PARTICLE SWARMS AND THE FREQUENCY ASSIGNMENT PROBLEM

by

WILLIAM BEZUIDENHOUT

DISSERTATION

submitted in the fulfilment
of the requirements for the degree

MAGISTER SCIENTIAE

in

INFORMATION TECHNOLOGY

in the

FACULTY OF SCIENCE

at the

UNIVERSITY OF JOHANNESBURG

SUPERVISOR: DR. G.B. O'REILLY

JUNE 2010

Contents

Part I

Background

Chapter 1

Introduction

1.1 Test

blah blah blah blah

Chapter 2

Cellular Technology

2.1 Introduction

In the information age we currently live in almost every device has some sort of wireless technology it uses to provide a specific service. Radios for audio entertainment; Television remotes to change channels; Cellular phones for communication; Wireless access points to create wireless LAN's [2]. Wireless technology is now part of our everyday life.

The popularity and rapid adoption of wireless technology hasn't been kind to the management, planning and operation of wireless networks; It has actually worsen a problem known as the Frequency Assignment problem (FAP) which is present in all forms of wireless communication especially in GSM Cellular networks.

In this chapter we will start of by giving a brief history of the Frequency Assignment Problem. After the brief history overview we will give a description of the problem and explain some of the underlying concepts needed to understand the Frequency Assignment Problem. We will then move on to discuss the different variants present in the current domain. Further more we will then discuss the two underlying approaches used in solving the Frequency Assignment Problem (FAP).

2.2 GSM Networks

The General System for Mobile Communications (GSM) is a system for multi-service cellular communication which is capable of providing voice as

well as data services. Most cellular networks in operation are GSM based. The primary service that GSM caters for is voice communication, but other data services such as Short Message Service (SMS), Multimedia Message Service (MMS) and Internet connectivity services such as GPRS are becoming more important [23].

GSM is one of the most widely used radio communications technologies, which is why we need to look at the history behind it in order for us to understand the domain of radio communication better.We will now present a brief history of the GSM network specification.

2.2.1 A Brief History of GSM Networks

In the early 1981 a group known as the Groupe Speciale Mobile (GSM) was established to develop a Europe wide radio communication system using the reserved 900 MHz band[1].

At the start of the GSM specification in the early 1980's it was initially thought that the system would be analogue based, but this soon changed with the *Integrated Service Digital Network* (ISDN) specification nearing completetion. As such the GSM specification started following much of the same design principles and access protocols that ISDN exhibited. After the completion of the ISDN specification and the advantages it brought to the field, it became unofficially clear that GSM would be based on digital transmission and that speech would be represented by a digital stream of 16 kbits/s [64].

Before the switch to digital transmission was finalized the GSM first wanted to evaluate the spectral efficiency of analogue and digital based transmission. Spectral efficiency plays an important part in wireless communication since the radio spectrum is a limited resource and whichever transmission technology is used, should maximise the utilization of the spectrum. Maximum utilisation is an important problem which we will discuss in detail in later sections of this chapter. The Spectral evaluation was conducted over a period of 3 years from 1984-1987. In 1987 a report was published about the 3 year evaluation and subsequently it became official that the

[1] In 1990 the United Kingdom requested that 1800MHz band be added to the scope of the GSM standard group. This variant of the GSM specification became known as the *Digital Cellular System 1800* (DCS1800) [64].

GSM system would be digital based using *Time Division Multiple Access* (TDMA) [60, 64].

By the early 1990s GSM became an evolving standard and the first GSM based network was demonstrated in 1991[2]. The following year a number of GSM networks were operating in Europe due to mobile terminals / equipment capable of operating on the networks becoming more widely available to the general public. In the same year an operator in Australia became the first non-European operator to implement a GSM based network [23].

The collective subscriber base of GSM networks surpassed the million subscriber mark in 1993. Due to this phenomenal growth in GSM network use, numerous extensions were made to the GSM specification. Some of the extensions that were made are the following [23]:

- Half rate speech telephony
- Improved SMS
- Line Identification
- Call waiting
- Call holding

The specification with these extensions defined is known as the GSM Phase 2. As the world shifted towards more digital and data intensive services it became difficult to deliver these services over GSM networks. This difficulty was due to the restriction that data could only be transmitted at 9.6 Kbps. A move to eliminate this restriction was made with the specification of GSM Phase 2+.

The new specification defined new technologies such as General Packet Radio System (GPRS) and *Enchanced Data rate for GSM Evolution* (EDGE) which were designed with the primary goal of making more bandwidth available for data transmission. These new technologies have an inherent requirement that there be a higher signal to noise ratio present at transceivers. This requirement has an impact that effects radio interfaces and more importantly Frequency planning [23].

[2]Near the end of 1991 the GSM group was renamed to *Speciale Mobile Group* (SMG) to eliminate confusion with the standard and the group.

The actual signal to noise ratio at a receiver is dependent on a number of factors that include [2]:

- Frequency used at the transceiver
- Strength of the signal
- Weather conditions
- Shape of the surrounding environment
- Direction of the transmission

Even taking these factors into account the calculation of the signal to noise ratio at a transceiver is not trivial. For a more in depth discussion on the calculation the reader is directed at the survey by [2].

As the GSM standard matured as a cellular technology, industry experts already began specification of the next generation of cellular networks which would in time, replace the GSM cellular system. The specification of a new standard is considered to be a natural evolution of the technology. Each standard is designed with specific use cases in mind as to what its users might want to do as well as what is possible with the technology at the designers disposal. As time goes by, the technology improves and users habits and needs change, thus the standard must be improved upon to serve these new needs and incorporate new technology.

The *Universal Mobile Telecommunications System* (UMTS) can be considered the 3rd generation (3G) of cellular networks. UMTS was designed from the beginning to operate in parallel with the legacy GSM system. This decision was made to make the deployment of the system as hassle free as possible for the network operators. The first standard of the UMTS was issued in the beginning of 2000 and subsequently most modern networks are based on it or are migrating their networks to it.

UMTS is a large improvement of the GSM in two areas namely Data Transmission bandwidth and Frequency Planning due to UMTS utilising *DS-CDMA* (direct sequence code division multiple access) and *WCDMA* (Wide Band Code Division Multiple Access). The higher data transmission speed (2 Mb/s) can be attributed to UMTS using the DS-CDMA scheme. The scheme also allows more users to be served than previous generation of networks. A direct consequence of WCDMA which sends information over

a wide-band of 5 MHz is that no frequency planning problem comparable to that of GSM has to be solved [23, 95].

In this section we gave a brief overview of the history of the GSM Network specification. In the next section we will present an explanation of the topology of GSM network as well as look at the underlying problems present in a GSM networks.

2.3 Topology of a GSM Network

GSM networks consists of a variety of different subsystems to realise the goal of establishing a radio communication link between two parties. The hierarchy of systems and their respective connections to each other is illustrated in figure 2.1. We will now briefly explain each subsystem.

Mobile Station (MS)

A Mobile station (MS) as it is defined in the GSM spec refers to any mobile device that is capable of of making and receiving calls on a GSM network. The MS is the main gateway for a user to gain access to the GSM network. The MS has two features which play an important role throughout the GSM Network, namely:

Subscriber Identification Module (SIM) — Usually inserted into a mobile devices. The SIM contains the *International Mobile Subscriber Identity* (IMSI) and is used throughout the network for Authentication as well as being a key part in providing encrypted transmissions.

International Mobile Equipment Identity(IMEI) — Used to identify mobile station equipment. Primarily used in the denial of service to equipment that has been blacklisted[3] and tries to gain access to the network.

The MS has the capability to change the transmission power is uses from its base value to a maximum value of 20 mW. The change in transmission power is automatically set to the lowest level by the Base Transceiver Station to ensure reliable communication after evaluating the signal strength as measured by the MS [60].

[3]Equipment can be blacklisted for a variety of reasons e.g. theft

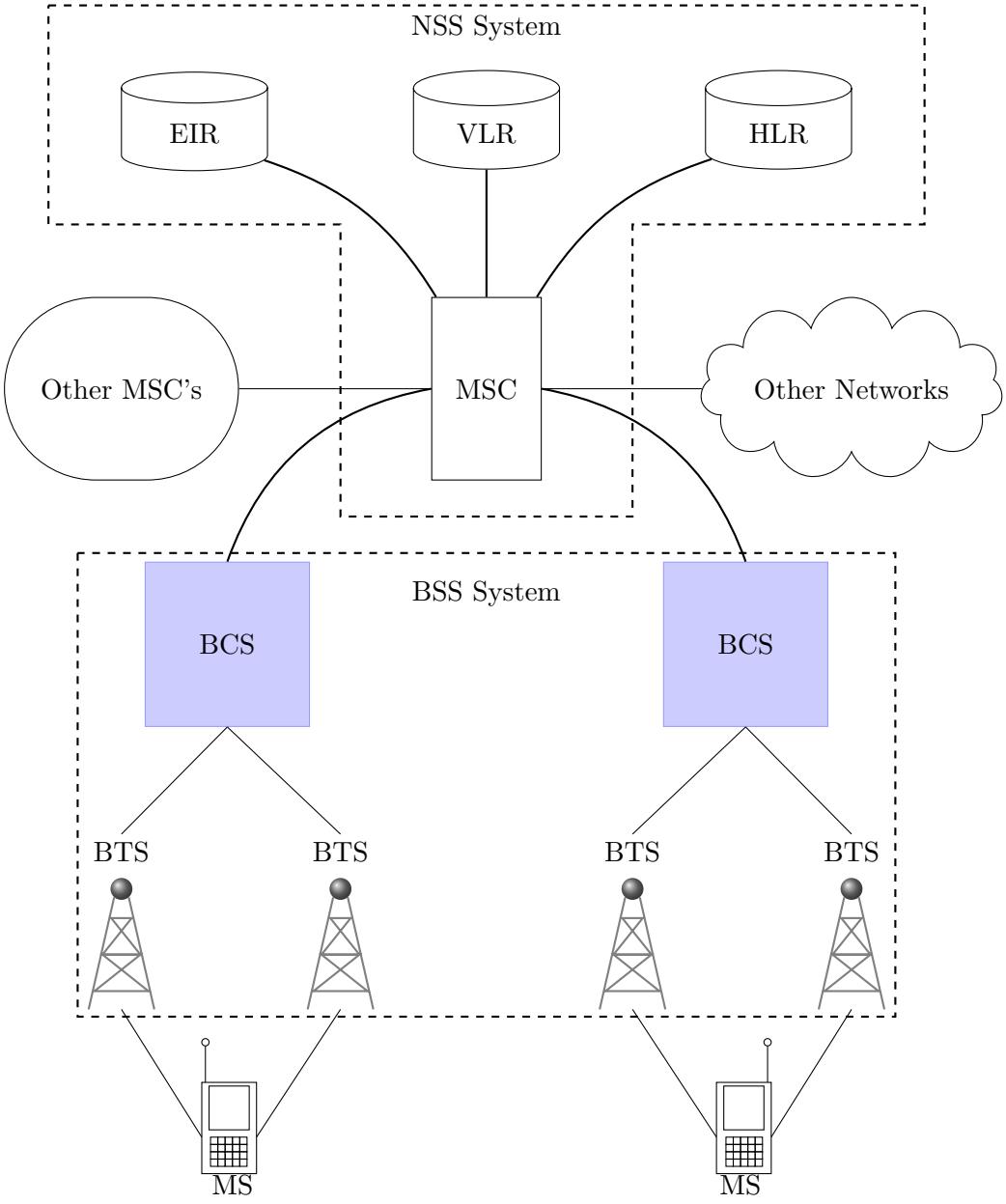

Figure 2.1: GSM Architecture

2.3.1 Base Station Subsystem (BSS)

According to the GSM Phase 2+ specification this system is viewed by the *Mobile Switching Centre* (MSC) through an Abis radio interface as the system responsible for communicating with Mobile Stations in a particular location area. The BSS usually consists of one *Base Station Controller* (BSC) with one or more *Base Transceiver Stations* (BTS) which it controls. The communication link between the MSC and BSC is the called the A-interface and the communication link between the BSC and BTS is called the Abis interface. The definition of these communication interfaces is beyond the scope of this disseration, the interested reader is directed to the book *GSM System Engineering* by Asha Mehrotra. A BTS has similar equipment to that of a Mobile Station. Both have transceivers, antennas and the necessary functions to perform radio communication.

In a GSM network the Service Area (SA) is subdivided into Location Areas(LA's) which are then futher divided into smaller radio zones called cells [78]. A cell is served by only one BTS and is usually regarded to be in the center of a cell as can be seen in figure 2.2. Even though cells are modelled as being hexagons (see figure 2.2) the actual coverage area of a cell has no predefined regular shape. Futhermore a cell is divided into 1 to 3 service sectors and each sector is allocated an antenna/transceiver [60]. Depending on how many sectors are at a cell, the operating angle of the antennae needs to be adjusted accordingly to ensure 360 degree service. If there is only one sector an omni-directional antenna is used, otherwise the antennae operating angle are adjusted to $\frac{360^\circ}{n}$ where n is the amount of antennae [23].

Each sector operates one or more elementary transceivers called TRXs. The amount of TRXs per sector is determined by the expected peak traffic demand that the cell must be able to handle. Each TRX can handle 7 to 8 communication links or calls in parallel except the first TRX, which handles fewer calls than normal due to it being responsible for transmitting cell organisation and protocol information [23]. TRXs are able to handle 7-8 calls in parallel due to the use of *Frequency Division Multiplexing* (FDM) and *Time Division Multiplexing* (TDM) schemes. TRXs are assigned channels which enable them to provide conversion between digital traffic data on the network side and the radio communication between Mobile Stations and the GSM network. [11, 55]

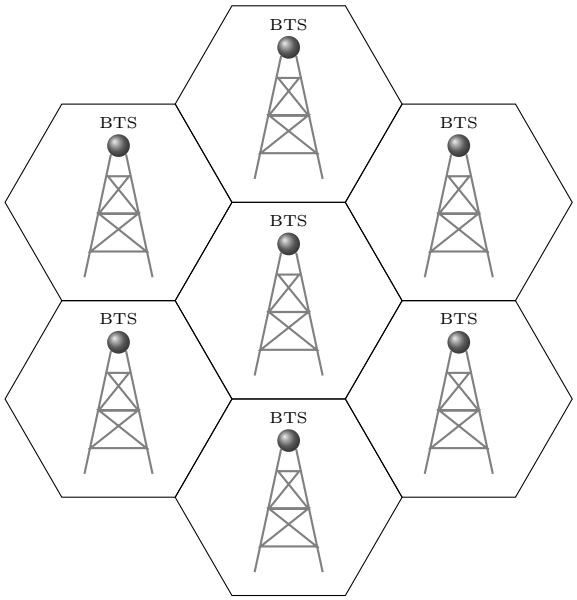

Figure 2.2: Cells with BTS's

2.3.2 Mobile Switching Centre (MSC)

The MSC is at the heart of cellular switching system and forms part of the *Network Switching Subsystem* (NSS). The MSC is responsible for the setting up, routing and supervision of calls between GSM subscribers. The MSC has interfaces on the one side to communicate with GSM subscribers (through the BSS) and on the other it has interfaces to communicate with external networks. The MSC interfaces with external networks to utilise their superior capability in data transmission as well as for the signalling and communication with other GSM entities [64].

The most basic functions that an MSC is responsibile for in a network are the following [70]:

- Voice call initialization,routing,control and supervision between subscribers.
- Handover process between two cells.
- Location updating.
- MS authentication.
- SMS delivery.

- Charging and Accounting of services used by subscriber.
- Notification of other network elements.
- Administration input or ouput processing functions.

To achieve most of these functions the MSC has an integrated *Visitor Location Register (VLR)* database that stores call setup information for any MS that is currently registered for service with the MSC [64, 70].

The VLR retrieves this information from the *Home Location Register (HLR)* which contains all the registered GSM subcriber information for the network. This information enables the MSC to quickly retrieve the nesaccery information to setup a call between two entites [60, 78].

A requirement for being able to communicate with other network elements such as *Public Switching Telephone Networks* (PSTN) is the ability to multiplex and demultiplex signals to and from such network elements. This operation is a necacity, since the incoming or outgoing connection bitrate from the source entity might either be to low or to high for the receiving entity.

A typical scenario where this operation proves vital is when a mobile subcriber makes a call to a subcriber on a PSTN. The connection bit rate needs to be changed at the MSC from a wireless connection bitrate to a bitrate suitable for transmission over a PSTN.

2.3.3 Network Databases

The HLR, AUC (Authentication Center) and EIR (Equipment Identity Register) are the 3 'back-end' databases which stores and provides information for the rest of the GSM Network. We will now briefly discuss what the purpose of each database is and its core functions.

Home Location Register (HLR) The HLR is a database that permenantly stores information pertaining to a given set of subscribers. The HLR needs to store a wide range of subcriber parameters because it is the reference source for anything GSM subscriber related in the network.

Subscriber parameters that are stored in the database include: Billing information, routing information, identification numbers, authentication parameters,subscribed services. The following information is also stored but

the information is of a temporary nature and can change at anytime: Current VLR and MSC the subscriber is registered with; Wheter the subscriber is roaming [60].

Authentication Center (AUC) The Authentication center is the entity in the GSM network that performs security functions and thus stores information that enables it to provide secure over the air communication. The information that is stored contains authentication information as well as keys that are used in ciphering of information.

During an authentication procedure no ciphering key is ever transmitted over the air, instead a challenge is issued to the mobile who needs to be authenticated. This callenge requires the mobile station to provide the correct *Signed Response*(SRES) with regard to the random number generated by the AUC. The random number and ciphering keys that are used change with each call that is made, thus an attacker would gain nothing by intercepting a key, since it will change with the next call [60].

Each mobile that is registered in the HLR database needs to be authenticated and each call that is instansiated needs to retrieve keys from the AUC to establish a secure communication link. The AUC is sometimes included with HLR to allow for fast communication between the two entities [60].

Equipment Identity Register (EIR) The EIR is a database that stores the IMEI numbers of all registered mobile equipment that has accessed the network. Only information about the mobile equipment is stored, nothing about the subscriber or call is stored in the database.

Typically there is only one EIR database per network and interfaces to the various HLR database contained in the network. The IMEI's are grouped into 3 categories: *White List*, *Black List* and the *Gray List*. The White list contains only the IMEI numbers of valid MS's; the Black List stores the IMEI numbers of equipment that has been reported stolen and the Gray List stores the IMEI numbers of equipment that has some fault (faulty software, wrong make of equipment).

2.3.4 GSM Network Management entities

In a GSM network most of the elements that form part of and make the network function are often distributed in a wide geographical area to provide

the best network converage for the customer.

For a network to function properly and efficiently network engineers need to be kept up to date on the state of the network and be alerted if *any* problems occur. For this purpose there exists two systems in the GSM network architecture that allows for this functionality required by network engineers.

The one system is called the Operations and Management center which is responsible for centralized regional and lcoal operational and maintenance activities. The other system called the Network Management System unlike the OMC provides global and centralized managemenent for operations and maintenance of the network suppored by the OMCs [60].

We will not discuss the OMC and NMC in a bit more detail where we'll define the most critical functions they peform.

Operational and Managemenet Center The OMC is capable of communicating with GSM entities using two protocols namely SS7 and X.25. The SS7 protocol is usually used when the OMC is communicating within the GSM network over short and medium distances. The X.25 protocol is used for large external data transfers. All communication where the OMC is involved typically occurs over fixed line networks and/or leased lines. The OMC is usually used for day to day operation of a network [60].

