,38,48,53,62,66
25	5	10, 28,49,61,68

Generation number for which minimum value is achieved = 73641.

The results for EX1, EX2 and KUNZ4 in Table 5.3 show that interference-free assignments can be found for these benchmarks, as evidenced by a zero interference value. We can see that for HEX1 and HEX2 we obtained lower interference values than other methods including genetic algorithms without stochastic ranking [86]. In [86], the search space is limited to feasible regions. Hence, the approach in [86] is a special case of the GA-SR approach presented here, i.e., if we restrict our search to the feasible space.

5.4 Conclusions

In this chapter, we have considered the problem of assigning channels to calls in a cellular mobile communication network. While many researchers have studied the problem of finding the minimum number of channels required to obtain an interference-free assignment, it is more useful in practical situations, where the number of available channels is substantially fewer than the minimum number for interference-free assignments, to find the assignment which minimizes the severity of interference while meeting the demand, given the number of available channels. Stochastic ranking is an interesting approach to balance the dominance of objective and penalty functions. In this chapter, we have combined stochastic ranking proposed by Runarsson and Yao with GA and applied to the CAP2. The simulations done on the 10 benchmark problems showed that genetic algorithms with stochastic ranking can achieve desirable results, satisfying the demand constraint and with lower (or equally low) interference when compared to other heuristic approaches. We also showed the GA with stochastic ranking can obtain better results than using GA alone.

Chapter 6

Apply Genetic Algorithm on NTU-SingTel Project

6.1 Introduction

In Singapore the number of users in mobile communications continues to grow; however, the number of usable channels (frequencies), which are necessary for the communication between mobile users and the base stations of cellular radio networks, is very limited. Hence it is necessary to design a network carefully in order to ensure an acceptable signal-to-interference (S/I) ratio. The recent NTU-SingTel joint research project concerns assigning calls in SingTel mobile network to frequency channels in such a way that interference between calls is minimized while demands for channels are satisfied. Optimal channel assignment can enhance traffic capacity of a cellular mobile network and decrease interference between calls, thereby improving service quality and customer satisfaction. It is well known that looking for an optimal channel assignment belongs to a class of problems with extremely high complexity (NP-hard).

Namely, even the most powerful computer cannot find the optimal channel assignment of a moderate network size within a reasonable amount of time using exhaustive search. In this NTU-SingTel joint research project, we use some new techniques to obtain optimal channel assignment for SingTel mobile network, while the specifications and requirements from SingTel mobile network are satisfied. The outcome of this project will be tested by SingTel Company in a more realistic operating environment.

In this project, we use the same format of SCA problem as in [2] which is a generalized quadratic assignment problem, treating the noninterference constraints as soft constraints in the objective function and the demand satisfaction constraints as hard constraints. The advantage of this approach is that a solution, which minimizes the severity of any interferences, will always be found, which is particularly useful if the demands and constraints are such that no interference-free solution exists. We transform the original data set provided by SingTel, such as the interference table and traffic information, etc, into the compatibility matrices needed in the model [2]. One reason for choosing GA is that we find the compatibility matrices are not symmetric. From eqn. (2.2), it can be found that if the compatibility matrices are not symmetric, we cannot obtain symmetric weights, which are essential for Hopfield neural networks. Thus, the algorithm based on Hopfield neural networks cannot be used to solve this problem. GA has not such limitation and it has a high probability for leading to good solutions on channel assignment problems [37]-[42].

6.2 The Data Set of NTU-SingTel Project

In the NTU-SingTel joint research project, we assign channels in two layers, i.e., micro layer and macro layer. Macro cells are large cells each spanning several kilometers, which are often used to cover the rural areas. Micro cells are small cells, spanning only

100 to 200 meters. Macro cells can span the area covered by a grid of many micro cells to reduce the handover rate of moving mobile stations. Because SingTel mobile communication network use different frequency “sub band” for macro and micro layers, we need not consider the interference between macro and micro layer. Hence, we can consider the channel assignment problems in macro layer and micro layer separately. The problem descriptions are listed in table 6.1, where N is the number of cells and M is the number of available channels.

Table 6.1 Descriptions for SingTel data sets

Problem	N	M	C	D
Macro cell	18	51	C_{macro}	D_{macro}
Micro cell	24	14	C_{micro}	D_{micro}

Two demand vectors used for macro layer and micro layer are give below:

$$D_{macro}^T = (4, 4, 5, 4, 4, 5, 4, 4, 4, 4, 5, 6, 4, 4, 3, 5, 4, 3);$$
$$D_{micro}^T = (4, 2, 4, 3, 4, 3, 4, 3, 6, 3, 2, 2, 4, 3, 4, 2, 3, 3, 3, 3, 3, 3, 4).$$

SingTel provides us a map on cell physical location and an interference table instead of compatibility matrices. The interference table states the total coverage area of each cell and the area of serving cell potentially affected by CCC interference from a certain cell (km^2).

Figure 6.1 A simplified layout of macro layer.

Table 6.2 A simplified interference table of cell 1 in macro layer.

Cell (#)	2	3	4	5	6	7	8	9
Area (km ²)	A _{1,2}	A _{1,3}	A _{1,4}	A _{1,5}	A _{1,6}	A _{1,7}	A _{1,8}	A _{1,9}
Cell (#)	10	11	12	13	14	15	16	17
Area (km ²)	A _{1,10}	A _{1,11}	A _{1,12}	A _{1,13}	A _{1,14}	A _{1,15}	A _{1,16}	A _{1,17}

Since the location map of base stations and the interference tables are commercial confidential, we show the simplified versions of them in this thesis. In table 6.2, we suppose that cell 1 is the serving cell.

$A_{x,y}$ = The area of the serving cell x affected by co-channel interference from cell y .

It should be noted that $A_{x,y} \neq A_{y,x}$. The “affected areas” given in the interference table specify a way to assign channels. For example, if a single channel is assigned to two neighboring cells, the “affected area ” will be very high due to the co-channel interference; if two very apart clean channels are assigned to two neighboring cells, the “affected area” will be very small.

In order to transform the original data set into the form of compatibility matrices, following steps were used:

1. We set all the diagonal elements of compatibility matrices to be 2 corresponding to CSC.
2. We assign $C_{ij} = 1$ in compatibility matrix when cell i is affected by cell j .
3. From the map on cell physical location (Figure 6.1), we assign $C_{ij} = 1$ whenever two cells (called cell i and j) are adjacent to each other. For example, according to figure 6.1, $C_{3,4} = C_{4,3} = 1$

Two matrices, C_{macro} and C_{micro} are created by following the above three steps

$$C_{macro} = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 2 \end{bmatrix}$$

$$C_{micro} = \begin{pmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \end{pmatrix}$$

6.3 Implementation of GA on NTU-SingTel joint research project

Because the channel assignment problems of this project belong to the CAP2, we use the same formulation of SCA discussed in previous chapter. A set of binary variables is defined:

$$U_{jk} = \begin{cases} 1, & \text{if channel } k \text{ is assigned to cell } j; \\ 0, & \text{otherwise.} \end{cases} \quad (6.1)$$

where $j = 1, \dots, N$ and $k = 1, \dots, M$. Thus, the task becomes to minimize the total interference of all assignment in the mobile network while the demand for channels in

each cell is satisfied:

Minimize

$$Interf(U) = \sum_{j=1}^N \sum_{k=1}^M U_{jk} \sum_{i=1}^N \sum_{l=1}^M Q_{ji(|k-l|+1)} U_{il}, \quad (6.2)$$

Subject to

$$\sum_{k=1}^M U_{jk} = D_j, \quad \forall j = 1, \dots, N \quad (6.3)$$

where N is the number of cells and the number of available channels is given by M .

From the interference table, we found that in some cells, the area affected by the co-channel interference is less than 5% of the total coverage area, while in others, this percentage may be up to near 90%. In order to reflect the degree of effects between cells more accurately, we proposed to add a new ratio matrix R to equation 6.2. R is of dimension $N \times N$ and each element R_{ij} is given by:

$$R_{ij} = \frac{A_{i,j}}{A_i} \quad (6.4)$$

where A_i is the coverage of cell i (km^2). Thus, the new objective function becomes:

$$Interf(U) = \sum_{j=1}^N \sum_{k=1}^M U_{jk} \sum_{i=1}^N \sum_{l=1}^M Q_{ji(|k-l|+1)} R_{ij} U_{il}, \quad (6.5)$$

We use GA to minimize eqn. (6.5). Firstly, we initialize the first generation randomly and evaluate each individual for its cost value, i.e., eqn (6.5). Normally, the first generation does not contain the best solution. Hence, genetic algorithm operators, such as selection, crossover and mutation, are used to generate the next generation.