The OMC has support for alarm handling. An alarm in a GSM network goes of whenever a predefined expected condition does occurs. Engineers are able to define the severity of an alarm which defines who and what is futher alerted when and if the alarm is escalated to a higher level [60].

To give one an idea of when and why and alarm goes of consider the following scenario: The MSC controls a set of BSS systems. Now for some reason a certain region experiences a power blackout. All the BSS affected by the blackout switch over to reserve power if availble. A first alarm is sounded to let the engineers / network know that the BSS is using reserve power. When the BSS reserve power is depleted the MSC sounds an alarm letting the network know that a specific BSS cannot be contacted.

Typically in the above scenario the severity of the first alarm will be of a medium priority. The second alarm is much more serious and its severity will be of a high priority.

The OMC is also capable of fault management in the GSM network. The OMC is able to activate,deactivate,remove and restore a service manually or automatically of network devices. Various tests can be run as well as diagnostic information can be retrieved on the network devices to detect any current or future defects [60].

Network Management Center The NMC is similar to the OMC but it is not restricted to only regional GSM entities as it is in charge of the all GSM entities in the network. The NMC provides traffic management for the global network and also monitors high priority alarms such as overloaded or failed network nodes. It is usually used in long term planning of a network, but it has the capability to perform certain OMC functions when an OMC is not staffed.

2.4 GSM Network problems

2.5 Summary

Chapter 3

The Frequency Assignment Problem

3.1 Introduction

The Frequency Assignment Problem (FAP)[1] is a generalisation of the graph-colouring problem and is subsequently an NP-hard problem. This is because one has a finite amount of frequencies which needs to be assigned to antennae/transceivers (TRX's) where the amount of transceivers to be assigned frequencies greatly out weigh the amount of available frequencies.

It is inevitable that a network will have interference and we can thus only minimise the amount of interference that might occur on the network - an optimisation problem. Using exact algorithms to find a solution is not practical since the time to find a solution will be polynomial. Generally Metaheuristic algorithms are used to find optimal solutions to NP-hard problems [55]. We will discuss Metaheuristic algorithms which are generally used to find solutions for NP-hard problems in Chapter 4.

A contributing factor to the difficulty of the FAP is due to the scarcity of usable frequencies in the radio spectrum, which forces network operators to reuse their allocated/licensed frequencies in their respective networks. The scarcity of the usable frequencies in the spectrum can be attributed to the overuse of certain bands as well as large scale reuse of frequencies in networks. This has put strain on the spectrum and has complicated the

[1] Also known as Automatic Frequency Planning (AFP) or Channel Assignment Problem (CAP) [55]

management of networks significantly because interference is more likely to occur.

Frequency assignment is the last step in a long process of network setup. Before frequencies are assigned base stations need to be placed and need to be configured, which include azimuth and tilt of the antenna for optimal network coverage. After the base stations are configured they need to be allocated a certain amount of transmitters to achieve a target network capacity [22].

Frequency assignment is only a means to achieve the targeted network coverage as well as network capacity.

In this section we gave a brief introduction as to what the Frequency Assignment Problem is and why it occurs as well as why it is a problem. In the next section we will describe the types of Frequency Assignment in use today and we will also discuss some of the varaints of FAP and also formally define which of the variants we will concentrate on.

3.2 Frequency Assignment Types

In this section we will discuss the different methods that exist to allocate frequencies to cells in a cellular network. We will also state which method we will use through out this paper.

Within the FAP domain there exists different types of the FAP which have emerged over the years as the domain and requirements have changed. We will discuss these FAP variants in section 3.4.

There are a variety of FAP in the domain (which will be discussed in later sections of this chapter) but most of these problems can be classified into two categories based on the assignment scheme they use:

(a) *Fixed Frequency/Channel Assignment* (FFA/FCA) is the process of permanently assigning frequencies to cells (cellular towers). The frequencies assigned are fixed and cannot be changed on the fly while the network is active , since the frequencies assigned to the cell form part of a delicate frequency plan designed to keep interference to minimum.

(b) *Dynamic Frequency/Channel Assignment* (DFA/DCA) is the process of allocating channels to cells as they require it to meet the current traffic demand imposed on them by clients.

Each cell can be assigned multiple frequencies based on the amount of transmitters or TRX's it has. The amount of TRX's in a cell depends on the expected amount of traffic the particular cell must handle.

Most of the research in the FAP has concentrated on the FFA. The reason for this is because FFA is a static technique, which allows it to come up with a better solution since it has more time for calculation. FFA is also easier to implement in practise and allows the network operators to cater for the worst case scenario - heavy traffic load on the network.

The DFA is at the moment a very hard problem because the network frequency plan is constantly changing, which means as the traffic on the network increases the longer the DFA focused algorithm will take to allocate a frequency. This increase in processing time is because the algorithm has to take into account more constraints with a lower available frequency pool. DFA must do this process within seconds since a cell needs to serve clients.

Most researchers have concentrated on solving the FFA using heuristic approaches like neural networks, local search techniques and more recently meta heuristic approaches which include genetic algorithms, simulated annealing , ant colony optimisation and particle swarm optimisation.

In this section we have given a description of the Frequency Assignment Problem and introduced some concepts which we will use throughout the dissertation. In the next section we will present the Mathematics that govern the Frequency Assignment Problem.

3.3 Interference

In this section our disuccsion will focus on Interference. We will describe what interference is and why it is important for cellular networks as well as give an overview of when interference occurs.

Interference occurs when frequencies assigned to connections differ by a small margin. The amount of inference on a connection defines the *Quality of Service* (QoS). One can naturally make the deduction that the more frequencies differ used on connections in a area, the better quality of service one will experience in that area.

Cellular networks use the amount of interference on their networks as qualitative measure for their QoS. A network with high interference would

experience a lot of dropped connections/calls, which occurs when the interference is too high to sustain a connection or call for communication, consequently their QoS degrades as interfernece increases.

In the literature a variety of methods are given to calculate the amount of interference in a network. These methods range from theoretical approaches to precise measurements. Regardless of what method is used the end result which they all produce is called an *Interference Matrix* [55].

An Interference Matrix concists of a number of cell pairs (i,j), where i is the cell receiving interference and j the cell whose allocated channel is providing the interence. Each cell pair in the matrix has two corresponding values that indicate the level of interference if the *Electromagnetic constraints* are violated [2, 23, 55]. These values are usually normalized to be between 0.0 and 1.0 [22].

Primarily Interference occurs when the Electromagnetic constraints are violated, which are defined as:

Co-Channel — When a cell i and a cell j operate on the same frequency or channel interference will occur [2, 23, 55, 87, 89]. This is called co-channel interference. This constraint is the most important constraint that must not be violated to ensure proper performance and reliability of a modren cellular network [89].

Adjacent Channel — When a cell i and a cell j operate on adjacent channels, their allocated frequencies differ by one i.e. cell i operates on channel f then if cell j operates on either channel f - *1* or f + *1* then adjacent channel interference will occur [2, 23, 55, 87, 89]

Co-Site — If cell i and cell j are located at the same site, then their allocated frequency ranges must differ by a certain distance in the frequency domain. This distance is known as the reuse distance [24, 115].

A fourth constraint, kown as the Handover constraint, is also applicable in Cellular networks. This constraint imposes a separation in frequencies when one cell hands over a call to another cell. If this constrain is violated a mobile subscriber will experience a dropped call since the handover between cells fials.The above constraints only account for factors that are in our control.

Interference also occurs due to techincal limitations, natural phenonema and other external factors like other systems. Thus another constraint is

imposed on the frequency that is allocated to cell. This constraint is known as the *separation* constraint which imposes a minimum separation between frequencies assigned to a cell [23, 87]. To avoid clashes with other operator frequencies each cell may also have a set of locally forbidden frequencies which are not allowed to be used under any circumstance.

In this section we described what interfenece is and what the consiquences are of too much interfence in a network. We also laid out under which circumstances interference can occur in a wireless network. In the next section we will give a brief overview of in which industries the Frequency Assignment Problem is applicable.

3.4 FAP in the industry

In this section we will list some of the industries where the FAP is encountered. We provide a brief overview how the problem differs compared to other industries. We will also give some references to literature where the FAP and the particular domain are discussed.

3.4.1 Satelite communication

The FAP in the Satelite communication domain occurs with respect to the ground terminals that transmit and receive signals via a satelite. One would assume that the problem includes the satelite, but the problem is only concerned with the frequencies that the ground terminals use. It is interesting to note that the ground terminals can be a base station or a handheld device (e.g GPS or Satelite phone).

In Satelite communication, a signal is transmitted to one or more satelites via an uplink from a ground terminal. The signal is received by the recipient statelites and relayed to the interested ground terminals who receive the signals via a downlink.

The frequencies used by the ground terminals for uplink and downlink communication are separated by a large distance in the frequency domain - the typical distance is much larger than the bandwidth. When frequencies are assigned to transmitters, downlink transmitters are ignored and only uplink transmitters are considered [2].

A radical difference with regard to the use of frequencies compared to the standard FAP in Cellular Networks is that, frequencies are only allowed to be used once. This is sepecific to the satelite domain to avoid interference [2].

There isn't a lot of research on the FAP in Satelite Communication. One of the few recent papers on FAP in this domain is a paper by Lui et al. [54] where the authors employ a Chaotic Neural Network to minimize the interference in a satelite communication system with which they achieve very good results which in most benchmarks finds the global optimum.

Another papers conetrating on this domain is a paper by Houssin et al. [37] where the allocation of frquencies assigned in the satelite system is optimised using Space Division Multiple Access (SDMA). SDMA was developed for use in 3G networks and forms part of the Multiple Access family of techniques (CDMA,TDMA) that are in use in wireless networks today.

It is interesting to note that the authors concentrate on optimising the amount of users served by the system and not interference incurred by the allocated frequencies.

3.4.2 Wireless mesh networks and WLANs

Wireless mesh networks and WLANs[2] are the most recent applications where the FAP is encountered.

Multiple WLANs are increasingly being used to provide backbone support for large fixed line netwroks, enterprise networks, campuses and metropolitan areas. To be able to provide backbone support for these networks, a primary design goal when desiging and deploying these networks is capacity. A limiting factor for WLAN capacity is interference which affects multihop hop settings. Thus the overall network interference needs to be minimized to increase the capacity of the network [96].

Most wireless networks operate on the IEEE 802.11 a/b/g standard. A IEEE 802.11n standard is available but hasn't been finalized yet, even though one can already find wireless hardware operating on this standard. According to the IEEE 802.11g standard only 13 Frequencies are available for use and in some geographical areas a futher limiting constraint is imposed

[2]Both applications use the same standard and encounter similar problems in their respective domains.

which only allows a certain subset of frequencies to be used in the particular area [2].

Typical approaches allocating frequencies include using DCA and FCA[3]. DCA isn't very popular since the dynamic switching of channels lowers the response time on commidity hardware since there is a delay in miliseconds when switching channels. Typical packet transmission times are in microseconds. To garuntee uptime and high responsiveness, FCA is the preferred approach [96].

The FAP in Wireless Mesh Netowkrs and WLANs differ to the standard problem in that it introduces an extra constriant. Channels assigned to links on a node cannot be more than the available interfaces on that particular node. This constraint is known as the *interface constraint* [96]. Another aspect to consider is the placement of access points (AP) in the network, which is similar to the problem cellular networks face with regard to base station placement [2].

There is a wealth of literature on Wireless Mesh Networks. In research done by Subramanian et al. [96] the authors formulate a lower bound using semi-definite techniques and linear programming. Using these lower bounds with their discussed algorithms they get very promising results on a simulation benchmark (ns2). Their discussed solution imposes no specific hardware or topology changes in the wireless network.

In research conducted by Chen et al. [15], the authors follow a different route than traditional proposed algorithms with regard to interference. The authors present algorithms that focus on the interference as perceived by the user and not the Access Point (AP). The authors also use site specific knowledge provided by Blueprints, Google Earth and Google Maps in their algorithms to predict potential path loss when allocating a frequency to a AP.

3.4.3 Military field communication

In a Military context the FAP is a very difficult problem to be solved due to its dynamic nature. During deployment connections need to be established rapidly between nodes with not gauruntee that the nodes would stay static

[3]Discussed in section 3.2 on page 24

at locations. Usually nodes are military field phones can be any transceiver device.

Due to the nature of the problem the DCA scheme is used to allocate frequencies to nodes. The Military FAP differs due to the property that any of the nodes are mobile and can move at any moment to a new location, potentially interfering with another connection. Two frequencies need to be assigned to each connection that is established, one for each direction of communication. These allocated frequencies must also differ by a certain distance in the frequency domain to prohibit alternating directions of communication interfering.

A lot of literature can be found on Military field communication. This is due to two organizations CELAR[4] and EUCLID[5] making data available to various research groups and allowing them to develop algorithms for frequency assignment.

A comprehensive study by Dupont et al. [21] on the 36 instances of real life data obtained from CELAR. The Authors state that the CELAR data actually has 3 subproblems which occurs in seperate stages. In the first stage a Constrained Satisfaction Problem is encountered when assigning initial frequencies. The second problem occurs when new links are established and frequencies need to be assigned, this is known as the second stage. The last and final stage occurs when a new link cannot be assigned a frequency and thus a repiar is needed. For each stage the authors developed algorithms to try and optimily solve it to produce and overall optimal solution.

For the instances made available by EUCLID the study by Aardal et al. [1] provides results from various groups who worked on the instances provided, which is known as the CALMA project. Various optimization and approximations algorithms were implemented by the research groups and new lower bounds were also found. The authors present each algorithm implemented by the underlying research group. Results are shown for Minimum Interference problems as well as Minimum Span problems.

3.4.4 Television and Radio Broadcasting

The FAP encountered in broadcasting very much resembles the problem domain found in Cellular networks. The only notable difference is the required

[4]Centre Electronique de L'Armement

[5]European Cooperation on the Long Term Defense

distance allocated frequencies must differ in the frequency domain is larger in broadcasting than in cellular networks [2].

Since the problem resembles the problem found in Cellular networks, there are few articles that specifically discuss frequency assignment in broadcasting as a main topic. Research that specifically discusses FAP in broadcasting is presented by Idoumghar and Schott [40]. The authors present a distributed hybrid genetic algorithm and a cooperative distributed tabu search algorithm. They compare these algorithms with their sequential counterparts of their algorithms and with a ANTS algorithm. The benchmark instances they used were provided by the TDF-C2R Broadcasting and Wireless research center.

3.4.5 Cellular Communication

Cellular communication[6] can be considered the main driving force with regard to research in the Frequency Assignment domain. As new standards are developed and used in 3G networks, in general a frequency assignment problem still needs to be solved since these techniques do not eliminate interference entirely, but the do make it *less* likely to occur.

One such technique that is mostly used in GSM networks is called Frequency Hopping, which as the names emplies occurs when the transmitters "hops" onto different frequencies according to a predefined sequence of frequencies. The frequency can change per packet if the underlying hardware can handle it otherwise it switches per connection [2, 23, 67].

The FAP in the Cellular domain is the most researched topic and is considered the default domain of the problem. As such, most of the literature concentrates on this domain and one can find a lot of research in the literature presenting viable algorithms that produce real world solutions [23].

Because the problem is NP-Hard most presented algorithms are either of the metaheuristic type or more recently of the swarm intelligence type. Both of these algorithmic types are discussed in Chapter 4 and 5 respectively.

Besides optimizing algorithms, there is also a wealth of literature on upper and lower bounds for the FAP. Using lower bounds in FAP orientated

[6] An overview of a Cellular Communication technology called GSM is presented in Chapter 2.

algorithms can produce very favourable results as demonstrated in the paper presented by Montemanni and Smith [68]. Using a lower bound in conjunction with their algorithm they achieved a new optimum in a variety of benchmarks, most notably in the COST 259 benchmark.

Other papers in the literature contribute by providing different modeling techniques such as the research done by Borndrfer et al. [11]. The interested reader who wants to know more about the problem domain, general modeling techniques used as well as the most common algorithms used in directed to the study by Aardall et al [2].

In this section we discussed the different industries in which the Frequency Assignment Problem is encountered and also gave a brief description on how the problems are different with regard to what constraint each problem domain imposes on the frequencies for the particular industry. We also gave some brief references to some literature where the FAP is discussed with regard to the particular domain. In the next section we will give a brief description of the different types of Frequency Assignment Problems as well as give a small discussion on the literature found for the individual problems.

3.5 Frequency Assignment Problem types

We will start of giving a brief overview on one of the first and oldest problems in the domain and we will end of discussing the problem we will base our implementation on. In this section we will give a brief explanation on some of the various problems that exist for the FAP. We will start of giving a brief overview on one of the first and oldest problems in the domain and we will end of discussing the problem we will base our implementation on.

3.5.1 Minimum Order FAP

The Minimum Order FAP (MO-FAP) was the first FAP that emerged in the 70's. The MO-FAP is concerned with assigning frequencies to transmitters while interference is minimized as well as minimizing the amount of different frequencies that are used.

In MO-FAP frequency re-use is prioritised and the usage of a frequency has a certain cost associated with it. The reason for this is because when

the wireless network industry started out, operators were billed according to the amount of different frequencies they used. In the beginning frequencies weren't cheap since they were sold per unit [2, 67].

Over the years as the law governing the wireless spectrum changed and new technology as well standards emerged, thus MO-FAP has lost its relevancy. Companies aren't billed according to the different frequencies they use, but they purchase licenses from a regulatory body. This license usually stipulates what frequency band the network is allowed to use.

In Some instances a certain band of frequencies is put up for auction by a regulatory body, to which interested parties can bid to own the specified spectrum. Due to the shift in how frequencies are allocated to network, neither the regulatory bodies nor the network operators care about the amount of different frequencies are used. Thus MO-FAP has lost its relevancy in the modern wireless industry.

3.5.2 Minimum Span FAP

The Minimum Span FAP (MS-FAP) is a problem that is very relevant today, especially when network operators want to deploy a new network in a region. The MS-FAP is concerned with keeping the interference below a certain level during assignment as well as minimizing the span. The interference threshold used, is specifed by the network designer as the minimum allowable interference on the network.

The span is defined as an interval on the frequency domain. This interval is calculated by taking the difference of the maximum and minimum frequnecies used during assignment. With the span value, network operators are able to request certain frequency bands and know their network will be able to operate at suitable interference levels [2, 67, 86].

The MS-FAP and MO-FAP are two very similar problems, the only difference is that MO-FAP focuses on minimizing different frequencies and MS-FAP forcuses on minimizing the interval of frequencies used during assignment [2]. The Philadelphia benchmark is usually used to gauge how good the algorithm performs.

3.5.3 Minimum Interference FAP

The Minimum Interference FAP (MI-FAP) or Fixed Spectrum FAP (FS-FAP) is typically encountered after the network operator has obtained a frequency band from a regulatory body. Other problems use matrices to forbid certain frequencies from being with certain transmitter [2, 23, 32, 67].

Unlike previous discussed problems, in MI-FAP any available frequency in the allocated band may be used even though it produces interference. The other problems are concerned with the frequencies used, even though they might be violating some constraints that incur a huge amount of interference. The interference value doesn't play a large role in their respective objective functions. In MI-FAP the objective is to minimize the total amount of interference on the network. It is important to note that this amount of interfernece might not necessarily be zero [2, 23, 32, 67].

The MI-FAP is the most encountered problem currently in cellular networks, since there are more operating networks than new networks being designed in the cellular industry today. This particular problem for form the focus for this dissertation.

Since MI-FAP is very close to real world instance problems, authors tend to use real world instances or benchmarks that resemble real world instance to test the quality and efficiency of their algorithms [2, 23, 32, 67]. We'll benchmark the quality and efficiency of our solution with the COST 259 benchmark which is discussed in section 3.7.

In this section we laid out the different types of Frequency Assignment Problems there are in the literature. We also gave a brief discussion on some of the literature found on the individual problems. Finally we formally stated on which one of the frequency assignment problems we will be concentrating on, namely the Fixed Spectrum Minimum Interference Frequency Assignment Problem (MI-FAP).

In the next section we will give a Mathematical definition for the Fixed Spectrum MI-FAP which will form the bases for the objective/cost function that we are going to minimize to find an optimal frequency plan.

3.6 Fixed Spectrum MI-FAP Mathematical Formulation

In this section we will give a Mathematical definition of the Frequency Assignment Problem which will form the core of what our algorithm discussed in this dissertation will optimize. We'll start of by denoting the symbols we will use and then we will give the Mathematical definition of the cost function we will minimize.

The Frequency Assignment Problem can be represented as a graph colouring problem hence it is known to be NP-Complete. Before we can formally define the Frequency Assignment Problem we first need to introduce some symbol definitions.

$$G = (V, E) \tag{3.1}$$
$$V = \{v_0, v_1, ..., v_i\} | i \in \mathbb{N} \tag{3.2}$$
$$E = \{v_0v_1, v_0v_2, ..., v_iv_j\} | v \in V, \forall ij \in \mathbb{N}, i \neq j \tag{3.3}$$
$$D = \{d_{01}, d_{02}, ..., d_{ij}\} | \forall \{i, j\} \in E, \exists d_{ij} \in \mathbb{N}^+ \tag{3.4}$$
$$P = \{\{p_{\bar{0}0}, p_{\bar{0}1}^{=}\}, \{p_{\bar{1}0}, p_{\bar{1}1}^{=}\}, \ldots, \{p_{\bar{i}0}, p_{\bar{i}1}^{=}\}\} | \forall \{i, j\} \in E, \exists p_{ij} \in \mathbb{N}^+ \tag{3.5}$$
$$F = \{0, 1, 2, 3, ..., k\} | \forall k \in \mathbb{N}, \forall v \in V \exists f \in F \tag{3.6}$$
$$d_{ij} < |f(i) - f(j)|, \forall ij \in \mathbb{N}, i \neq j \tag{3.7}$$

Let G (see 3.1) be a weighted undirected graph, where V (see 3.2) is a set of vertices. Each $v \in V(G)$ represents a transmitter in the frequency assignment problem.

E (see 3.3) is a set of edges. An edge consists of two vertices v_i and v_j that are joined because there exists a constraint on the frequencies that can be assigned between the two vertices or transmitters. Each edge has two associated labels d_{ij} and p_{ij} [11, 68].

The label d_{ij} that is part of the set D (see 3.4) denotes the maximum separation that is requried to exists between frequencies assigned to two transmitters v_i and v_j. Using $f(i)$ to denote the frequency assigned to i, we can determine using equation 3.7 if the interference involving the transmitters v_i and v_j is acceptable [11, 68]

The other label, p_{ij}, forms part of the set P (see 3.5) which is refered

to as the Interference Matrix[7]. Each label p_{ij} contains two values which represents interference[8]:

- $\bar{p_{i0}}$ represents the value for co-channel interference [11, 68].
- $\bar{\bar{p_{i1}}}$ represents the value for adjacent channel interference [11, 68].

Lastly we have the set F (see 3.6) that denotes a set of consecutive frequencies for every transmitter in V [11, 68].

Formally the Fixed Spectrum Frequency Assignment Problem (FS-FAP) can now be defined as a 5-tuple $FS-FAP = \{V, E, D, P, F\}$ with a required mapping of $f : V \rightarrow F$ [68]. The objective of the FS-FAP is to find an assignment of frequencies to transmitters that minimizes the sum of total interference (see 3.9).

$$c(p_i) = \begin{cases} \bar{p_{i0}} & \text{, if } |f(i) - f(j)| = 0 \\ \bar{\bar{p_{i1}}} & \text{, if } |f(i) - f(j)| \leqslant d_{ij} \\ 0 & \text{, if } |f(i) - f(j)| > d_{ij} \end{cases} \tag{3.8}$$

$$TotalInterference = \sum_{i=0}^{\mathbb{P}} c(p_i), p \in P \tag{3.9}$$

In this section we Mathematically defined the Frequency Assignment Problem using the symbols we defined. In the next section we will give a brief discussion on the different Frequency Assignment Benchmark Problems that exist and also define the benchmark we will be using in our implemention.

3.7 FAP Benchmarks

In this sections will discuss some of the most used benchmarks in the FAP domain. We will start of with the first benchmark that was introduced in the 70s and end of with a disuccsion on the benchmark we will be using to test our implementation.

[7] Discussed in section 3.3 page 25

[8] Interference values can be zero in some cases

3.7.1 Philedelphia Benchmark

The Philedelphia benchmarks are derived from an instance that was introduced in 1973 by Anderson. Each instance is a hexogonal grid of cells that overlaps the area of interest. At the center of each cellthere is a transmitter. Past approaches used these hexagonal systems to model modern cellular networks [2, 58].

In this benchmark interference is measured by a co-channel reuse distance. This distance stipulates that the difference of the frequencies assigned to two cells must greater of equal to a certain value d. A frequency cannot be assigned to a cell if it violates this minimum distance [2, 58].

These benchmarks are typically used to test algorithms developed for MS-FAP, since there is no concept of cost or penalty for interference incurred by violating constraints.

3.7.2 CELAR

In 1994 EUCLID introduced a project called CALMA which was a combined effort by various Europena goverments that were part of EUCLID to investigate algorithms for Military applications. The project was granted to six research groups. Within the project 36 instances were made available by CELAR for Radio Link Frequency Assignment [2, 21].

All the CELAR instances have the constraint that the difference between frequencies assigned to interfering radio links must be greater that a certain predefined distance in the frequency domain. This is a soft constraint and may be violated. Another constraint in the CELAR instances is that each pair of parallel links must differ by an exact predefined distance. This constraint is a hard constraint an may not be violated [21].

These instances were initially not available to the general public as it was contained to be within the CALMA project. In 2001 the CELAR launched an the International ROADEF challenge, were certain instances from the CALMA project were made available for the research teams taking part in the challenge. The instances made available had been modified to take polarizations and controlled relaxations of certain EMC constraints [36].

3.7.3 COST 256

The COST (COoperation europene dans le domaine de la recherche Scientifique et Technique) 259 is a set of real world GSM instances made available by die European Union. The instances are publicly available and can be downloaded for free at http://fap.zib.de/ (FAP Web 2007). The website also constains the most recent results obtained by researchers using these instances [2, 23].

The instances are fairly difficult due to the large amount of transmitters (900 - 4000) that need to be assigned frequencies, with a relatively small amount spectrum of frequencies. The main important characteristic of this benchmark is that is resembles real world GSM network data, which is why we the authors have selected this as the primary benchmark we will be concentrating on [2, 38].

More specifically we will concentrate on a small subset of the instances that are available, namely Siemens1, Siemens2, Siemens3 and Siemens4. In the paper by Montemanni and Smith [68] the same subset of problems is used and to date their algorithm has produced some of the best results.

3.8 Summary

Chapter 4

Heuristics Algorithms

4.1 Introduction

Meta-heuristics is a sub domain of the artificial intelligence domain. It evolved out of a need for more efficient search techniques with regard to hard problems.

Meta-heuristics forms part of a collective body of algorithms that use heuristics to search a particular domain's problem space, for the most optimal solution adhering to certain hard and soft constraints. Some of the most important Algorithms that form part of this collective body is:

- Tabu Search
- Simulated Annealing
- Genetic Algorithm

The above mentioned algorithms aren't the only algorithms to form part of this sub-domain, but they are the algorithms that have received the most attention in the literature and generally produce good results [62].

In this chapter our main focus will be to discuss each of the above listed algorithms. We will start of by briefly discussing the characteristics of meta-heuristic algorithms after which we will discuss each of the above algorithms in detail. We will also provide a literature study for each algorithm in order for us to see how an algorithm needs to be changed and optimised for a particular problem domain.

4.2 Characteristics of Meta-heuristics

NP-Complete problems have been proven to not be solvable in polynomial time by traditional search methods such as A* search, Breath First Search and Depth-First Search. Meta-heuristic algorithms on the other hand are much more efficient in searching the problem space and produce much better results in a short amount of time.

These algorithms are considered to be *general-purpose* algorithms and can thus be applied to a wide variety of optimization problems with only small modifications that need to made to the algorithm model [56].

Meta-heuristic Algorithms do not search statically by testing and evaluating every possible permutation in the solution space. Instead these algorithms make use of certain strategies and heuristics (specific to the problem domain) to search the solution space intelligently through trail and error [7].

These algorithms iteratively move through the solution space, using a heuristic to guide the search to move to more desirable regions in the solution space where there is a high probability of obtaining high quality candidate solutions [62, 68].

Meta-heuristic based search methods aren't guaranteed to find the most optimal solutions in the solution space, instead these methods are usually used to find near-optimal solutions. Thus most algorithmic development in the meta-heuristic domain focus developing new techniques that will increase the probability that a good solution will be obtained in difficult combinatorial problems [7].

Similarly, Meta-heuristics aren't guaranteed to find "good" solutions or perform well in each problem domain it is applied. The quality of the solution and performance of the meta-heuristic is very much depended on the expertise of the algorithm designer [113].

The standard meta-heuristic algorithms won't take advantage of specific domain knowledge to exploit the search domain and will produce relatively poor results. It is up to the algorithm designer to modify the algorithm sufficiently based on domain knowledge he/she as obtained [113].

Although heuristics play a key role in the performance of meta-heuristic algorithms, it isn't the only factor that has an impact on performance and results. Algorithms also use techniques and concepts from other system paradigms like multi-agent systems.

In multi-agent systems, multiple agents have to communicate with each other and the system as a whole has to perform some sort of autonomous self-organization. This social and self-organization concepts enable these systems to be distributed,robust and flexible. Which is why in meta-heuristic algorithms that are population-based,hybrid and/or distributed these same concepts are used to better exploit the solution space [61].

Meta-heuristics tend to slowly converge on an optimal solution, hence wasting valuable computing cycles. Therefore, a recent trend in research using meta-heuristics for problem solving often pair the algorithms with local search methods to increase the convergence rate of the algorithm to obtain a solution faster [35].

In this section we introduced the characteristics of meta-heuristics which sets these algorithms apart from the conventional algorithms used on difficult problems. We gave a general overview on how solutions are obtained as well as the quality of solutions. We also briefly discussed why for each problem domain the algorithm used, must be changed to fit the domain.

In the next section of this chapter we will discuss the Tabu Search meta-heuristic which we are investigating.

4.3 Tabu Search

4.3.1 Overview

Tabu Search (TS) was first proposed by Glover as a new searching technique to help algorithms avoid getting stuck in local optima present in combinatorial and optimization problems [82]. Since Glover introduced the algorithm in the 1980's, Tabu Search has been applied to a wide range of problems that include a wide variety of problems such as the Vehicle Routing Problem, Frequency assignment Problem, Capacitated-Lot Sizing Problem, Nurse Scheduling and the Resource Constrained Assignment Problem. Even though the problems mentioned differ by a large margin, the algorithm has been relatively successful in most of optimization problems it has been applied to. If we observe the results presented in the following research [13, 31, 65, 68, 74, 75, 82, 95, 98, 108] we can deduce that Tabu Search has on average obtained the best results compared to previous attempts with other algorithms.

Tabu search resembles in its most basic form the Hill-climbing search algorithm, but it differs in the sense that Tabu Search keeps a memory of its recent moves in the solution space [98].

General search algorithms like Hill-climbing, Random-restart or Scatter search tend to get stuck on local optima. The local optima might be a very attractive solution and thus general search algorithms will not move to better solutions since according the algorithms built in strategy it has found the best solution, but in actual fact its solution is the best it its *local* search space but not in the *global* search space. This is why an important characteristic that algorithms being applied on optimisation problems need to posses is breaking out of local optima.

In the next section we will explain what makes Tabu Search such a better algorithm that previous algorithms and why it is able on average to produce better results.

4.3.2 Important Tabu Search characteristics

In this section we will discuss some of the key characteristics and techniques that Tabu search exhibits that enables it to find relatively good solutions in a short amount of time. We will start of briefly discussing how the start solution Tabu Search iteratively improves upon. After initial solution generation we will give an overview of research done on neighbourhood strategies for TS. One of the most important features of TS will be discussed in the memory structures section. Finally, we will finish of this section with a discussion on the two search phases present in TS.

Initial Solution Generation

The core feature of the TS algorithm is sequentially improving the initial solution [117]. Thus an important consideration to make is how initial solutions are generated for the TS to start on. Random initial solutions might seem to be a good starting point, but by introducing randomization it becomes hard to control the quality of the end solution. Hence the generation of starting solutions must be controlled to limit the infeasibility of potential solutions [117].

Neighbourhood search

Tabu Search uses a neighbourhood local search process to explore the solution space. There is no set process of how neighbourhood candidate solutions are selected. Depending on the problem the TS is applied different neighbourhood solution selection strategies are needed. The overall quality of the solution produced by TS is also dependent on the neighbourhood search strategy used [117].

The TS algorithm isn't limited to just one neighbourhood search strategy. In the paper by Gopalakrishnan et al. [31] five neighbourhood move strategies are developed and are used interchangeably, in some cases a strategy is used three times in a row due to stagnation in the search space. However to combat this stagnation, the authors opted to use all the move strategies 15 percent of the time, and the last four moves strategies for 85 percent of the time when generating neighbourhood solutions.

Other neighbourhood strategies developed is one developed by N. A. Wassan [108]. In the authors paper a neighbourhood selection strategy is used that exchanges route nodes from initial vehicle routes for the Vehicle Routing Problem. This route exchange enables the TS algorithm to search much more broadly due to the constant supply of different solutions. Since initial solutions are constantly modified it enables the TS procedure to be a very fined grained process, because often a small changes in a potential solution can have a big impact on the overall proposed solution by the TS algorithm.

In the research done by Zhang et. al [117] an interesting neighbourhood selection scheme called *dynamic penalty* is discussed. When the algorithm moves onto an infeasible solution a penalty is imposed. By dynamically changing the penalty that is imposed the "feasibility" of solutions produced is influenced. Therefore, when and if the algorithm continually produces infeasible solutions, the penalty imposed is increased as to guide to algorithm to produce more feasible solutions. Finally, in the case when the algorithm is stuck on local optima, the penalty is reduced, which allows the algorithm to consider moving onto infeasible solutions thus escaping local optima.

Considering all the research done to develop new neighbourhood selection strategies that improve Tabu Search to search the solution space more efficiently and produce better faster solutions, Tabu Search still has some

draw-backs, especially with problems that have very large solution spaces [6].

Tabu Search is an iterative algorithm, executing a set of operation sequentially until a stopping criterion is met as can be seen in the flow-diagram presented earlier. At each iteration the algorithm has to determine feasibility of the immediate neighbourhood candidate solutions [6, 65]. Therefore each candidate must be evaluated by some function, which may be a costly operation in terms of computational cycles as well as in terms of time. Hence, this constant evaluation can drastically reduce the overall performance of the algorithm, since it spending more time calculating feasibility than actually searching the solution space [6, 65].

Memory structures of Tabu Search

The Hill-climbing and Random-restart algorithms are able to break out of local minima, but there is nothing stopping these algorithms from avoiding the local optima with their second or n-pass in the search space. Tabu Search differs from these algorithms by incorporating an important concept; the notion of memory.

In its most basic form Tabu Search keeps a local memory of all its recent best moves, and puts them into a *Tabu List* that has a predefined size. In the literature the Tabu list is also referred to as the *Tabu tenure* [31, 43, 108, 117]. The algorithm is not allowed to move to any solution that is in the Tabu list unless a solution that is *Tabu* is better than any current moves available in the immediate search neighbourhood [31,43,108,117]. The process of overriding a solutions Tabu status in the Tabu tenure is called the *aspiration criterion* [31, 43, 108, 117]. With the use of the Tabu tenure and the aspiration criterion, the algorithm is able to avoid cycling,local optima as well as searching in a to narrow region [5, 75].

Research done by Ashish Sureka and Peter R. Wurman makes an important distinction with regard to the memory scheme that is used in the TS algorithm. Two memory schemes are discussed; *explicit* memory and *attribute-based memory* [98, 99]. Between the two memory schemes the explicit memory scheme is the most used in the literature [65].

With explicit memory the algorithm stores a complete solution in the Tabu tenure, hence the algorithm is prohibited to move to that position in the solution for as long as the solution is in the Tabu tenure [98, 99]. With

attribute-based memory the algorithm stores the *operation* the is used to move from the previous solution, to the current solution [98, 99]. Therefore with attribute-based memory, the Tabu tenure intended function is changed from prohibiting certain solutions already encountered, to rather prohibit making changes to the current solution that would lead to solutions already present in the Tabu tenure [98, 99].

In research conducted by D.M. Jag and G.T. Parks and T. Kipouros and P.J. Clarkson, the authors add two additional memory structures called *Medium Term Memory* (MTM) and *Long Term Memory* (LTM) besides the standard Short Term Memory, typically referred to as the Tabu List [41] . Each additional structure remembers a different set of solutions for use by the diversification and intensification phases in the algorithm. These two phases will be discussed in the next section.

STM purpose is similar to the traditional Tabu list, to store the most recent solutions produced by the algorithm. MTM is designed to remember optimal or near optimal solutions. These solutions are therefore used later in the intensification phase. Finally, the LTM structure stores all the regions that the algorithm has already explored and is thus used in the diversification phase of the algorithm [41].

Search phases

As Tabu Search searches through the solution space, it goes through two cycles of search phases called *diversification* and *intensification* [13, 27, 35, 43].

The diversification phase in the TS algorithm, is the phase where the algorithm is directed to areas in the solution space which hasn't been explored yet. Diversification is usually applied by the algorithm as soon as mechanisms monitoring the memory, notice that solutions being produced are being repeated [27, 108].

In the literature diversification is achieved by new and innovative methods. A diversification strategy developed in research presented by Wassan [108] discusses a strategy called *escape diversification* where the algorithm is taken out of its current position in solution space as soon as solutions are being repeated.

In Research done by Fescioglu-Unver and Kokar [27] a strategy is presented that consists of two components namely the *Observer* and the *Diversifier*. The goal of the Observer is to continually monitor the best solution obtained by the algorithm whether it violates the *stagnation period*. The stagnation period is defined as the amount of iterations where the current best obtained solution hasn't changed [27].

As soon as the stagnation period is exceeded by the algorithm the Observer component activates and transfers the necessary information needed by the Diversifier component. The Diversifier component dynamically changes the size of the Tabu tenure based on the information the Observer gathered. The diversifier mainly targets older moves to diversify, but for short burst of time it would decrease the Tabu list size to a very small value in an attempt to combine new and old moves [27].

The specific mechanism used to define a new position where the algorithm can continue search, should ideally select areas in the solution space which have not been explored yet. Therefore, the diversification phase typically makes extensive use of the knowledge present in the long term memory structures as an indication to what areas of the solution space have been previously explored and which areas have not [13, 27, 35, 43].

Intensification is usually the first phase of the Tabu Search algorithm, since it is responsible to build up a history in memory for which the diversification phase can act upon. Fescioglu-Unver and Kokar also presented a intensification strategy based on control theory in their research [27]. The authors identified the repetition length as a critical value for their intensification strategy to be based upon. The repetition length is a control measure that defines how many times a solution may be repeated.