This process will go through a pre-defined number of generations. The simulation will end either when the optimal solution is obtained or the number of generations has been

exhausted.

6.3.1 Choices of GA Parameters

Through the simulations, we use different GA parameters to observe the convergence behaviors. We found that when the population size equals to 110, the performance of GA is the best. That means the micro layer and macro layer problems can reach the optimized cost more frequently than using other value. If the population size is too small, it is not likely to find a global minimum earlier and it may not get the optimal value. If the population size is too large, it will take much more time to process. Crossover probability (P_c) and mutation probability (P_m) also can affect the performance of GA. We have tried many sets of parameters. We find that for micro layer, when $P_c = 65\%$ and $P_m = 35\%$, we can get the most satisfactory results and for macro layer, the most satisfactory is obtained when $P_c = 60\%$ and $P_m = 40\%$. After a lot of simulations, we chose a set of parameters listed in Table 6.3

Table 6.3 the parameters of genetic algorithm

Problems	Macro cell	Micro cell
Total population size	110	110
Chromosome length	918	168
Crossover probability	0.6	0.65
Mutation probability	0.4	0.35
Maximum number of generations	1000	1000
Computational time	6564	1308

6.3.2 Simulation Results

By using GA, we have obtained a near interference-free assignment on macro cell layer. We used the crossover and mutation operators proposed by K. Smith [40]. Due to there

are only 14 channels available, the total interference (Equation 6.5) of micro layer is 1.929680. Table 6.4 gives two results obtained by our method and previous work [89]. Both work utilized the same GA. The differences were the way to build compatibility matrices and the cost function used. In our work, we adopted (6.5), instead of (6.2) used in [89]. Table 6.4 showed our method improved the results. The actual channel assignment solutions of macro and micro layers obtained by GA with minimum interference are listed in Table 6.5 and Table 6.6.

Table 6.4 The number of cells affected by interference.

	Macro Cell (18 cells)	Micro cell (24 cells)
Previous research	8 cells	23 cells
Our research	4 cells	14 cells

6.4 Conclusions

In this chapter, we consider the practical channel assignment. We use genetic algorithm to solve problems of different layers, macro layer and micro layer. Genetic algorithm has not the limitation of nonsymmetric compatibility matrix and GA can often generate good results on combinatorial optimization problems. We propose a technique to transform the practical data set of SingTel into compatibility matrices. Our research is very instructive to apply CAP on real mobile communication system. The outcomes of this project have been tested in a more realistic environment.

Table 6.5 Channel assignment for macro layer

Base Station No.	#Channels	Assignment channels
1	4	20,36,40,44
2	4	21,27,32,45
3	5	8, 39,41,43,51
4	4	10,19,37,40
5	4	3, 5, 33, 42
6	5	9, 20,26,35,50
7	4	17,28,36,49
8	4	12,19,24,37
9	4	1, 6, 15, 25
10	4	12,16,24,30
11	5	5, 13,28,34,47
12	6	3, 22, 33,42,46,48
13	4	4, 7, 23, 38
14	4	13,18,44,47
15	3	16, 46, 48
16	5	2, 11,14,29,31
17	4	9,26,35,50
18	3	7, 22, 34

Table 6.6 Channel assignment for micro layer

Base Station No.	#Channels	Assignment channels
1	4	2, 6,9,12
2	2	3, 5
3	4	3, 6,9,12
4	3	1, 4, 6
5	4	1, 5, 8, 14
6	3	3, 9, 12
7	4	1, 8, 11, 13
8	3	3, 5, 14
9	6	1, 3,5,8,11,13
10	3	2, 12, 14
11	2	1, 14
12	2	3, 5
13	4	1, 4, 7, 10
14	3	3, 5, 12
15	4	4, 6, 9, 11
16	2	4, 14
17	3	1, 4, 14
18	3	3, 5, 14
19	3	1, 8, 11
20	3	3, 5, 14
21	3	1, 3, 5
22	3	4, 7, 10
23	3	8, 11, 13
24	4	2, 7, 10, 13

Chapter 7

Conclusions and Recommendations

In this chapter, we draw the conclusions of the thesis and some recommendations for future work are provided.

7.1 Conclusions

To deal with constrained optimization problems, common approach is to introduce a penalty function (controlled by a number of penalty coefficients) into the objective function to transform a constrained optimization problem into an unconstrained one. However, how to determine the hard-setting penalty coefficients turns out to be a difficult problem itself. A novel constraint-handling technique, stochastic ranking was proposed by Runarsson and Yao [16]. Using this method, we need not set the penalty coefficients and the dominance of objective function and penalty function can be adjusted easily by adjusting a probability P_f . The fact that the new technique can

obtain good results when applied to deal with continuous optimization problems motivates us to apply it for combinatorial optimization problems, i.e., channel assignment problem that is an NP-hard problem. We combine genetic algorithm with stochastic ranking. One reason for choosing GA is that it does not introduce any specialized constraint-handling variation operators. What we should notice is that the constraint-handling technique based on the stochastic ranking scheme can be used in any evolutionary algorithm, not just GA. For example, in [16], evolutionary strategy (ES) is used to achieve stochastic ranking to solve continuous optimization problem.

The thesis mainly concentrated on solving two different combinatorial optimization problems: channel assignment problems (CAP2) and practical channel assignment (NTU-SingTel joint research project).

- **Static Channel Assignment Problem:** We have discussed the problem of assigning channels to calls in a cellular mobile communication network in previous chapter. The SCA problem was formulated as a generalized quadratic assignment problem with interference constraints in the cost function, and traffic demands were treated as hard constraints [2]. It will be possible to extent to other kind of technical restrictions. The simulation results obtained so far in three classes of benchmark data sets indicate that the GA-SR is very convenient to balance the dominance of the objective function and penalty function and is very effective in solving channel assignment problem in mobile network planning.
- **Solving Practical CAP2 (NTU-SingTel joint research project):** Genetic algorithm is a very interesting approach for combinatorial optimization problems, which mimics the process of natural selection and natural genetics. Unlike to other heuristic methods such as simulated annealing which only examines a point at one time, and is one-dimensional in their search, GA searches from one population of points in search space to another, and tend to focus increasingly on areas with

deeper minima. To solve the static channel assignment problems in cellular mobile networks of SingTel, the genetic algorithm is implemented. This algorithm does not require the compatibility to be symmetric. The mutation operator has shown its effect in preventing local minima. In the end, the actual assignment outcomes of GA are provided and they have been tested in the realistic operating environment by SingTel Company.

7.2 Recommendations for Future Research

In this thesis, stochastic ranking combined with genetic algorithm is used to solve combinatorial optimization problems. There are many unsolved problems in these topics, which need further investigation. Future research can be conducted on the following aspects:

1. Although the approach, stochastic ranking, provides a new idea of constraint handling, further studies, in particular, strict comparison with the excellent techniques already in use are indispensable for the evaluation of its true ability. We also note that the largest number of cells analyzed in this thesis is 25, it is necessary to show the effectiveness of GA-SR on even larger size of CAPs.
2. Our simulations on CAP2 show that although GA-SR performs better than other approaches and we need not set the penalty coefficients, one still need to choose the GA parameters (crossover probability, mutation probability and population size) carefully. Invalid or poor solutions will result from parameters selected arbitrarily. For example, if mutation rate is too small, it will be difficult to find the global minimum; if mutation rate is increased beyond some critical value, too much randomness will be introduced into the population, and therefore leading to instability in results. However, we are in

the process of developing a dynamic mechanism to effectively solving optimization problems without the need to select too cautiously. We may change the crossover and mutation rate by observing the individuals in each generation. We also try to find a way, which can set the probability P_f dynamically. For example, if we need the process of search dominated by the penalty function, P_f should become less. On the other hand, if we hope the process of search to be dominated by the objective function, P_f should be adjusted larger. That how large the P_f should be can be determined by the information of current population such as the ratio of the number of feasible solutions to infeasible solutions. It will be interesting work to dynamically set P_f . Implementation of GA-SR to solve other practical optimization problems, such as the dynamic channel assignment problem will be continuously studied. The present work is only focused on static channel assignment. It is imperative to dynamically alter the channel assignments as demand increase, if efficiency of the frequency spectrum is to continue to be a goal of the optimization process. In a dynamic channel assignment problem, the demand for channels becomes a function of time.