Repetition length was chosen because the authors observed through experimentation, that as solutions were repeated the algorithm was intensifying around a point in solution space. As the repetition length was increased that algorithm is forced to find more diverse solutions thus moving away from the intensification point [27].

In this sub section we discussed the different phases of the Tabu Search algorithm. We discussed the intended purpose of each phase and presented some relevant research done in this area.

In this section we presented the pseudo code for the Tabu Search algorithm as well as presented a Flow diagram depicted when and why certain

phases are activated in the algorithm. We then gave a overview of the core characteristics that are important for the Tabu Search algorithm. In the next section we will discuss Simulated annealing, where will start off by giving the pseudo code and flow diagram for the algorithm and end of the section by giving a overview on the core features of the algorithm.

4.3.3 Algorithm and Data flow

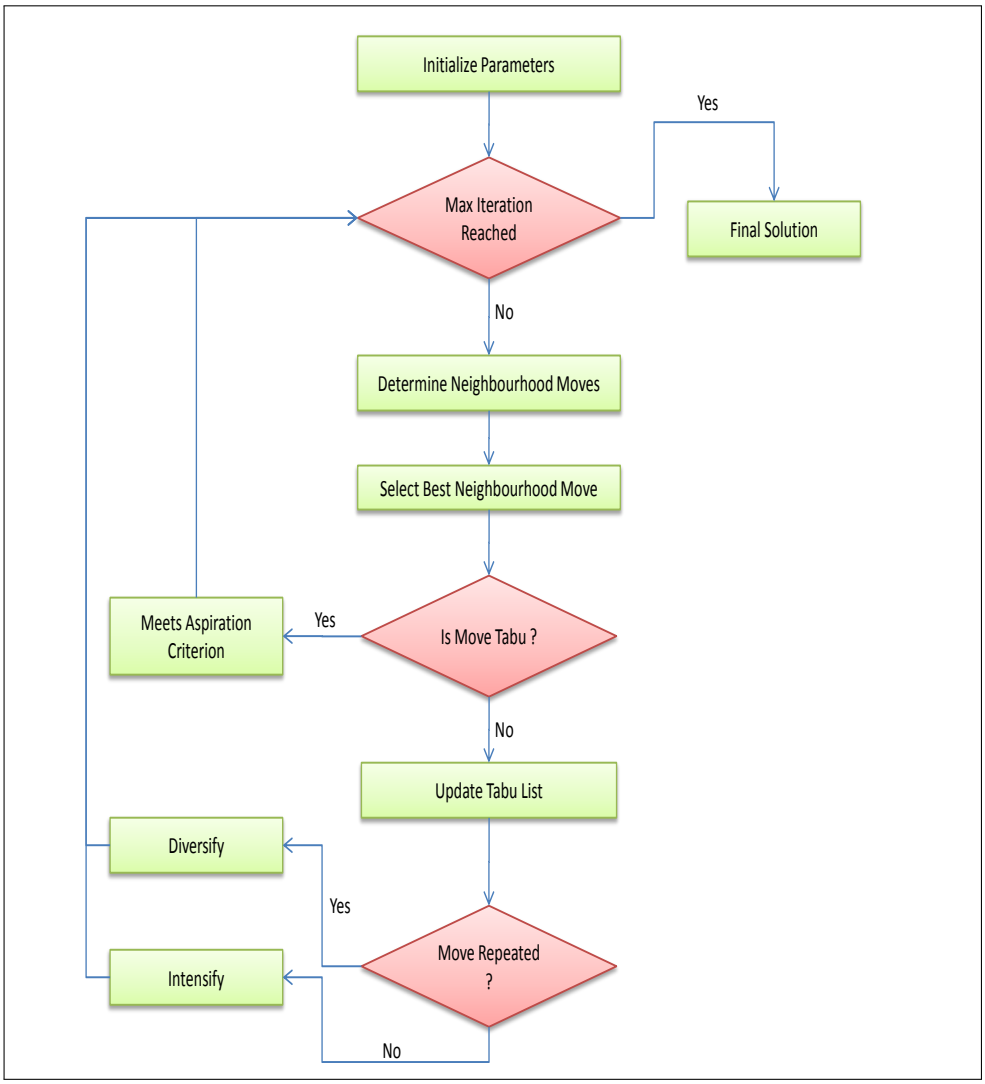

Figure 4.1: Flow Chart for Tabu Search Algorithm

4.4 Simulated Annealing

4.4.1 Overview

Simulated Annealing (SA) is a heuristic search technique proposed in the 80's by Kirkpatrick to solve combinatorial optimisation problems. The technique is based on a natural process which is known in metallurgical as Annealing [49, 73, 97, 105]. Kirkpatrick was the first to use the simulated annealing to solve optimisations problems but the basic algorithm structure was defined in by Metropolis et. al. in 1953 [72, 105]

Annealing is the natural process of crystallization when a solid is heated to a high temperature and then systematically cooled to a low temperature to reach a crystallized form [4,51,63,105]. This crystallized form of the solid is known the be the global minimum of the solids internal energy state. When the solid is rapidly cooled from a high temperature, the molecules have no time to reach a thermodynamic equilibrium stage [4, 51, 63, 105]. Therefore, the molecules of the solid have high energy and the resultant structure has no real crystalline form, thus the solid energy is at a local minima [51,63,105]. When the solid is slowly cooled in a controlled manner, the molecules are able to reach a thermal equilibrium at each temperature [4, 51, 63, 73, 105].

In the algorithm the energy state is the *cost function* that needs to be minimised, and the molecules are the *variables* which represent the solutions, and thus their state needs to be optimised to reach the desired energy state.

The following equation is the standard probability function that is used to determine when an uphill move performed by the algorithm. This function is known in the literature as the *Metropolis Criterion.*

$$M_{AC} = \begin{cases} 1, & \text{if } f(y) \leq f(x) \\ exp(-\frac{\Delta E}{T_k}), & \text{otherwise} \end{cases} \tag{4.1}$$

The function f is the objective function or a function that determines the state of a given position in solution space [112]. The parameter T_k is the temperature of the algorithm at iteration k [112]. Finally, ΔE is the change in "energy" between two solutions x and y [112].

The main purpose of the SA algorithm (like most optimization algorithms) is to minimize or maximise the cost function [97]. This cost function

typically evaluates a solutions desirability compared to other solutions in the immediate *neighbourhood* of the algorithms current position [17]. Typically a neighboring solution is only selected as the new best state if its desirability ranks hire than the current solution. When the algorithm moves to a better solution from the previous solution, the move is typically referred in the literature as a *downhill* move [105].

The best state isn't always selected, in some case the algorithm is also able to move to solutions which are worse than the current solution. A worse solution is only selected based on some probability which is controlled by the *annealing temperature* of the algorithm [17]. At a high annealing temperature the probability that the algorithm will select a bad solution is very good. As the annealing temperature decreases so does the probability that a bad solution will be selected [105]. When the algorithm moves to a worse solution, the move is typically in the literature referred to as a *uphill* move [105]. Uphill moves allows the algorithm to breakout of local minima and can lead the algorithm down a different path which may ultimately result in obtaining the global optimum [97].

The SA algorithm is also very popular due to the basic structure of the algorithm being generic [81]. Therefore, applying the algorithm to other problems is relatively trivial since only small changes are required. These changes usually need to applied to the *Neighborhood selection* scheme and the *Cooling Schedule* [81, 104]. Both of these concepts will be discussed in the next sub section.

In this subsection we gave a brief overview of the Simulated Annealing (SA) algorithm. We briefly discussed what the algorithm is based on and how the algorithm goes about searching the solution space. We also introduced some concepts like move probability selection and annealing temperature, which forms part of the core the algorithm. In the next section we will explain some of the concepts we've touched upon in this section as well as more advanced concepts that makes the algorithm unique.

4.4.2 Important Simulated Annealing characteristics

In this section we will provide five brief discussions on the characteristics of the Simulated Annealing Algorithm that we think make the algorithm unique. We will start of with a discussion on Markov Chains. An discussion

on one of the most important defining characteristic, namely the cooling schedule, will follow after the Markov Chain section. Furthermore we will discuss the importance of the initial temperature generation and what impact it has on the overall algorithm. Finally, we will end of this section with an overview on the efficiency of the algorithm.

Markov Chain

The SA Algorithm is typically modelled by using Markov chains due to each Markov chain represents a set of trails that the algorithm has executed at the same temperature. It has been proven with the use of Markov Chain theory that SA will find the global minimum in the solution space [57]. This proof is only valid when the following properties for the underlying Markov chain hold [72]:

- It must be irreducible
- It mustn't be periodic
- The detailed balance condition must hold

Due to the above proof the SA algorithm has been applied to a wide range of optimisation problems because as long as the algorithm designer can uphold these properties the find the global optimum. Even though the algorithm will find the global optimum, the algorithm is known to take a very long time to do so.

Cooling Schedule

The Cooling Schedule / Annealing Schedule is the most defining characteristic of the SA algorithm. It is the procedure where the natural annealing process is mimicked. The temperature of the SA algorithm is a control parameter that defines how much the algorithm moves around in the solution space.

In general, when the SA algorithm temperature has a very high value most solutions that are produced from the neighborhood are accepted [57]. Thus the algorithm moves freely in the solution space with little constraints. As the temperature decreases the probability that the algorithm will select bad or just any solution gets lower. When the temperature is very low, the

SA algorithm is similar to a greedy algorithm in a sense that it only accepts downhill movements [57].

In the literature there are three annealing schedules in common use are namely *the logarithmic schedule*, the *geometric schedule* and the *Cauchy schedule* [72, 97].

The standard and most common used schedule is commonly known as the logarithmic schedule and is based on Boltzmann annealing [72]. The main disadvantage of this schedule is that is slow due to its logarithmic nature [72]. It also requires that moves be generated from a Gaussian distribution for it to be able to reach the global minimum [97]. The logarithmic annealing function has the following form:

$$T_k = \frac{T_0}{ln(k)}, \text{where k is the iteration value} \tag{4.2}$$

The Cauchy schedule is faster than the logarithmic schedule. Similar to the logarithmic, this schedule also has a movement requirement. Moves must be generated from a Cauchy distribution for the algorithm to be able to reach the global minimum [72, 97]. The Cauchy schedule is also typically referred to as Fast annealing [72]. The schedule has the following form:

$$T_k = \frac{T_0}{k} \tag{4.3}$$

Finally, the fastest annealing schedule is known as the geometric or exponential annealing schedule [97]. This schedule introduces the concept of *re-annealing*. Re-annealing is a procedure by which all SA temperatures are rescaled [72]. This schedule has no move generation requirement to reach the global minimum, since there is no regious proof in the literature to prove it [97]. The geometric schedule has the following form:

$$T(i) = T_0 exp(-C_i), \text{where C is a constant} \tag{4.4}$$

Initial temperature

The initial temperature is a very important parameter to define in the SA algorithm, since it defines a point from which the cooling schedule will start of from. Therefore, depending on what the initial value of the temperature is the final result that the algorithm will produce can be influenced [81,93,110].

When the initial temperature is set to a very high value the algorithm takes a long time reach a result. On the other hand if the initial temperature is set to a very low temperature the algorithm might converge to quickly and thus produce a result which may be the local minima [81, 93, 110].

The initial temperature together with the cooling factor allows the algorithm designer to defines the time window for the algorithm to escape local minima, as well as the rate of convergence to an optimum solution [81, 110].

A low initial temperature together with a small cooling factor makes the time window for the algorithm to leave a local optimum very small [110]. With a high initial temperature and cooling factor value that is almost 1, the time window for the algorithm to leave the local optimum is much larger [110].

When the algorithm is near a global optimum, a low initial temperature and low cooling factor will allow the algorithm to reach the optimum faster in the solution space. In contrast, if a high temperature and a very low cooling factor is used the algorithm will take longer to reach the optimum even tough it is near the global optimum [110].

Move generation

Most of the research done on the SA algorithm focuses on the annealing schedule and not so much on the move/solution/neighbourhood generation. Typically an initial solution is generated and then small changes are made to the solution to represent a new solution. The solution is said the be perturbed to the next solution.

In research done by Tseung and Lin [105] a initial solution isn't modified but a move generation technique known as *Pattern* search is used. Pattern search has two forms of movement namely the exploratory move and the patter move. The exploratory move continually changes the certain variables of a solution [105]. This is done so that it can rapidly find and identify a "downhill" move. The Pattern move uses the information gathered by the exploratory move to move towards the minimum of the function [105].

Algorithm efficiency

The algorithm is also efficient with regards to CPU cycles when compared to the Genetic Algorithm because it only has to evaluate a certain number

of moves each iteration, instead of a evaluating a whole population each iteration. Unlike Tabu Search the basic SA algorithm does not keep any memory and is therefore memory efficient, but in contrast suffers the risk that solution will cycling. Which is hwy the number of iterations spent at a temperature, since the longer the algorithm spends at a temperature the higher the probability is that solutions will cycle.

4.4.3 Algorithm and Data flow

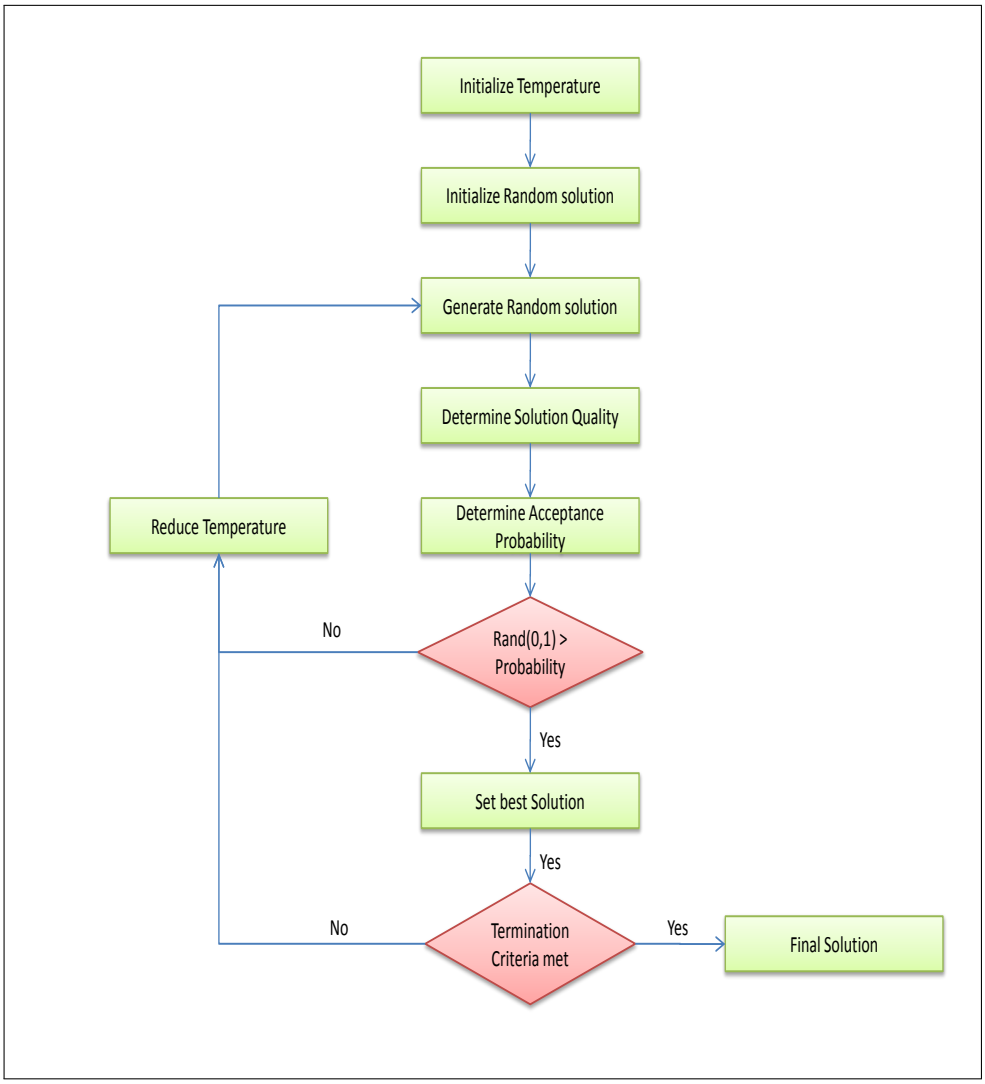

Figure 4.2: Flow Chart for Simulated Annealing Algorithm

Algorithm 1 Pseudo Code for the Simulated Annealing Algorithm

```
if i ≥ maxval then
  i ← 0
else
  if i + k ≤ maxval then
    i ← i + k
  end if
end if
```

4.5 Genetic Algorithm

4.5.1 Overview

Genetic Algorithm (GA) is a stochastic search method which is based on the natural process of genetic evolution and the Darwinian concept of "survival of the fittest" [29, 33, 48, 106]. GA was initially developed for adaptive systems by Holland but has then, been widely used in the optimization field of study due to its effective exploration of the solution space as well as its relative success in multi-dimensional problems [29,106,107]. The wide use of the GA algorithm can also be attributed to its generic algorithm structure as well as the ease of implementation of the algorithm [48, 106]. GA are application dependant and thus the designer needs to tailor the algorithm to his needs to obtain good results [33].

The GA search procedure involves searching the solution space through artificial evolution and natural selection [42, 91, 106]. An individual or point in the solution space is known as *chromosome* in the literature [14]. An initial set of chromosomes (referred to in the research as the *population*), are randomly generated [33, 42, 91, 106]. Unlike other algorithms, GA does not concentrate on one point when searching the solution space, but concentrates on a wide range of points represented by the population [29, 42, 106]. Each chromosome in the solution space represents a string encoding of the problem parameters [106]. Encoding problem parameters also attributes to the wide use of the GA since difficult mathematical problems can now be easily modelled [33].

The population is artificially evolved each iteration by using a set of stochastic operators [103]. This set consists of a selection operator, crossover operator and mutation operator [91, 103]. Each operator plays an impor-

tant role in emulating the evolutionary process. The selection operator is in charge of applying the *objective function* to each chromosome in the population [14,48]. Depending on if the selection operator is setup for maximization of minimization, each individual is ranked based on its "fitness" or objective function value. In accordance with the Darwinian theory, only the fittest individuals are selected from the population [14]. The fittest individuals are copied and sent to the the next phase of the algorithm which is known as the *Reproduction* phase [14].

Depending on how complicated the objective function is and how large the population is, the selection phase may be the most computationally expensive as well as time consuming [33]. The Reproduction phase mostly consists of string manipulations on the chromosome, to drive to search forward and it thus a phase which is completed quickly, especially if the string encoding is of a binary nature [33, 48]. These string manipulations occur through the application of the crossover and mutation operators [85]. The reproduction phase is where a new population is generated for the next generation to be evaluated by the selection operator. The reproduction is said to generate "offspring" from the selected fittest population who are in turn known as the "parents" of the offspring [14, 85]. Reproduction will occur until a certain number of predefined generations is reach or a suitable solution is found to be greater/less than a certain fitness threshold that would indicate a good chromosome [80].

There are two basic forms of the GA where both forms differ in the way that offspring and parents are handled [106]. The one form is called the *generational* GA and the other form is known as the *steady-state* GA [106, 109]. With the generational GA the offspring aren't immediately used in the next generation, instead they are kept in a pool until the pool reaches a certain size [106]. The offspring are then used to replace the parents entirely in the next generation [106]. In the steady-state GA the offspring are continually integrated with the population, thus offspring and parents occupy the same population pool every generation [106, 109].

The GA search process moves around in the search space using probabilistic rules rather than deterministic rules [106]. The probabilistic transition rules aid the algorithm to avoid local optima regions in the solution space [42].