3. We apply simple genetic algorithm on the NTU-SingTel project. Sometimes, it needs large population size. We can use Hybrid GA instead of simple GA. Although the simple GA is shown to be very useful in finding feasible channel assignments for some scenarios of a cellular system, they present some difficulties in fast convergence for some highly combinatorial problems

Bibliography

- [1] A. Gamst, "Some lower bounds for a class of frequency assignment problems", *IEEE Trans. Veh. Tech.*, vol. VT-35, pp. 8-14, Feb., 1986.
- [2] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural network", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, Feb., 1997.
- [3] M. Duque-Anton, D. Kunz and B. Ruber, "channel assignment for cellular radio using simulated annealing", *IEEE Trans. Veh. Technol.*, vol. 42, no. 1, February, 1993.
- [4] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems", *Proc. GLOBECOM'82*, pp. 309-315, 1982.
- [5] D. Kunz, "Channel assignment for cellular radio using neural networks", *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 188-193.

-
- [6] D. Manuel, D. Kunz and R. Bernhard, "Channel assignment for cellular radio using simulated annealing", *IEEE Trans. Veh. Technol.*, vol. 42, no.1, pp. 14-20, 1993.
- [7] K. N. Sivarajan, R. J. McEliece and J. W. Ketchum, "Channel assignment in cellular radio", *Proc. 39th IEEE Veh. Technol. Soc. Conf.*, pp. 846-850, May, 1989.
- [8] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks", *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, November, 1992.
- [9] P. Chan, m. Palaniswami and D. Everitt, "Neural network-based dynamic channel assignment for cellular mobile communication systems", *IEEE Trans. Veh. Technol.*, vol. 43, no. 2, May, 1994.
- [10] J. Kim, S. H. Park, P. W. Dowdy and N. M. Nasrabadi, "Cellular radio channel assignment using a modified Hopfield network", *IEEE Trans. Veh. Technol.*, vol. 46, no. 4, November, 1997.
- [11] S. Kim and S.L. Kim, "A two-phase algorithm for frequency assignment in cellular mobile systems", *IEEE Trans. Veh. Tech.*, vol. 43, No. 3, pp. 542-548, 1994.
- [12] R. Mathar and J. Mattfeldt, "Channel assignment in cellular radio networks", *IEEE Trans. Veh. Tech.*, Vol. 27, No. 2, pp. 57-64, 1978.

-
- [13] F. Box, "A heuristic technique for assignment frequencies to mobile radio nets", *IEEE Trans. Veh. Tech.*, Vol. 27, No. 2, pp. 57-64, 1978.
- [14] N. Funabiki, N. Okutani and S. Nishikawa, "A three-stage heuristic combined neural-network algorithm for channel assignment in cellular mobile systems", *IEEE Trans Veh. Tech.*, vol. 49, No. 2, pp 397-403, 2002.
- [15] H.G. Sandalidis, P.P. Stavroulakis and J. Rodriguez-Tellez, "An efficient evolutionary algorithm for channel resource management in cellular mobile systems", *IEEE Trans. Evolutionary computation*, vol. 2, no. 4, pp. 125-137, 1998.
- [16] P.T. Runarsson and Xin Yao, "Stochastic ranking for constrained evolutionary optimization", *IEEE Trans. Evolutionary computation*, vol. 4, no. 3, pp. 284-294, 2000.
- [17] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. New York: Wiley, 1968.
- [18] S. Kazarlis and V. Petridis, "Varying fitness functions in genetic algorithms: Studying the rate of increase in the dynamic penalty terms", in *Parallel Problem Solving from Nature*. Berlin, Germany: Springer, 1998, vol. 1498, Lecture Notes in Comput. Sci., pp. 211-220.
- [19] W. Siedlecki and J. Sklansky, "Constrained genetic optimization via dynamic reward-penalty balancing and its use in pattern recognition", in *Int. Conf. Genetic Algorithms*, 1989, pp. 141-149.

-
- [20] Z. Michalewicz, N. Attia, "Evolutionary optimization of constrained problems", in *Proc. 3rd Annu. Conf. Evolutionary Programming*, A. V. Sebald and L. J. Fogel, Eds. River Edge, NJ: World Scientific, 1994, pp. 98-108.
- [21] J. Joines and C. Houck, "On the use of nonstationary penalty function to solve ononlinear constrained optimization problems with GAs", in *Proc. IEEE Int. Conf. Evolutionary Computing*. Piscataway, NJ: IEEE, 1994, pp. 579-584.
- [22] C. R. Reeves, "Genetic algorithms for the operations researcher", *INFORMS J. Comput.*, vol. 9, no. 3, pp. 231-247, 1997.
- [23] P. D. Surry and N.J. Radcliffe, "The COMOGA method: Constrained optimization by multiobjective genetic algorithms", *Contr. Cybern.*, vol. 26, no. 3, 1997.
- [24] E. Camponogara and S.N. Talukdar, "A genetic algorithm for constrained and multiobjective optimization", in *Proc. 3rd Nordic Workshop Genetic Algorithms and Their Appl. (3NWGA)*, J.T. Alander, Ed. Vaasa, Finland: Univ. Vaasa, Aug. 1997.
- [25] K. Deb, "An efficient constrained handling method for genetic algorithms", in *Computer Methods in Applied Mechanics and Engineering*, 1999.
- [26] F. Jimenez and J.L. Verdegay, "Evolutionary techniques for constrained optimization problems", in *Proc. 7th Eur. Congr. Intelligence Techniques and Soft Computing (EUFIT'99)*. Berlin, Germany: Springer-Verlag, 1999.

-
- [27] H. -P. Schwefel, *Evolution and Optimum Seeking*. New York: Wiley, 1995.
- [28] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: Wiley, 1997.
- [29] D.-E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-esley, 1989.
- [30] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: Univ. Michigan Press, 1975.
- [31] W.K. Lai and G.G. Coghill, "Channel assignment through evolutionary optimization", *IEEE Trans. Veh. Technol.*, vol. 45, no. 1, pp. 91-96, 1996.
- [32] A. Gamst, R. Beck, R. Simon and E.G. Zinn, "An integrated approach to cellular radio network planning", *Proc. 35th IEEE Veh. Technol. Conf.*, pp. 21-25, 1985.
- [33] W.K. Hale, "Frequency assignment: theory and application", *Pro. IEEE*, vol. 68, pp. 1497-1514, Dec., 1980.
- [34] M.R. Garey and D.S. Johnson, "The complexity of near-optimal graph coloring", *J. ACM*, vol. 23, pp. 43-49, 1976.
- [35] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biol. Cybern.*, vol. 43, pp. 59-69, 1982.
- [36] T. Back, *Evolutionary Algorithms in Theory and Practice*. New York: oxford Univ.

Press, 1996.

- [37] C.Y. Ngo and V.O.K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm", *IEEE Trans. Veh. Technol.*, vol. 47, pp. 163-172, Feb 1998.
- [38] J.S. Kim, S.H. Park, P.W. Dowd and N.M. Nasrabadi, "Channel assignment in cellular radio using genetic algorithm", *Wireless Personal Commun.*, vol.3, no.3, pp.273-286, Aug. 1996.
- [39] M. Cuppini, "A genetic algorithm for channel assignment problems", *Eur. Trans. Telecomm. Related Technol.*, vol.45, no.1, pp.91-96, 1996.
- [40] K.A. Smith, "A genetic algorithm for the channel assignment problem", *IEEE Global Tech. Conf.*, vol.4, 1998.
- [41] J.-K. Hao and R. Dorne, "Study of genetic search for frequency assignment problem", Technical Paper, EMA-EERIE, Parc Scientifique Georges Besse, 1995.
- [42] D. Beckmann and U. Killat, "A new strategy for the application of genetic algorithms to the channel assignment problem", *IEEE Trans. Veh. Technol.*, vol.48, no.4, July 1999.
- [43] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos", *Neural Networks*, vol.8, no.6, pp.915-930, 1995.

-
- [44] C.R. Reeves, "Modern Heuristic Techniques for Combinatorial Problems", *Oxford: Blackwell*, 1993.
 - [45] M.R. Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", San Francisco: W.H. Freeman, 1979.
 - [46] O. Watanabe, "Approximate solvability of NP-like problems", *Operat. Res.*, vol.39, pp.105-111, 1994.
 - [47] R.M. Karp, "Reducibility among combinatorial problems", in R.E. Miller and J.W. Thatcher (Eds), *Complexity of Computer Computations*, New York: Plenum Press, 1972.
 - [48] P. Crescenzi and V. Kann, "NPO problems: definitions and preliminaries", A compendium of NP optimization problems, 1998.
 - [49] T. Ibarake, "Combinatorial optimization and its complexity", *J. Japan. Soc. Artif. Intellig.*, vol 18, pp. 335-341, 1994.
 - [50] C.H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity classes", *J. Comput. System Sci.*, vol. 43, pp. 425-440.
 - [51] S. Aora, C. Lund, R. Motwani and M. Szegedy, "Proof verification and hard-ness of approximation problems", *Proc. 33rd IEEE Annual Symp. on Foundations of Computer Science*, pp.14-33, 1992.