Some sequential search algorithms require that the objective function be

differentiable [85]. These sequential algorithms use the derivative to obtain gradient information so that they can move in the solution space [85, 103]. The derivative of the objective function may be used in to increase the efficiency of the GA but is by no means a core requirement of the algorithm [42, 85, 103]. GA makes no assumptions about the solution space and primarily works on the information provided by the objective function [42, 85].

In this section we gave a overview of the Genetic Algorithm. We introduced the core concepts on which the algorithm is based upon as well as defined how the algorithm searches for solutions in a given domain. In the next section we will provide the pseudo code for the basic algorithmic structure as well as a flow diagram of the search procedure.

4.5.2 Important Genetic Algorithm characteristics

In this section we will give an overview on characteristics which make the GA search procedure unique. We will start of by discussing initial population generation, after which we will discuss the core operators that the GA uses. We will end of this section with a discussion on the efficiency of the GA.

Initial Population Generation

Initial population generation is the very first activity that the GA performs. Out of this population potential mating candidates are selected based on their fitness which indicates the desirability. Generally the initial population is generated by means of randomization [103]. Since the algorithm searches multiple points simultaneously in the solution space, it is desirable that the initial population have a wide diversity with regard to the solution they represent [29, 71]. By controlling the initial population generation we can control, to a small degree, the amount of exploration the algorithm does initially as well as avoid premature convergence [71]. Therefore, care must be taken in the selection of the particular randomization scheme that will be used to generate solutions.

In a survey done by Andrea Reese, two randomization schemes are defined namely pseudo-random number generators (PRNGs) and quasi-random number generators (QRNGs). PRNGs where found to be heavily problem dependant, improving the search efficiency in some instances and in other

instances having no considerable impact. QRNGs on the other hand, were shown to significantly improve the final solution produced by the GA as well as lowering the number of generations for the solution to be obtained [84].

Not all GA algorithm use randomization entirely for their initial population generation. In research done by Amit Nagar, Sunderesh S. Heragu and Jorge Haddock an algorithm is presented that generates an initial population through some aid of a branch-and-bound algorithm. The branch-and-bound algorithm provides the GA with an upper bound of acceptable solution in the solution space. The initial population is then randomly generated in the constrained space defined by the upper bound [71].

Selection Operator

The Selection operator is the first operator to be applied to the population after each generation. This operator is in charge of evaluating the current population to determine which individuals will survive and which will be "killed" off [71, 85, 88]. The individuals who survive and thus have the highest fitness are moved to a "mating pool" [14]. Individuals from this mating pool will be used in the the reproduction phase to generate a new population [33, 48].

By favouring high fitness individuals above low fitness individuals the operator guides the search towards better high quality solutions [85]. But care must be taken if the operator is to keen on high quality individuals since it may eliminate diversity in the population and thus result in premature convergence for certain problems [85]. If the solution space is known to have only one optimum, then a strict selection policy may be used, therefore directing the search into a gradient based direction [85]. In contrast, with a solution space that is known to have multiple optima, a forgiving selection policy might be more favourable since it allows the solution space to be more widely explored [85].

The most widely adopted selection scheme is known as the *Roulette Wheel* selection scheme [85, 88, 109, 116]. With this scheme an individual is selected based on a probability defined by the fitness of the individual divided by the collective fitness of the population [109].

Crossover Operator

The Crossover operator is usually the first operator applied to the population in the reproduction phase. The crossover operates exclusively only on the chromosomes that are in the mating pool. This operator is the main process by which by the GA algorithm is able to diversify as well as exploit certain optimal regions [71, 88]. Crossover works by interchanging and matching two parent chromosomes randomly selected from the mating pool to produce a single chromosome known as the offspring [14, 88, 106]. Since two chromosomes are combined or partially changed, some historical information is retained in the new chromosome [106].

In some algorithms like for instance the one presented in Nagar et. al., before the crossover operator is applied to the two parent chromosomes, the parents are first evaluated to determine if they represent suboptimal regions. If the either of the parents are from a suboptimal region, a disruption operator is applied that interchanges certain domain specific information between the parents. After the disruption operator is applied the crossover operator is applied [71].

There are a variety of ways with which values are interchanged between chromosomes in the crossover operation i.e, Fixed point crossover, Two Point Crossover, Uniform Crossover and Gaussian Crossover. Fixed point crossover operates on binary parents where by a point is selected in one parent and then all other bits are replaced by the other parents bits [14]. Two point crossover generates two random indices's which dictates a certain segment in the one parent to be interchanged with the other parent [85]. Uniform crossover is the most basic of all crossovers since it randomly selects bits from one parent to be replaced by another parents bits and is usually used when a large solution space must be search [107, 109]. Finally, the Gaussian crossover, interchanges bits between parents based on a Gaussian distribution [107, 109]. Depending on the state of the algorithm, crossover operators can also be interchanged or even paired if the algorithm needs better search performance for large or small solution spaces [3, 107].

Note, in all above crossovers it is assumed that the chromosomes are bit encoded, but these crossover do require them to be. All these crossover operators are able to work on any encoding, it just depends on what is considered to be a "bit" if a non-binary encoding is used.

Mutation Operator

The mutation operator is a probabilistic operator, which means it is applied infrequently and thus only with a certain probability will it be applied to parent solution. The operator typically changes some small in the solution regardless of the fitness of the chromosome [14, 116].

Given enough time, the mutation operator enables the algorithm to search the entire search space [106]. The operator also aids the algorithm with regard to escaping local optima in the solution space [106].

Due to the way the crossover works, some information may be lost when it is replaced by another chromosomes bits [33, 85]. Mutation is a source of new information which is continuously inserted into the algorithm, hence it works against information loss [33, 85, 88]. Usually, the mutation operator has no previous information on the chromosome it is mutating, thus it is entirely possible that the mutation may modify the chromosome for the worse [33]. A worse solution might lead the algorithm out of local optima or lead it down a new path to find the global optima, but this isn't always the case and thus in the literature the probability of the mutation operator is set to be very low [48, 85, 106].

In a survey done by Engelbrecht another mutation operator is dicussed. Instead of mutating a small part of a randomly selected chromosomes, this operator generates new offspring to be inserted back into the population. The operator randomly generates a new chromosome and then using any of the previous discussed crossover operators (see page 58) a

The mutation operator isn't always simple random operations. In research done by Il-kwon Jeong and Ju-jang Lee, a mutation operator is presented that incorporates the Simulated Annealing algorithm. Th SA mutation operator generates a new chromosome whose fitness is also calculated. If the new chromosome fitness is worse than the chromosome to be mutated, then depending on the SA mutation temperature as well cooling schedule , the newly generated chromosome might replace the chromosome to be mutated. Otherwise, if the new chromosome has a better fitness than the chromosome to be mutated, then it is replaced [48].

Elitism Operator

The elitist operator differs from the crossover and mutation operator in a sense that, it doesn't modify the chromosomes in any way. Instead, the operator works on the population [79]. The elitist operator ensures that the best chromosomes do not get lost from one population to the next, when the crossover and mutation operators are applied [3]. Thus the elitist operator is only applied after the crossover and mutation operators have generated a new population.

The elitist selects a certain number of high quality chromosomes from the parent population and transfers them in the new population without any modification from the other two operators [79]. The operator does not just move parents into the new population. The operator typically replaces sub par chromosomes in the new population with the higher quality parents [39]. Therefore, the elitist operator helps the algorithm retain knowledge gained from previous generations and prevents the best solution from extinction [77]. Finally, the retainment of knowledge and best solution aids the algorithm with regard to global convergence [94].

Algorithm Efficiency

The GA is a powerful, yet simple algorithm and tends to find good solutions given enough time. But the algorithm does has its disadvantages. One of the major disadvantages occurs when the GA is applied to problems which have very large solution spaces. In these problem, the population size is a very sensitive parameter. If the population is too small the algorithm won't have enough diversity to search and tends to premature converge.

A large population is preferred in large problem spaces, but then the algorithm is very computationally expensive since more time is spent evaluating than evolving new populations and the speed of the algorithm convergence decreases drastically. Hence, the population size must be fine tuned to achieve optimal performance in large problem spaces

Another disadvantage of GA, is that it is memory intensive, most sequential algorithms search on a single point bases through the solution space. As discussed above(see page 54), searches multiple points simultaneously and therefore requires more memory the keep track of all the possible solutions.

Finally, the algorithm has some sort of hill-climbing through the mutation operator, but the probability of the mutation operator is far too low for the algorithm to be considered to have a real hill-climbing capability.

4.5.3 Algorithm and Data flow

In this section we will give the start of by providing the pseudo code of the Genetic Algorithm. We will also provide a flow diagram depicting the general search procedure flow of the algorithm.

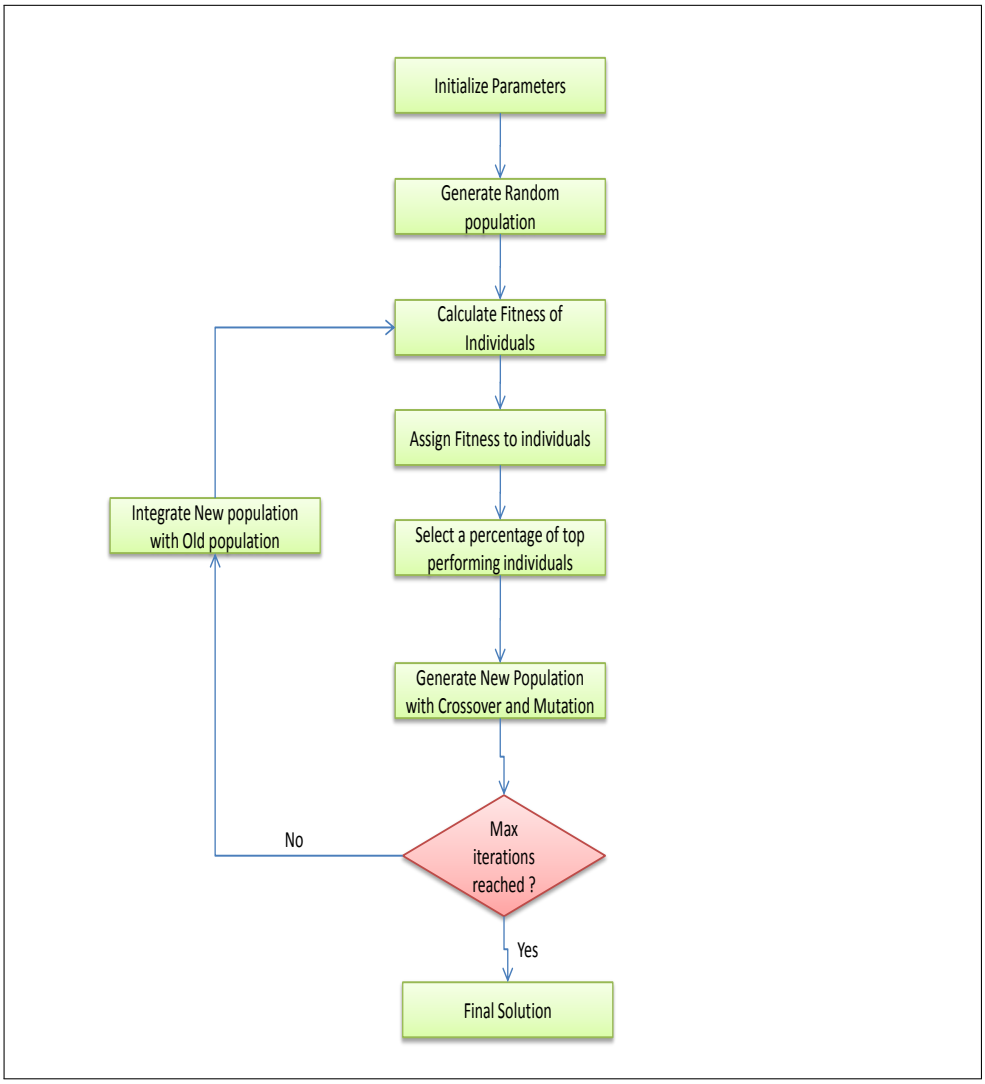

Figure 4.3: Flow Chart for Genetic Algorithm

4.6 Summary

Chapter 5

Swarm Intelligence

5.1 Introduction

We stand a lot to gain in Artificial Intelligence by studying the underlying inner workings of nature itself; this is why there is a branch of Artificial Intelligence which incorporates some of natures processes, like Genetics, which can be seen in action in the Genetic Algorithm.

There are other approaches in Artificial Intelligence which also have their routes in Nature, like Animal Learning /Dog Learning. These approaches only look at a single agent thought process when it maps percepts (senses) to actions in its respective environment [10].

Swarm Intelligence is an approach more concerned with the underlying processes and behavior patterns when multiple agents (insects, animals) come together and perform a task as one collective entity. This study of animals or insects in groups has already contributed to Artificial Intelligence as a whole [16, 53]. Each individual in the swarm might not be able to solve the problem on his own, but by interacting with other individuals and performing primitive actions, the individual can contribute to solving the problem as a whole entity [16].

New algorithms which incorporate the lessons learned from the study of the collective intelligence are now being used and form part of the Meta-heuristic Algorithm group. The initial algorithms developed with regard to Swarm Intelligence, where based on the coordination and behavior exhibited by schools of fish and flocks of birds. The newer generation of algorithms include:

- Ant Colony Optimization
- Artificial Bee Colony Optimization.
- Particle Swarm Optimization

These newer generations of algorithms are primarily being used in combinatorial NP-Hard problems, where there is no defined solution, only an optimization that comes close to a solution. Swarm Intelligence works on a key feature observed in nature, the notion of emergent behavior. Whenever an entity does something unconventional and gains success, it can be considered as exhibiting Emergent behavior as if it is emerging out of the masses way of doing things.

Typical emergent properties exhibited by Swarms are self-organization and synchronization [12]. In Swarm Intelligence, when an agent exhibits emergent behavior, the behavior of the agent propagates through the whole swarm. The behaviour prograte from one agent to another through social interaction which brings forth information exchange [12]. Therefore, the whole swarm adapts to a new way of doing things through information that was shared by one agent. The information can be anything that contributes to the swarm as a whole, like for instance the information might be about a new solution that is better than all previous solutions [12].

Social interaction is but one component of self-organization. Other components that form part of self-organization are [12]:

- positive and negative feedback
- increasing the magnitude of fluctuations

This is why Ant Colony Optimization, Particle Swarm Optimization and Artificial Bee Colony Optimization work so well in NP-Hard problems. The swarm always moves to a better state than previously observed.

NP-Hard optimization problems are only one of the fields of applicability for Swarm Intelligence. Other fields where Swarm Intelligence is being applied include neural network training, network routing, clustering [34] , search engines and load balancing []. Swarm Intelligence thus seems more suited towards problems with combinatorial complexity (B. Denby, 2003).

In this chapter we will first start off with a discusson on Ant Colony Optimization in section 5.2. Particle Swarm Optimization will be discussed

in section 5.4. Finally, we will end of this chapter in section 5.3 with a discussion on the Bee Colony Algorithm. Each section is divided into 3 sub-sections. First, we will give an overview of the algorithm, where we'll explain basic concepts about the algorithm and give an outline of the search process the algorithm uses. The second sub section will give an in depth discussion on some of the core characteristics that make the algorithm unique. We will finish of each section with a diagram depicting the general process flow and peseudo code of the algorithm.

5.2 Ant Colony Optimization (ACO)

5.2.1 Overview

Ant Colony Optimization is a class of algorithms incorporating different aspects that ants exhibit when they perform certain activities i.e, gather food, build nests and construct cemetries. The first ACO algorithms that were developed were based on the foraging behaviour that was exhibited by ants when finding the most optimal path towards a food source. The foraging behaviour was noticed by Deneubourg when he performed the Bridge experiment [20, 26].

Deneubourg seeked to know how ants where able to find the shortest path towards a food resource and communicate it to the whole nest [20]. Thus an experiment was setup up to study the ants behaviour. In the experiment a food source was placed a certain distance away from the nest. Two paths

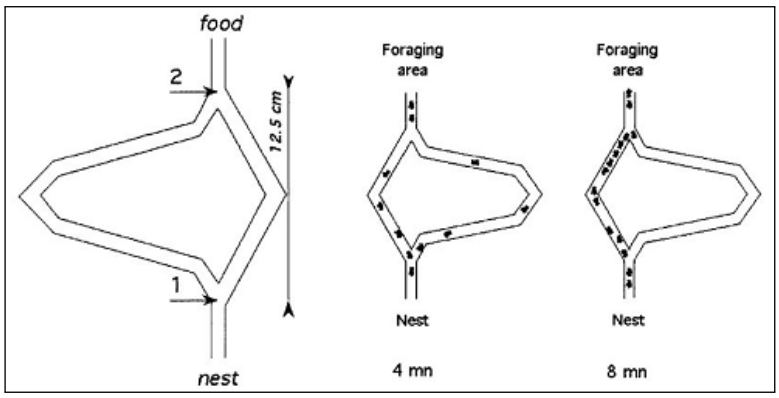

Figure 5.1: The Ant bridge experiment [20]

were setup towards the food resource, one path was purposely made longer than the other path as can bee seen in figure 5.1 [20].

The ants would then go out of the nest to gather food for their colony and would in order to do so need to take one of the provided paths toward the food souce. What J.L Deneubourg found was that the ants would initially oscillate between the bridges with no clear distinction of the more dominant route to take to retrieve food. But after a certain amount of time more and more ants would start preferring the path which is the shortest path towards food [20].

Naturally the questioned that was asked was how were the ants able to communicate to each other that the one food source was closer than the other? The answer lies in the use of *stigmergy* by ants. Stigmergy is defined as the method animals and insects use to facilitate communication through interaction. Interaction occurs through signals that the individuals receive which might require them to perform a specific action [9, 20, 26].

Two forms of stigmergy can be observed in nature. The one form , *Sematectonic stigmergy* , is a direct and physical form of interaction since it relies on altering the environment or by using some for of physical action to communicate [26]. Examples of this type of stigmergy include nest building and brood sorting [26]. The other form, *Sign-based stigmergy* is an indirect form of interaction, where communication occurs through some sort signal mechanism [26]. Ants use sign-based stigmergy when they are retrieving food. As the ant moves their path is reinforced with a chemical signal that alerts other ants to the desirebility of the path [26]. The chemical signal that ants use to indicate optimal paths is called *pheromones*. The concept of pheromones is a critical concept of ACO hence, we will we will provide an in depth discussion on pheromones in sub section 5.2.2.

The ACO algorithms are considered stochastic search procedures due to their inate use of randomness when exploring the solution space [18,114].The ACO class of algorithms has been applied to a wide range of problems that include — INSERT EXAMPLES. The algorithm does have some disadvantages. One of the primiary disadvantages of ACO is that ittends to get stuck on local minima in the solution space therefore, the solution space isn't adaquately explored and the algorithm prematurely converges to a local optima [114].

Improving the exploration of a population of agents in Swarm Intelligence

algorithms is no easy feat, especially for the Ant System which uses positive feedback(Gang Wang, 2005). To combat this shortcoming the MAX-MIN Ant Optimization algorithm was developed. The MAX-MIN algorithm is based on the original Ant system algorithm. MAX-MIN addresses the local minima problem by imposing dynamically changing bounds on the pheromones of the ants such that it is always with limit of the heuristic and best current path of the colony (Gang Wang, 2005).

The first algorithm to be based on the behaviour of ants was called the Ants System (AS) [9, 26]. The AS was initially applied to the Traveling Salesman problem (TSP) as a proof of concept application, but its performance was lackluster compared to other algorithms applied to the the same problem [9,26]. Subsequently, various algorithms have been developed that improve on the AS. These improved algorithms include the Ant Colony System (ACS), The ANTS system, Ant-Q and AntTabu [9, 26]. Where applicable we will refer to these algorithms to indicate how the have improved the AS system.

In this sub section we gave an overview of the ACO algorithm. We discussed how the algorithm was developed from observing ants. We also discussed how the algorithm utilizes the core communication conceptused my ants, called pheromones, is used to search the solution space. Finally we gave a small indication of of the typical problems at which ACO excells at as well as some of the disadvantages the algorithm has in its standard form.

In the next section we will give a more in depth discussion on the core characteristics of the ACO as well as additional characteristics that were developed in response to improve the algorithm efficiency.

5.2.2 ACO characteristics

In this section we will discuss characteristics which are important and unique to the ACO class of algorithm. We will discuss the following topics (in order of discussion): Pheromones, State Transition Rules and Pheromone evapuration.

Pheromones trail

The pheromone technique used by ants forms part of the core methodology used by the Ant Colony Optimization (ACO) algorithm [28]. As an ant

moves it lays down pheromones to mark the path it is walking. Pheromones decay over time therefore, as the ant follows the pheromone trail it reinforces the pheromone that is already on a path [28].Thus the more ants following a path the stronger the pheromone trail for that path and the stronger the pheromone the higher probability is that an ant will follow the path [28].

In the bridge experiment the ants started following the shorter path, because an ant following the shorter path would reinforce its pheromone trail faster than on the longer path. Initally, the ants oscillated between the two paths since there was no clear indication to the colony what the shortest path was, therefore the ants initially randomly selected the paths they would follow [28]. Once a path as been marked with pehromones the ant no longer selects a path based solely on randomness. Instead the ant selects a path based on a probability transition rule [20]. Transition rules will be discussed in the next sub section.

With the use of pheromones ants are able to communicate the best and shortest path. The more ants following a preferred path the more pheromones would be laid on that specific path and in return increase the strength of the pheromones [114]. The increase in strength of the pheromones on a path would thus let ants more clearly distinguish between paths it should and should not take [114]. Therefore, a pheromone provides positive feedback to the colony.

When the algorithm starts there are no pheromones to indicate path and the ants are placed at certain positions which are problem dependent. Initially, all the ants will choose random paths. After all the ants have completed their paths, each path is evaluated [26]. The amount of pheromone marking a path is in the standard ACO related to the cost function. Therefore, a low cost function value will have a high pheromone dosage indicating the path the ant took from each node and a high cost function value will have a low dosage [26]. The pheromone of each path hence, allows the colony to remember good and bad decisions from previous iterations [26]. This is a form of local pheromone updating which will be discussed in the pheromone update section. In the iterations following the initial one, the ants will at each node decide based on a probabilitly function whether it should follow the pheromone trail laid down in previous iterations or choose a new path to another node. Thus as the ACO iterates through more iterations the stronger particular path pheromone trail will become, hence signalling the

path out as the best found [26].

The pheromone trail was initially developed with only one colony in mind [26]. In the research done by Tiwari et al. pheromones in multiple colonies are considered. The basic principle of how pheromones are used by the ants stays the same, but the meaning of the pheromone changes if an ant of another colony encounters the pheromone trail. The ant will not follow or even consider the pheromone trail since any pheromone encountered from other colonies respulse the ant. Thus pheromones only provide positive feedback if the ant is from the same colony, otherwise the pheromone gives negative feedback, in a way warning the ant to stay away. This repulsion strategy promotes exploration among the multiple colonies [102].

In this section we gave an in depth discussion on what pheromones are as well as how they are applied in the most basic of cases. In the next sub section we will discuss the different transition rules that each algorithm in the class of ACO algorithms use.

State transition rules

As discussed in the previous sub section the ants select which path next to follow based on a probability. This probability is also known as the *transition probability*. In this sub section we will provide some of transition probabilities in use. Pheromone trails is the core concept upon which the Ant System is based and was the first algorithm developed based on ants that used the pheromones concept [26].

$$p_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}}{\sum_{u \in N_i^k(t)} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)}, & \text{if } j \in N_i^k(t) \\ 0, & \text{if } j \notin N_i^k(t) \end{cases} \tag{5.1}$$

The first transition probability is formualted in equation (5.1) and is used by individual ants of the AS algorithm [19, 26]. An ant k uses this equation to decide with what probability it will move from a node i to a node j [26]. τ_{ij} is the amount pheromone that exists between the path linking node i and j [20, 26]. Heuristic information is incorporated into the equation through the symbol η_{ij} which is the desirebility of the path from node i to node j as evaluated by an heuristic function [20, 26].

Through the use of parameters α to represent pheromone intensity and β to represent heuristic information the algorithm is able to achieve a good

balance between exploration and exploitation when $\alpha = \beta$ [26]. When $\alpha = 0$ no pheromone is taken into account, hence, any history that the algorithm has on the path between node i and node j is negelected and the algorithm degrades to a stochastic greedy search procedure. If $\beta = 0$ then the algorithm does not take into account the amount of desirebility the path between node i and node j as dictated by the problem specific heuristic function.

The set $j \in N_i^k(t)$ contains all the valid neighborhood moves ant k is allowed to make when moving from node i to node j. A tabu list is kept by each ant to trim the set of moves already performed previously, hence cycling is prevented.

Various other algorithms have been developed that improve on the basic AS. Each algorithm uses a different transition probability equation that might be very specific to the domain or a slight variant of the above equation as with the Ant Colony System. This section is not intended to give an exhaustive survey of different transition rules that exist in the literate. Therefore, we only discuss the first transition rule developed since most other rules can simply be considered deravites of the basic transition rule.

Pheromone update

As dicussed in the pheromone section, pheromones start to evapurate [1]over time hence, the path marked by the pheromone trail becomes less attrative to the ants. Therefore, a path that represents a good solution needs its pheromone trail to be continuously updated. In this sub section we will discuss the rules that govern when and by how much pheromones are updated.

Most of the variants that have been developed differ in the sense of what pheromone update rules they employ. In the literature pheromone update rules are classified into two groups [26]. The one group is called the global update rule. The other group is called the iteration based or local update rule of which an example has been given on page 68 [26].

The first local pheromone rule was presented in the AS algorithm [26]. The ants would retrace their path after each iteration, depositing pheromones on each link that makes the complete path. The following equation was used

[1]Pheromone evapuration is discussed on page 71

to update the pheromone:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t), \tag{5.2}$$
$$\text{where } \Delta\tau_{ij} = \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t)$$

Pheromone update rules that are in the global update group, only allow the pheromone trail of the path representing the best found solution by the algorithm since the first iteration, to be updated [26]. Thus the global rule favours intensification where the algorithm exploits the global knowledge gained by the ants to find a better solution. By updating we mean the pheromone is reinforced.

The Ant Colony System (ACS) was the first to use the global update rule together with the local update rule [26]. By using both types of rules the algorithm is able to efficiently exploit the history provided by the pheromones [26]. The global update rule used by the ACS is formulated in the following equation [26]:

$$\tau_{ij}(t+1) = (1-p_1)\tau_{ij}(t) + p_1\Delta\tau_{ij}(t), \tag{5.3}$$
$$\text{where } \Delta\tau_{ij} = \begin{cases} \frac{1}{f(x^{+}(t))} & \text{if } (i,j) \in x+(t) \\ 0 & \text{otherwise} \end{cases}$$

The parameter $x+(t)$ represents the best / shortest path so far found by the algorithm [26]. The rest of the parameters represents the same concepts as discussed on page 69

As can been seen in the following equation, the ACS uses a slight variant of the local update rule first used in AS [26].

$$\tau_{ij}(t) = (1-p_2)\tau_{ij} + p_2\tau_0 \tag{5.4}$$

In the above equation τ_0 is a small constant and $p_2 \in [0,1]$ is the constant that defines the rate of evapuration [26].

Pheromone evapuration

Initially when the pheromone concept was first implemeneted the ants of the colony rapidly converged on a solution and did not adaquately search the solution space for alternate path that might lead to beter solutions. To

combat this premature convergence and force the ants to explore the solution space more, the concept of *pheromone evapuration* was introduced [9,19,20, 26]. As discussed in the pheromone sub section (see 67) the pheromone marking a trail evapurates over time until it is reinforced by an ant. The evapuration of pheromones is governed by equation 5.5 [9, 19, 20, 26]:

$$\tau_{ij}(t) \leftarrow (1-p)\tau_{ij}(t), p \in [0,1] \tag{5.5}$$

The constant p defines the rate at which pheromone evapurates. If $p = 1$ the pheromone completely evapurates every iteration and the ants take no history into account with regard to there path selection and the search is completely random [20, 26]. Thus, the amount of exploration done by the algorithm can be controlled with the constant p [20, 26].

The above equation (5.5) was first introduced in the Ant System [9, 19, 20, 26]. Most subsequent algorithms that are form of the ACO class of algorithms also use the concept of pheromone evapuration, but they either use the standard equation or develop their own varaint [9, 19, 20, 26].

A more aggressive form of Pheromone evapuration is add to the Ant system discussed in the research done by [28]. The more aggressive form works beside the already present pheromone evapuration, instead this form seeks to add an additional search phase called *diversification*.

In the ant system developed by the authors a mechanism is added that monitors a certain number of solutionis that have been recently produced. If the solution hasn't changed a certain number of iterations the mechanism removes all pheromone trails currently in the system.

Therefore, the algorithm is forced to re-search the search space to create new solutions as it cannot rely on previous historic information provided by the pheromone trails. This mechanism forces the algoritm to diversify [28].

5.2.3 ACO Pseudo Code and Process Flow

5.3 Artificial Bee Colony Algorithm

5.3.1 Overview

The Artificial Bee Colony (ABC) algorithm is the youngest algoritm discussed in this chapter. It was first proposed by Karaboga in 2005 and

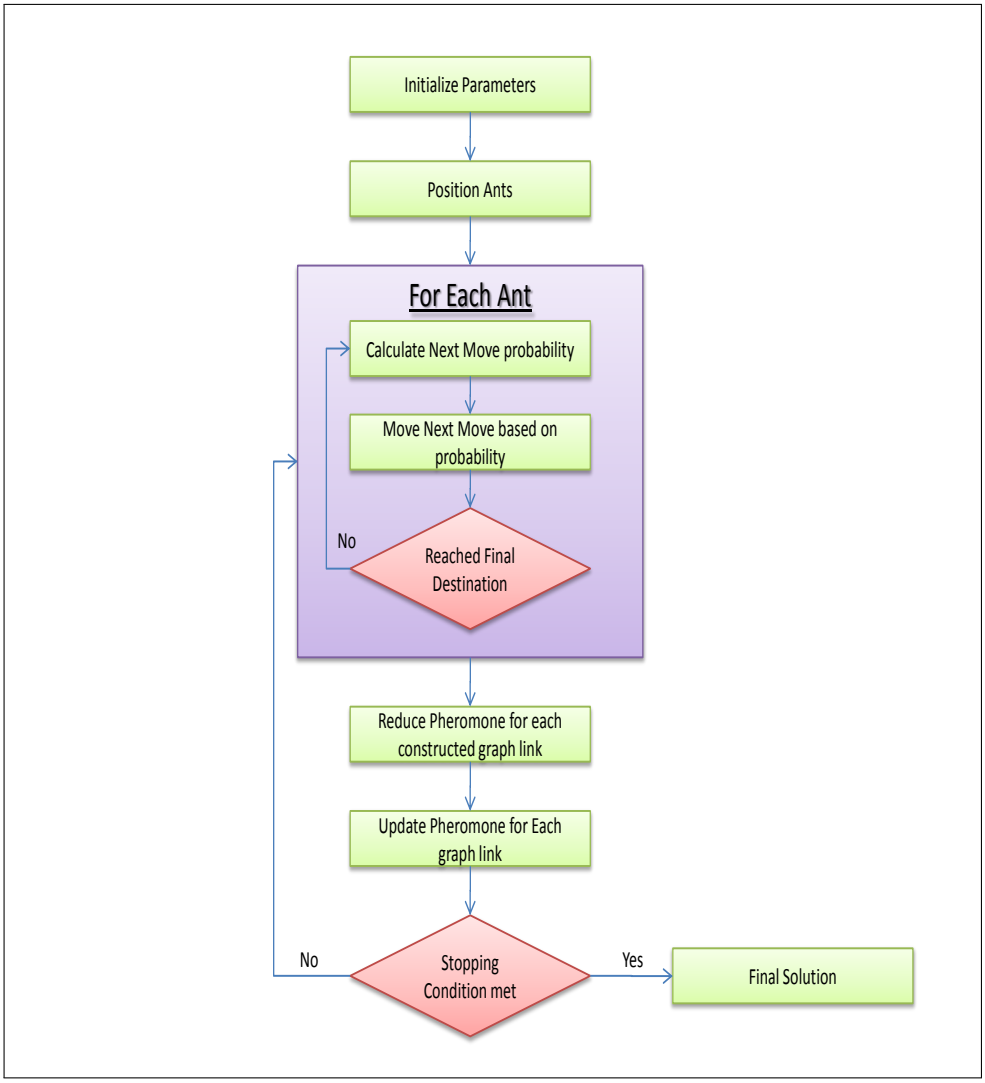

Figure 5.2: Flow Chart for ACO Algorithm

seeked to mimic the foraging behaviour exibited by bees [46, 47, 92]. Like ants, bees need to gather food to support the colony. To understand how the ABC algorithm tries to mimic the foraging behaviour of bees, we first need to explain how real bees forage [46].

In a bee colony there are a numerous number of bees, each with a specific role that performs certain actions for the colony. There are bees that protect the queen, bees that maintain the colony, bees that scout for resources and finally bees that gather food i.e the worker bees. The most important bees for foraging are those that scout and gather food [46].

The scout bees are sent out and as their role implies, they are responsible for exploring the surroundings of the hive to find suitable food sources [46]. If a food source has been found by scout bee, then it needs to return to the colony to share the information with the worker bees [46]. When the bee enters the colony it needs to communicate to the other bees by using some form of stigmergy (see page 66 [46]. The scout bee accomplishes this communication by performing a dance known as the "waggle dance" in the colony for all the bees to see [46]. This isn't a dance like in the traditional sense, since through certian movements the bee is able to commuicate a variety of characteristics about the food source that include [46]:

- How far the food source is from the colony.
- Quality of the food source.
- Path towards the food source.

Therefore, it can be concluded that with regard to foraging bees use *Sematectonic* stigmergy (see page 66) as the dance is a physical form of communication.

The dance is observed by "onlooker" worker bee [8, 46]s. These onlooker bees are currently "unemployed" in the colony [8, 46]. After the information of the scout be has been transferred the onlooker bees, becomes an "employed" bee since it is moving to the food source to start gather food [8, 46]. Thus it is the job of the worker bees to *exploit* the information provided by the *exploration* done by the scout bees [46, 47].

Worker bees gather food from the designated food source, until the food source reaches a certain quantatiy with regard to nectar content [46, 47]. Each time the bee returns to the colony it evaluates the current food source

versus other food sources discovered [46, 47]. Hence, if there is better food source found, the bee abandons the previous and starts gathering food from the new source [46, 47]. On the other hand, if the food source has been exhausted meaning there is no more nectar content to gather, the bee returns to the colony and becomes "unemployed" again [46, 47].

In the ABC algorithm, the food sources are the solutions [46, 47]. Each food source has an employed bee associated with it. Onlooker bees either wait for new food sources to be communicated to them or become employed bees by moving to another an attractive food source [46, 47].

As with real honey bees, a "waggle dance" is performed to all the onlooker bees by employed bees which provides information on the nectar amount (fitness value) that they represent [8, 46, 52]. The onlooker bees choose food sources depending on the nectar amount [8, 46, 52]. Therefore as the nectar amount of a food source increases the probability that the more onlooker bees will choose the source increases [8,46,52]. The probability will be discussed in the next sub section.

Bees can transition to different roles depending on their situation [46,47]. An onlooker bee becomes employed when assigned to a food sources and a employed bee can become a scout if his initial food source becomes exhausted [8,46,52]. Note, that not all employed bees of a food source become scouts, only the first employed bee of a food source transitions to a scout [8,46,52]. Scout bees are sent to randomly generated food sources [8, 46, 52].

The more onlooker bees a food source attracts, the more the neighborhood will get explored since the onlooker bees move to the food source and choose an immediate neighboring food source to be employed upon [46, 47]. Thus, this can be considered exploitation and the algorithm is therefore performing a local search. Finally, the numer of onlooker bees a food source has also indicates its desirebility, hence, a very good solution will have the majority of onlooker bees choosing it and search for nearby better food sources [46, 47, 52].

When a food source is abandoned, the previous bee that occupied the food source transitions to a scout bee [46, 47]. The scout bee is responsible for replacing the abandoned food source by find a new one, therefore, a new one is generated and communicated back to the colony [8, 46, 47]. The generation of food sources will be discussed in the next sub section.

Karaboga was not the first to base an algorithm on the above foraging

ehaviour. In the literature other bee foraging inspired algorithms have been developed such as the BeeHive Algorithm, Bee Colony Optimisation (BCO) and Bee Swarm Optimization (BSO) [47, 59, 100].

The BeeHive algorithm is based on the dance communication used inside the colony of bees. In BCO solutions are randomly generated and assigned to bees [47, 59]. The solutions are then progressively modified using certain strategies. Finally BSO solutions are iteratively constructed by forager (worker) bees and the best solution is communicated to the rest of the colony by performing a dance [47, 59]. All of the above mentioned algorithms were developed to be primarily used on combinatorial problems [46].

Another bee algorithm is the Virtual Bee Algorithm (VBA) which, like the previous algorithms and is also based on the foraging behaviour of bees, but it differs in the sense that it isn't designed for combinatorial problems [47]. Instead, VBA is designed for numerical function optimization [47]. In VBA bees would move around in the search space communicating to each other any target nectar food sources that were found [47].

Karaboga developed the ABC algorithm based on the previous research done on bee colony optimization and the above algorithms. The ABC algorithm is designed to be a multivairable optmization algorithm and has to date been applied to the Job Scheduling Problem, Clustering [59], Neural Network training and Reconfiguration of Distribution Networks [52]. Due to the nature of the algorithm being similar to that of the ACO, we can expect the ABC algorithm to be applied to a whole host of problems present in the literature.

In this sub sectino we gave a brief overview of the Artificial Bee Colony Algorithm. We discussed the real bee behaviour the algorithm tries to recreate and gave other algorithms that are also based on the same process. Finally, we defined the primary design goal of the algorithm and presented some of the problem on which the algorithm has been applied.

In the next sub section, we will discuss some of the core characteristics of the algorithm.

5.3.2 BEE algorithm characteristics

In this subsection we will discuss the characteristics of the ABC algorithm that define the algorithm and make it unique. We will start of by first

explaining how food sources are handled in the algorithm. We will then discuss how information is communicated to the colony. Finally, we will finish off this sub section with a more precise discussion on how foraging in modeled and used in the algorithm.

Food Sources

As discussed in the overview, food sources represent solutions to the problem the ABC algorithm is being applied to. When the algorithm starts, there are no defined food sources for the bees to evaluate and report on. Therefore, initially a finite amount of food sources are randomly generated. Since each food source needs an employed bee to evaluate the nectar amount of the source, the parameter that defines the amount of food sources also defines the amount of employed bees.

Employed bees evalute these food sources by determining their nectar amount. The nectar amount is directly related to the fitness value calculated using a domain specific cost function. After the amount if determined the employed bee advertises the food source to the colony by performing the waggle dance.