-
- [52] S.H. Zanakis, J.R. Evans and A.A. Vazacopoulos, "Heuristic methods and applications: a categorized survey", *JEurop. J. Operat. Res.*, vol.43, pp.88-110, 1989.
- [53] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by simulated annealing", *Science*, vol.220, pp.671-680, 1983.
- [54] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.6, pp. 721-741, 1984.
- [55] J.J. Hopfield and D.W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol.52, pp.141-152, 1985.
- [56] D.M. Well, "Solving degenerate optimization problems using networks of neural oscillators", *Neural Networks*, vol.5, pp.949-959.
- [57] C. Peterson and B. Soderberg, "Artificial neural networks", In C.R. Reeves (ED.) *Modern Heuristic Techniques for Combinatorial Problems*, Oxford: Blackwell, 1993.
- [58] D.E. Rumelhart, J.L. McClelland and PDP Research Group, "Parallel Distributed Processing", vol.1-2, Cambridge, MA: MIT Press, 1986.
- [59] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. USA*, vol.79, pp.2554-2558, 1982.

-
- [60] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, vol.81, pp.3088-3092, May, 1984.
- [61] Y. Akiyama, A. Yamashira, M. Kajiura, Y. Anzai and H. Aiso, "The Gaussian machine: a stochastic neural network for solving assignment problems", *J. Neural Network Comput.*, vol.2, pp.43-51, 1991.
- [62] K. Aihara, "Chaotic neural networks", *Bifurcation phenomena in nonlinear systems and theory of dynamical system*, H. Kawakani (Ed.), pp.143-161, 1990.
- [63] L. Wang and F. Tian, "Noisy chaotic neural networks for solving combinatorial optimization problems", *Proc. International Joint Conference on Neural Networks*. (IJCNN 2000, Como, Italy, July 24-27, 2000)
- [64] C.A. Skarda and W.J. Freeman, "How brains make chaos in order to make sense of the world", *Brain and behavioral Science*, vol. 10, pp. 161-195, 1987.
- [65] L. Wang and J. Ross, "Oscillations and chaos in neural networks: an exactly solvable model." *Proc. National Academy of Science (USA)*, vol. 87, pp.9467-9471, 1990.
- [66] Y. Yao and W.J. Freeman, "Model of biological pattern recognition with spatially chaotic dynamics," *Neural Networks*, vol. 3, pp.153-170, 1990.

-
- [67] K. Aihara, T. Takabe and M. Toyoda, "Chaotic neural networks," *Physics Letters A*, vol.144, no.6, 7, pp.153-170, 1990.
- [68] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos," *Chaos*, vol.2, no.3, p.377-386, 1992.
- [69] I. Tsuda, "Dynamic link of memory – chaotic memory map in nonequilibrium neural networks", *Neural Networks*, vol.5, pp.313-326, 1992.
- [70] J. Tani, "Proposal of chaotic steepest descent method for neural networks and analysis of their dynamics", *Trans. Inst. Electron. Inf. Commun.*, vol. J74-A-8, pp.1208-1215, 1991.
- [71] L. Wang and D.L. Alkon, (eds.), "Artificial neural networks: Oscillations, chaos, and sequence processing", (IEEE Computer Society Press), Los Alamitos, California, 1993.
- [72] H. Nozawa, "Solution of the optimization problem using the neural network model as a globally coupled map", in *Towards the Harnessing of Chaos*, M. Yamaguti (ed), pp.99-114, 1994.
- [73] L. Chen and K. Aihara, "Transient chaotic neural networks and chaotic simulated annealing," M. Yamaguti (ed.), *Towards the Harnessing of Chaos*. Amsterdam, Elsevier Science Publishers B.V. pp.347-352, 1994.
- [74] Y. Horio and K. Suyama, "Experimental verification of signal transmission using