Onlooker bees bear witness to a number of dances from a variety of employed bees. The onlooker bees therefore need to select a food source that is the most attractive while maintaining some diversity in the pool of solutions. Thus, an onlooker bee selects a food source based on a probability function which is formulated in equation (5.6) [46]:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{5.6}$$

The parameter p_i is the ith food source under consideration by the onlooker bee. As discussed above, the fit_i parameter represents the value of the cost function and is directly related to the nectar amount of food source i. The parameter SN is the maximum amount of food source and hences the maximum employed and/or onlooker bees [46].

Employed and Onlooker Bees

As outlined in the overview, when recruited onlooker bees reach the advertised food source that is stored in memory, they do not occupy the same

food source. Instead, the bees explore the immediate neighborhood of the food source that communicated to them. The onlooker bees seek to find a food source that improves on the previous one. Equation (5.7) is used by the bees to generate new food sources in the neighborhood of a food source x_i.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5.7}$$

The subscripts $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are indexes which are randomly chosen. D represents the maximum dimensionality of the vector a solution represent. The index k has an constrainted tied to it – whatever value is randomly assigned to k it *must* differ from the value i. The position of the new food source in the neighborhood of x_{ij}, is controlled by the ϕ_{ij} parameter which is a bounded random value between $[-1, 1]$.

From equation (5.7 it can be concluded that the randomness of the food source position decreases as the difference between $x_{ij} - x_{kj}$ decreases. Thus, as the algorithm moves closer to an optimal solution the finer grained the search process becomes of the algorithm.

After a new solution v_i is produced, the bee takes the new solution and the old solution from memory to compare their respective nectar contents. If the new solution is found to have higher quality nectar, the bee replaces the old solution in memory with the new solution. Otherwise, the bee abandons the new solution and keeps the old solution in memory. Thus, the bee seeks to always move towards a better solution and therefore, uses a greedy selection process.

One of the problems with the above approach is that little information about the food source is used in generating a neighboring food source. In the research by Singh [92] a slight variation is proposed to generating food source neighbors by using more global information. The authors add a constraint to the algorithm that all neighboring solutions generated by *employed* bees must be unique. When an employed bee generates a neighbor and a identical solution already exists in the system, a *collision* is said to have occured. A collision is solved by letting the employed bee transion to a scout bee so that a completely random solution can be generated [92]. Scout and Onlooker bee generated solutions are not checked if it collides with other solutions in the system since the aim for them is exploration and not exploiting like with employed bees.

Scout bee

The artificial bees are modeled to based on the behaviour of real bees. Thus certain food sources can also be abandoned by an employed bee when it has outlived its usefullness. Abandonment of a food source can occur due to the following aspects:

- The employed bee has reached the maximum allowed cycles to improve the nectar amount. The maximum cycles spends on a food source allows the algorithm to avoid local optima.
- The solution represented by the food source cannot be improved any futher by the bee.

When a food source in the algorithm is abandoned, it needs to be replaced with a new food source. An employed bee under goes a role transtion when it abandons a food source. The bee transition from an employed bee to a scout bee.

It is the responsibility of the scout bee to replace the abandoned food source with a new randomly generated one. Scout be uses equation (5.8) to produce new food source that will replace food source x_i.

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j) \tag{5.8}$$

In research done by Gómez-Iglesias et. al [30] an extension is made to the scout bees. The Scout bee individuals are divided into two types of bees namely *Rovers* and *Cubs* bees [30].

- Rover bees are similar to traditional scout bees and hence use diversification strategies to explore the solution space.
- Cub bees explore the solution space relative to a good solution found by a Rover through randomly changing configuration parameters.

By using two different scout bees a good balance is achieved when searching the solution space in the beginning where diversity if preffered and late in the algorithm where intensification is preffered [30].

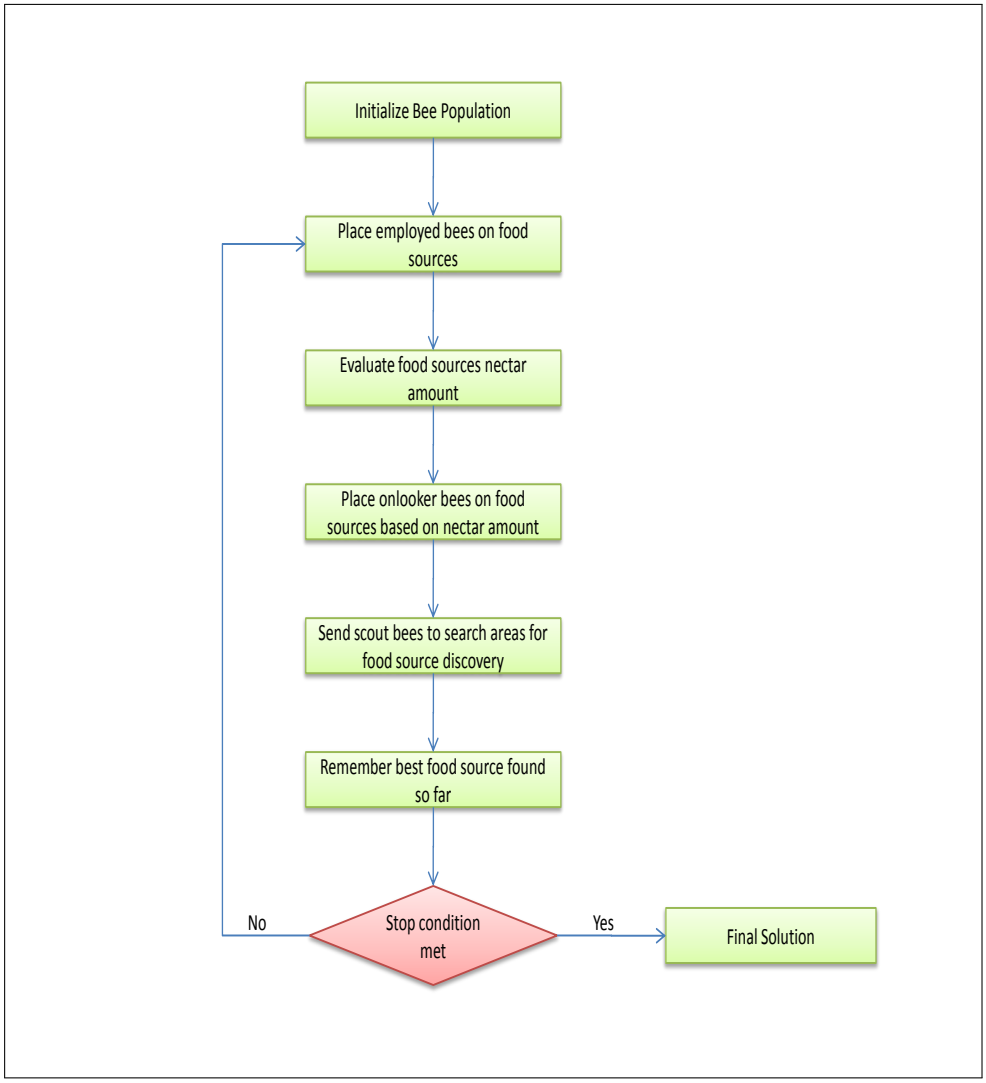

Figure 5.3: Flow Chart for the BEE Algorithm

5.3.3 BEE algorithm Pseudo Code and Process Flow

5.4 Particle Swarm Optimization (PSO)

5.4.1 Overview

Particle Swarm Optimization (PSO) is a self-adaptive, population-based stochastic search technique as was developed by Kennedy and Ebenhart in 1995 [90, 90]. The basic model of the algorithm is based on simulations done to recreate the natural behaviour of a flock of birds [111].

In the early stages of the particle swarm development, simulations were developed to closely model the stigmergy (see page 66) exhibited when a flock of birds cohesively move as one and is able to suddenly change direction in a unpredictable graceful manner only to regroup as one observed entity [44]. As the "leading" bird of the flock changes his movements the information is shared to the all birds that are in the immediaet neighborhood of the leading bird. As the informaiton is shared locally among the birds, each bird modifies his own movement to that of the leading birds movement.

Thus, because birds obtain information through observation of their neighboring birds, the stigmergy can be deemed of a physical nature. Therefore, the particular stigmergy used by birds is Sematectonic stigmergy (see page 66).

The simulations based on this behaviour of the flock, allowed researchers to discover the underlying patterns that governed the way birds are able to share information about the general movement of the flock. Based on these patterns and simulations the particle swarm alogrithm emerged into an optimization algorithm [26].

In the algorithm a particle is an individual. A group of particles, reffered to in the literature as swarm, is "flown" through the solution space of the problem the algorithm is being applied to. Each particle changes his movement based on information shared to him by neighboring particles in the swarm [25, 26]. As information is shared among particles, the success of one particle ripples through the rest of the swarm and each particle is able to utilize shared information that lead to success of another particle. Thus, each particle own personal experience and knowledge of the search space has an effect on its neighboring particles [25, 26].

There exists two primary PSO algorithms called Global PSO and Local PSO. The only difference between the two algorithms are how they go about sharing information to the rest of the swarm. The sharing models of these two algorithms along with other sharing models will be discussed in the PSO characteristics subsection [76].

In the swarm, each particle is a potential solution that is encoded in a D-dimensional vector [44, 83]. As a particle moves through the solution space, it continually evaluates current position and adjusts it accordingly to move in the general direction of the best particle of the swarm. Depending on the sharing model used, the best particle in the swarm is denoted as either *gbest* of the global PSO and *lbest* for the Local PSO [25, 26, 76].

The global PSO uses a star neighborhood for information sharing to allow for information to be shared by everyone in the swarm. In contrast, the Local PSO follows process natural birds more closely and subsequently uses the ring neighborhood for information sharing, hence, particles only share information with their immediate neighborhood and not with the whole swarm.

A particle evaluates current position by using a heuristic function or in more evolutionary algorithm terms, a fitness function. The fitness value indicates to the particle how far it is from an optimal position.

As a particle moves through the solution space it keeps a memory of the personal best position it has achieved since the start of the algorithm. In the literature and in the algorithm this personal best position is reffered to as *pbest* [76].

A particle moves with a certain velocity through the solution space. As the information is shared the particle must take advantage of the newly gained knowledge and therefore, needs to adjust its own velocity to match the swarm movement. The particle updates its own velocity to move in the direction of the *gbest* shared position, its own *pbest* positon and its current heading.

A particle needs to systematically explore the solution space. Hence, when the particle needs to update its personal velocity, it doesn't use "all" the information it has available to it otherwise it will start to cycle solutions. The particle uses a certain amount of global information together with a certain amount of local information to produce a direction and new velocity. The amount of global knowledge is reffered to in the literature as the *social*

component [25, 26, 76, 83]. The amount of information personal information used by a particle is referred to as the *cognitive* component in the literature [25, 26, 76, 83].

The PSO aglorithm is quick to converge on an optimum, which might not necessarily be the global optimum [83]. Hence, most of the research done on PSO has focused on the convergence of the algorithm as well as improving the diversity [25]. Most of these improvements and modifications will be discussed on the next sub section called PSO Characteristics.

In this section, gave an overview of the PSO algorithm. We started of by discussing how and why the algorithm was developed. We then discussed the what natural search procedure the algorithm incorporates and went on to explain how it is applied in the algorithm. After the search procedure was discussed, we starting giving a general outline of the a particle moves and how its velocity is updated.

In the next section, we will go into more depth about imporant PSO characteristics that we only touched upon in the overview, like Particle Velocity and Information sharing.

5.4.2 PSO characteristics

In this sub section, we will discuss some of the characteristics of the PSO algorithm. We will start off with a discussion on the *swarm size*. After the discussion on swarm size we will discuss the *particle velocity* where we will give the equation used to update particle positions as well as their velocities. A discussion on *Inertia* follows the particle velocity section. Finally we will end this sub section with a discussion on the different *information sharing* models of PSO.

Swarm Size

The swarm size dictates the search breath the algorithm has to maintain each iteration. The initializing of particles in a swarm are the same as traditional population-based evolutionary algorithm initilize their respective populations [115]. At start of the algorithm the swarm is initialized by randomly generating possible solutions that will represent the position of particles.

Even though the PSO algorithm in some aspects resembles evolutionary algorithms like the Genetic Algorithm, it does not continually generate new solutions to be re-inserted into the swarm to increase diversity [101]. Thus, the swarm size needs to be adjusted to get an optimal representation of the search space because as particles move in the swarm, the diversity among particles decrease rapidly as information is shared [25, 26].

A large swarm might increase diversity but at the expensive of computational time since the algorithm has to spend more time evaluating particles [25, 26]. Where as with a small swarm there might be low diversity but the algoritm requires little computational effort to evaluate and move the swarm around in the solution space [25, 26].

With a small swarm the algoritm will converge faster to a optimum, which is not gauranteed to be the global minimum [25, 26]. Therefore, care must be taken with the selection of the swarm size, because if a large swarm size is selected the algoritm will be slower to converge to an optimum but in turn, gains diversity which allows for a higher probability of finding the global optimum [69].

Particle Velocity

The velocity caluclation of each particle is where the optimization procedure occurs in the PSO algorithm and is the only means by which the PSO algorithm searches the solution space.

The velocity update is where the personal experience of a particle and the knowledge gained through social sharing are incorporated, to steer a particle into a certain direction by modifying its velocity. The most basic function to update a particle is formulated in equation (5.9).

$$v_i(t+1) = v_i(t) + c_1\phi_1(t)[pbest - x_i(t)] + c_2\phi_2(t)[gbest - x_i(t)] \quad (5.9)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5.10)$$

where $v_i(t+1)$ is the new velocity of a particle i for the next time step $t+1$. The cognitive component is represented by parameter c_1 and the social component is represented by the parameter c_2 (discussed on page 83). The current position of a particle in the solution space at time step t is represented by parameter $x_i(t)$. After the new velocity is calculated the position of the particle is updated for time step $t+1$ using equation

(5.10) [25, 26]. The velocity update can be visually depicted as shown in figure 5.4.

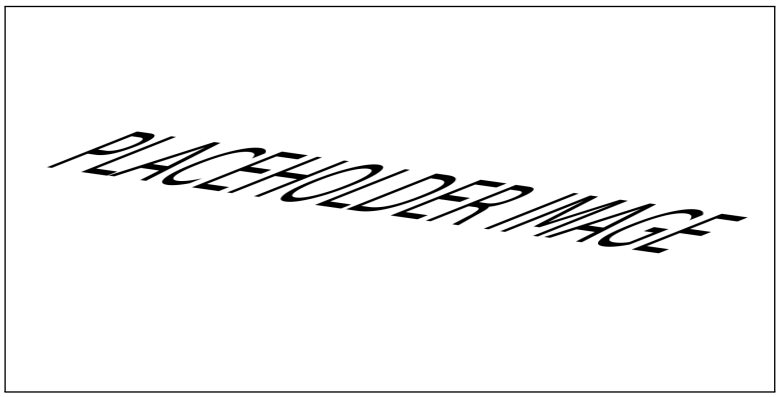

Figure 5.4: Visual particle velocity update [25, 26, 76, 83]

As discussed in the PSO algorithm overview pbest is the best position the particle has occupied since the start of the algorithm. In the literature there are two defined methods of determining gbest. The most common method used is where gbest is the best position obtained by a particle in the swarm since the start of the algorithm. Thus long term knowledge dictates the best position found which favours exploitation [25, 26].

With the second method of determining the gbest is the best particle position occupied by any particle in the swarm in the *current* iteration of the algorithm. Thus, short term knowledge dictates the best position found which favours exploration [25, 26].

As can be observed in equation (5.9) the new velocity is added to the old velocity. Therefore, the velocity of particle can get very large, especially for those particles that are far from the pbest and gbest positions. Large velocities are necessary for early exploration, but if the velocity gets too large, the rate at which the particle moves in the solution space is to high and good solutions might be missed [25]. Thus, the velocity of a particle needs to be clamped to ensure it step size stays within acceptable bounds. Equation (5.11) is used to clamp the velocity of a particle *before* its position

is updated [25].

$$v_i(t+1) = \begin{cases} v_i'(t+1), & \text{if } v_i'(t+1) < V_{max} \\ V_{max}, & \text{if } v_i'(t+1) \geq V_{max} \end{cases} \tag{5.11}$$

$$V_(max) = \delta(x_{max} - x_{min}) \tag{5.12}$$

where V_{max} is the maximum allowed velocity and $\delta \in (0, 1]$. The values x_{max} and x_{min} are the respective minimums and maximums of the domain the algorithm is being applied to [25]. The value of δ is very problem dependent and must be carefully chosen as the maximise the exploration - exploitation trade off [25].

Most of the literature has concentrated on the velocity of the particle as a focal point because it is the main function performing the optimization. In research done by Ratnaweera et. al. [83] a particle positions in the solutions space are continually monitored. If the particle appears to be stagnant in the solution space, the velocity is first updated, and then the particle is reinitialized with a random position. The new position of the particle is then updated with the new velocity. Thus knowledge of the previous discared particle is retained by using the velocity it had [83].

In research done by Kalivarapu et al. [45] an PSO algorithm is developed that seeks to incorporate the pheromone notion of ACO algorithms into the velocity updating of particles. The premise of the method is to allow greater sharing of information about promising areas between particles. The algorithm developed by the authors achieved promising results with the algorithm in certain cases finding solutions faster and better solutions than other PSO algorithms [45].

Other research done by Monson and Seppi [66] is more concerned with how particle presented. In the general PSO algorithm, particles have no physical form or volume, thus particles in the swarm move through each other. The authors changed this in their algorithm by letting each particle have a radius around itself. Therefore, as particles move through the solution space and if another particle at a certain time step occupies the same space, the particles are said to collide. As one would expect, when a collision occurs both particles are deflected into random directions [66]. At a greater expense of computational time due to constant collision detection, the PSO gains greater exploration in the solution space.

Finally, in the research by Lenin and Monan a PSO algorithm is developed that is called the Attract and Repulse PSO (ARPSO). The algorithm continually monitors the solutions in the swarm. If the algorithm picks up that a certain percentage of the swarm is stagnating, the algorithm activates the respulse state. In the repulse state particle are repelled from other particles in the swarm, this facilates greater exploration. After a certain number of iterations, the algorithm returns to its default state, where particles attract each other. The attract state facilates exploitation [50].

Inertia Weight

As a object moves with a certain velocity it carries momentum, if the object were to suddenly change direction, momentum would for a certain period still move the article in the previous direction. Inertia wieght seeks to add type of behaviour to the particles of the PSO algorithm. It was initially developed to negate the use of clamping a particle velocity. The velocity update equation with added inertia is formulated in equation (5.13) [25].

$$v_i(t+1) = wv_i(t) + c_1\phi_1(t)[pbest - x_i(t)] + c_2\phi_2(t)[gbest - x_i(t)] \quad (5.13)$$

The addition of inertia (w in equation (**??**) to the general velocity update equation is simple and ellegant which is why it has been adapted to a wide variety of PSO algorithms. The inertia value allows one to control the amount of control the social and cognitive components have with regard to velocity updates [25].

For values of w that are greater than 1, a large amount of momentum is preserved as the particles velocity are updated and hence, the particle explores more [25]. When $w < 1$, each time the particle updates it velocity is loses a certain amount of momemtum, hence, the particle seems to slow down allowing it to exploit the current solution space in finer detail [25].

Even though inertia was developed to remove the use of velocity clamping, it did not entirely achieve its goal. For values of w that are greater than 1, the particle keeps a lot of momemtum and accelerates even more. Therefore, as the particle accelerates its step size through the solution space gets larger and larger, increasing the probabilitly that it will miss a good solution. This disadvantage is only applicable for algorithms that keep the inertia value static [25, 26].