-
- synchronized SC chaotic neural networks,” *IEEE Trans. On Circuits and Systems I: Fundamental Theory and Applications*, vol.42, no.7, pp.393-395, 1995.
- [75] M. Gasegawa, T. Ikeguchi, and K. Aihara, “Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems,” *Phys. Rev. Lett*, vol.79, pp.2344-2347, 1997.
- [76] Hongtao Lu, Yongbao He and Zhenya He, “A chaos-generator: analyses of complex dynamics of a cell equation in delayed cellular neural networks,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.45, no.2, pp.178-181, Feb. 1998.
- [77] I. Toluda, K. Aihara, and T. Nagashima, “Adaptive annealing for chaotic optimization”, *Phys. Rev. E*, vol.58, pp.5157-5160, 1998.
- [78] S. Haykin and J. Principe, “Making sense of a complex world [chaotic events modeling],” *IEEE Signal Processing Magazine* vo.15, no.3, pp.66-81, May, 1998.
- [79] L. Wang and K. Smith, “On chaotic simulated annealing,” *IEEE Trans. Neural Networks*, vol.9, no.4, pp.716-718, 1998.
- [80] K. Jin’no, T. Nakamura, and T. Saito, “Analysis of bifurcation phenomena in a 3-cells hysteresis neural network,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no.7, pp.851-857, July 1999.
- [81] L. Chen and K. Aihara, “Global searching ability of chaotic neural networks.”

-
- IEEE trans. Circuits and Systems – I: Fundamental Theory and Applications*, vol. 46, no. 8, August, pp.974-983, 1999.
- [82] T. Kwok and K.A. Smith, "A unified framework for chaotic neural network approaches to combinatorial optimization," *IEEE Trans. Neural Networks*, vol.10, no.4, pp.978-981, 1999.
- [83] R. Kozma and W.J. Freeman, "Encoding and recall of noisy data as chaotic spatio-temporal memory patterns in the style of the brains," *Proc. Int. Joint Conf. Neural Networks*, Como, Italy, July 24-27, vol.5, pp.33-38, 2000.
- [84] K. Hirasawa, J. Murata, J. Hu and C.Z. Jin, "Chaos control on universal learning networks," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no.1, pp.95-104, Feb., 2000.
- [85] K. Tanaka, Y. Horio, and K. Aihara, "A modified algorithm for the quadratic assignment problem using chaotic-neuro-dynamics for VLSI implementation," *Proc. International Joint Conference on Neural Networks (IJCNN'01)*, vol.1, pp.240-245, 2001.
- [86] Lipo Wang, S.Arunkumaar and Wen Gu, "Genetic algorithms for optimal channel assignment in mobile communication", *Proceeding of the 9th internationally conference on neural information processing (ICONIP'02)*, vol.3, pp.1221-1225
- [87] Sa Li and Lipo Wang, "Channel assignment for mobile communications using stochastic chaotic simulated annealing", *The 6th International Work-Conference on Artificial and Natural Neural Networks*, IWANN 2001, pp. 757-764.

-
- [88] Zhenya He, Yifeng Zhang, Chengjian Wei and Jun Wang, "A multistage self-organizing algorithm combined transiently chaotic neural network for cellular channel assignment", *IEEE Trans. Vehicular Technology*, vol.51, no.6, pp.1386-1396, Nov. 2002.
- [89] Sa Li, "Solving combinatorial optimization problems by soft computing", M. E. thesis, Nan yang Technological University, 2002
- [90] Z. Michalewicz, G. Nazhiyath, and M. Michalewicz, "A note on usefulness of geometrical crossover for numerical optimization problems," in *Proc. 5th Annu. Conf. Evolutionary Programming*, L. J. Fogel, P. J. Angeline, and T. Bäck, Eds. Cambridge, MA: M.I.T. Press, 1996, pp. 305-312.
- [91] Homaifar, Lai, and Qi, "Constrained optimization via genetic algorithms", *Simulation*. 62, 1994, pp. 242-254.
- [92] Joines. J. A. and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs", in *Proc. Evolutionary Computation Conference- Poster Sessions*, part of the IEEE World Congress on Computational Intelligence. Orlando. 1994, pp 579-584.
- [93] S.Arunkumaar, "Genetic algorithm for optimal channel assignment for mobile communications", M.S. dissertation, 2002.

Author's Publications

1. Lipo Wang, S.Arunkumaar and Wen Gu, "Genetic algorithms for optimal channel assignment in mobile communication", Proceeding of the 9th internationally conference on neural information processing (ICONIP'02), vol.3, pp.1221-1225
2. Lipo Wang and Wen Gu, "Genetic Algorithm with Stochastic Ranking for optimal channel assignment in mobile communications", Accepted by the International Symposium on Computational and Information Sciences, 2004.