To allow for a greater trade-off between exploration and exploitation, the inertia value was made dynamic. Exploration is favoured early on in a optimization algorithm and in return exploitation is favoured later on the algorithm when it is near a optimum. Thus, various methods linear decreasing and nonlinear decreasing and have been developed that modify the inertia component as the algorithm moves around in the solution space [25, 26].

Finally, a similar inertia type component was developed from the ananlys of particle dynamics [25]. This new component is called the *constriction coeffcient* and like the inertia above, also modifies the velocity update equation slighty as can be seen in equation (5.14)is formulated in equation (5.15) [25, 26, 66].

$$v_i(t+1) = \chi[v_i(t) + c_1\phi_1(t)[pbest - x_i(t)] + c_2\phi_2(t)[gbest - x_i(t)]] \quad (5.14)$$

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (5.15)$$

The constriction coeffcient is represented by the value *phi* and allows one to omit the usage of velocity clamping. The constriction coeffcient evaluates to a ever decreasing value between $[0, 1]$. By using the constriction coeffcient the PSO algorithm is also gauranteed to converge for values of $phi \geq 4$ and $\kappa \in [0, 1]$. As with the inertia discussed above, high values of κ allows for greater exploration and slow convergeance. Where as low values of κ forces the algorithm to exploit the solution space and converge quickly [25, 26, 66].

5.4.3 PSO Pseudo Code and Process Flow

5.5 Summary

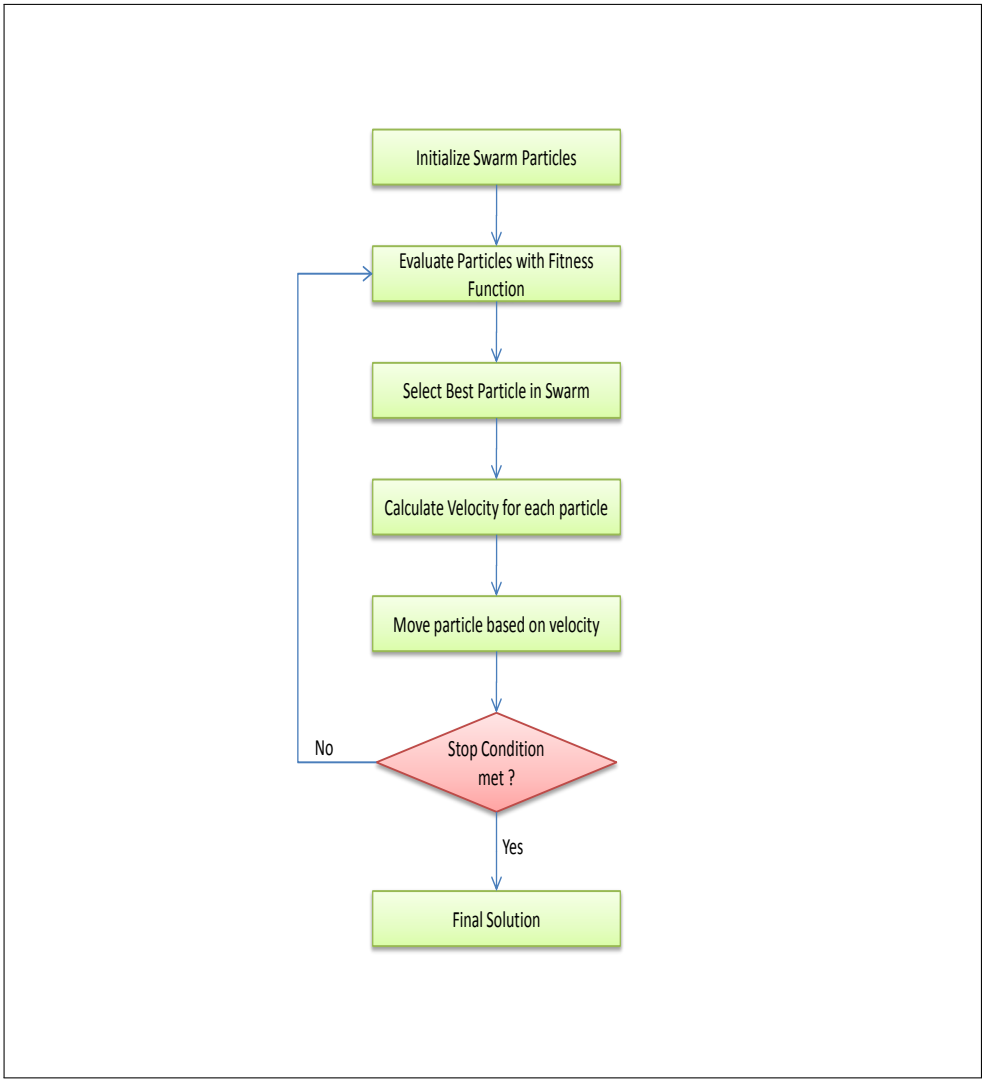

Figure 5.5: Flow Chart for PSO Algorithm

Part II

Implementation

Chapter 6

PSO on benchmark functions

6.1 Introduction

In this section we will give the formulation of all the benchmark functions we have implemented for the developed PSO to optimize. The functions vary from being relatively easy to optimize, to functions that contain lots of local minima and a slightly concealed global minmima. In total we will formulate 14 benchmark functions.

The reason why these functions have to variate amount of local and global minima, it to test various factors on how good the algorithm is that is being applied on the function. The factors that are tested are:

- Rate of convergeance
- Exploration
- Exploitation
- Diversity
- Breaking out of local minima
- Information sharing

6.1.1 Test Functions

DeJong F1 Function

$$f(x) = \sum_{i=1}^{n} x_i^2, -5.12 \leq x_i \geq 5.12, i \in \mathbb{N} \tag{6.1}$$

The DeJong F1 Function has the following global minium when $f(x) = 0, x(i) = 0, i : n$ where n is the amount of dimensions.

Shekel's Foxhole

$$f(x_1, x_2) = \{0.002 + \sum_{j=1}^{25}[j + (x_1 - a_{1j})^6 + (x_2 - a_{2j})^6]^{-1}\}^{-1} \tag{6.2}$$

where

$$a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & ... & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & ... & 32 & 32 & 32 \end{pmatrix}$$

the variables x_1 and x_2 are usually restricted to the square represented by $-65.356 \leq x_1 \leq 65.357, -65.357 \leq x_2 \leq 65.356$. The global optimum is when $f(x_1, x_2) = 0, \{x_1, x_2\} = \{-32, -32\}$

Rastrigin

$$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)],\ i \in \mathbb{N} \tag{6.3}$$

The values of the variable x_i is bounded by the hypercube $-5.12 \leq x_i \leq 5.12$. The global optimum for the function is when $f(x_i) = 0,\ x_i = 0,\ i = 1, \ldots, n$

Schwefel

$$f(x) = 418.9829n - \sum_{i=1}^{n}[x_i \sin\sqrt{|x_i|}], \qquad i \in \mathbb{N} \tag{6.4}$$

The variable x_i is restricted to be in the hybercube $-500 \leq x_i \leq 500, i = 1, \ldots, n$. The global optimum for the function is $f(x) = 0$ when $x_i = 420.9687$.

Griewank

$$f(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \qquad i \in \mathbb{N} \tag{6.5}$$

The variable x_i is bounded to within the hypercube $-600 \leq x_i \leq 600$. The global optimum of the function is when $f(x) = 0$ which occurs when $x_i = 0, i = 1, \ldots, n$

Salomon

$$f(x) = -\cos\left(2\pi \sum_{i=1}^{n} \sqrt{x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2} + 1, \quad i \in \mathbb{N} \tag{6.6}$$

Unlike the previous functions discussed in this section, the Salomon function imposes no constraint on the x_i variable. The global optimum is when $f(x) = 0$ and $x_i = 0$ where $i = 1, \ldots, n$

Ackley

$$f(x) = -20e^{-0.2\sqrt{\frac{1}{2}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{2}\sum_{i=1}^{n} \cos 2\pi x_i} + 20 + e^1, \qquad i \in \mathbb{N} \tag{6.7}$$

The variable x_i is restricted to the hypercybe represented by $-32.768 \leq x_i \leq 32.768$. The global minimum is when $f(x) = 0$ and is obtainable for $x_i = 0, i = 1, \ldots, n$.

Six-Hump Camel Back

$$f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^{\frac{4}{3}})x_1^2 + (x_1 x_2) + (-4 + 4x_2^2)x_2^2 \tag{6.8}$$

The variables x_1 and x_2 are subject to the following boundary constraints $-3 \leq x_1 \leq 3$ and $-2 \leq x_2 \leq 2$. The global minimum is when $f(x_1, x_2) = -1.0316$ and is obtained when $x_1 = -0.0898$ and $x_2 = 0.7126$

Shubert

$$f(x_1, x_2) = -\sum_{i=1}^{5}(i\cos(i+1)x_1 + 1)\sum_{i=1}^{5}(i\cos(i+1)x_2 + 1) \tag{6.9}$$

The search domain constrained to $-10 \leq x_i \leq 10, i = 1, 2, \ldots, n$. The global optimum which is when $f(x_i) = -186.7309$.

Himmelblau

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{6.10}$$

The variables x_1, x_2 are constraint to be within the hypercube represented by $-6 \leq x_1 \leq 6, -6 \leq x_2 \leq 6$. The Himmelblau function contains no local optima, but on the contrary, it has 4 global optima when $f(x_i) = 0$ which can be obtained at the following points $(x_1, x_2) \in \{(-3.779310, -3.283185), (-2.805118, 3.131312), (3, 2), (3.584$

Rosenbrock Valley

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \tag{6.11}$$

The variable x_i is bounded to with the following constraint $-2.048 \leq x_i \leq 2.048$. The global optimum is when $f(x) = 0$ and is obtained when $x_i = 1, i = 1, \ldots, n$.

Dropwave

$$f(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{\frac{1}{2}(x_1^2 + x_2^2) + 2} \tag{6.12}$$

The variables x_1 and x_2 are restricted to be within the following bounds $-5.12 \leq x_i \leq 5.12$ The variables x_1 and x_2 are restricted to be within the following bounds $-5.12 \leq x_i \leq 5.12$

Easom

$$f(x_1, x_2) = -\cos(x_1)\cos(x_2)e^{(-(x_1-\pi)^2-(x_2-\pi)^2)} \tag{6.13}$$

The variables x_1 and x_2 are restricted to be within the hypercube represented by $-100 \leq x_1 \leq 100, -100 \leq x_2 \leq 100$. The global minimum is when $f(x_1, x_2) = -1$ and is obtainable if $(x_1, x_2) = (\pi, \pi)$

Branins

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos x_1 + e \tag{6.14}$$

where

$$\begin{aligned} a &= 1 \\ b &= \frac{5.1}{4\pi^2} \\ c &= \frac{5}{\pi} \\ d &= 6 \\ e &= 10 \\ f &= \frac{1}{8\pi} \end{aligned}$$

The variables x_1 and x_2 are subject to the following boundary constraints $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 10$. The global optimum is when $f(x_1, x_2) = 0.397887$ and is obtainable when x_1 and x_2 have the following values:

1. $x_1 = -\pi,\ x_2 = 12.275$
2. $x_1 = \pi,\ x_2 = 2.275$
3. $x_1 = 9.42478,\ x_2 = 2.475$

Michalewicz

$$f(x) = -\sum_{i=1}^{n} \sin(x_i)[\sin(\frac{(1-x_i^2)}{\pi})]^{2m}, \qquad i, m \in \mathbb{N} \tag{6.15}$$

The variable x_i is usually constricted to the following defined boundary $0 \leq x_i \leq \pi, i = 1, \ldots, n$. The parameter m defines the steepness of the valleys in the function. The function has two approximated global minima's

1. $f(x) = -4.687,\ n = 5$
2. $f(x) = -9.66,\ n = 10$

Goldstein

$$\begin{aligned} f(x_1, x_2) &= (1 + (x_1 + x_2 + 1)^2) \\ &= *(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \\ &= *(30 + (2x_1 - 3x_2)^2 \\ &= *(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2) \end{aligned}$$

The variables x_1 and x_2 are subject to the following boundary constraints $-2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2$. The function has only global minimum when $f(x_1, x_2) = 3$ and is obtainable when $x_1 = 0$ and $x_2 = -1$

6.2 Graphs of Benchmark Functions

In this section we will present 3D surface and contour graphs for all the mathematical functions discussed in the previous section. These graphs allows one to see the search landscape more visually as well as to comprehend how difficult it is to find the global minima for a particular function.

6.2.1 Graphs

DeJong's First Function

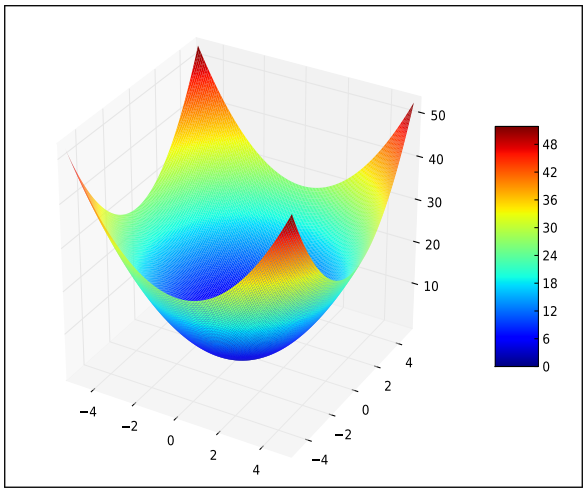

Figure 6.1: DeJong's First Function

Shekel's Foxhole Function

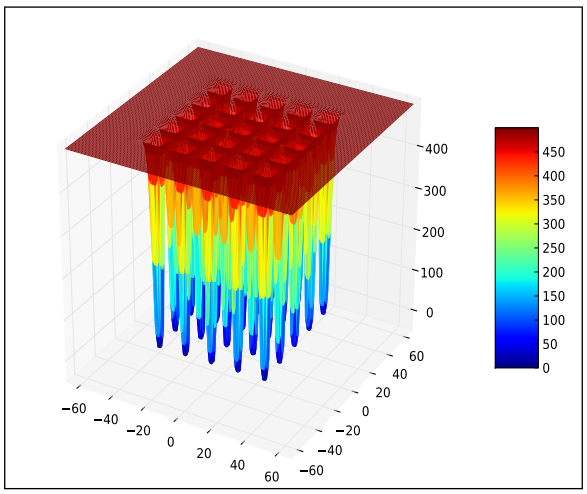

Figure 6.2: Shekel's Foxhole Function

Rastrigin Function

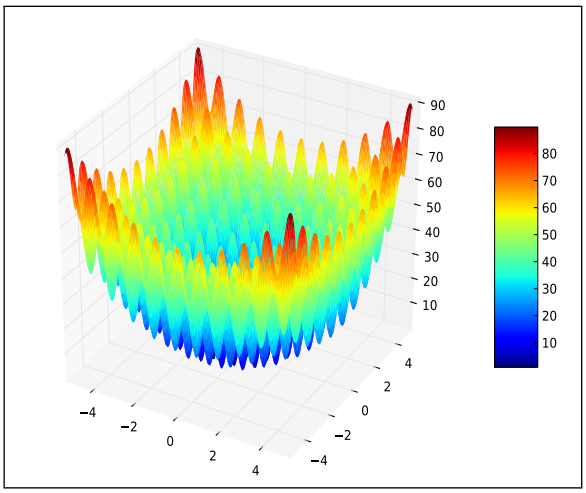

Figure 6.3: The Function

Schwefel Function

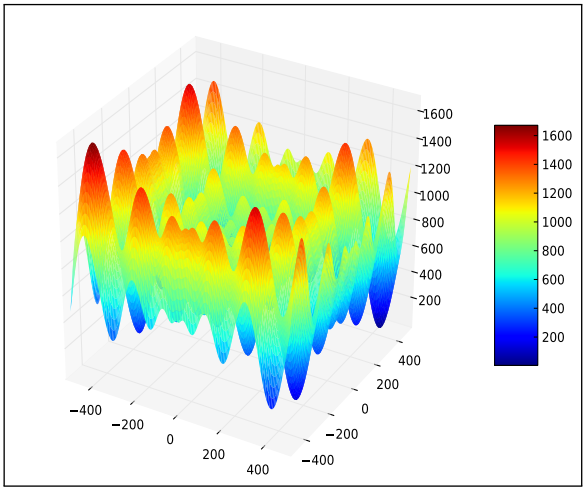

Figure 6.4: Schwefel Function

Griewank Function

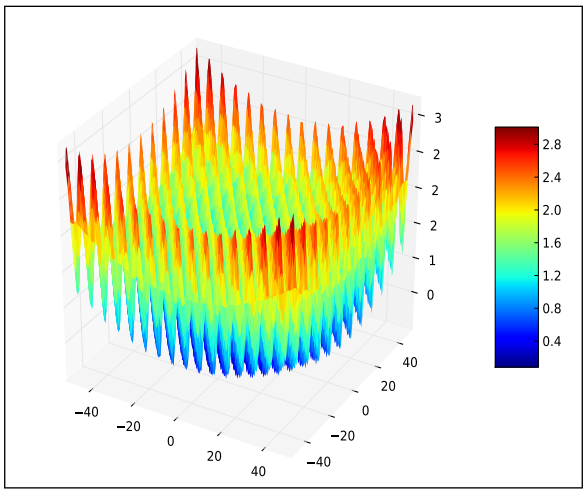

Figure 6.5: Griewank Function

Salomon Function

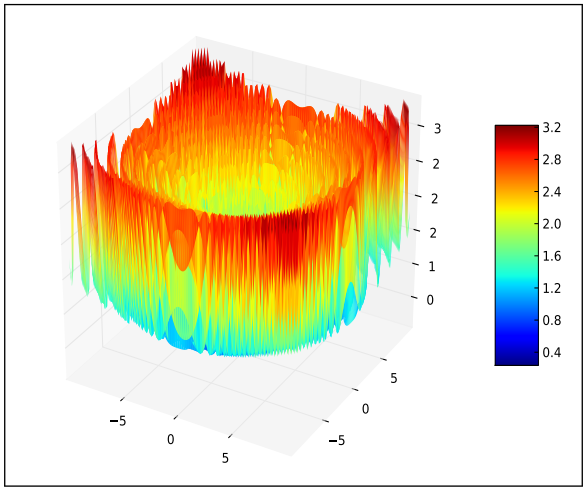

Figure 6.6: Salomon Function

Ackley

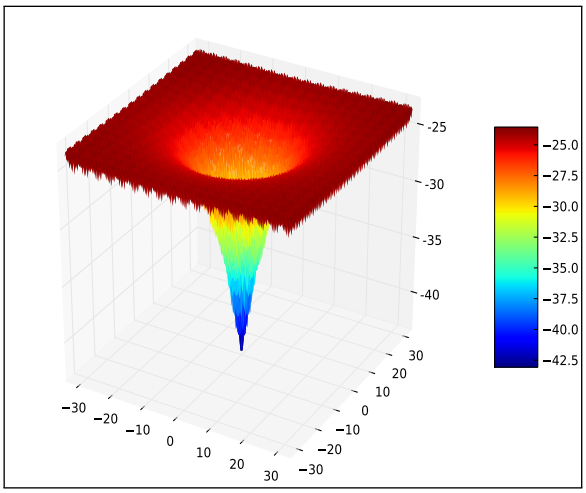

Figure 6.7: Ackley Function

Six-Hump Camel Back Function

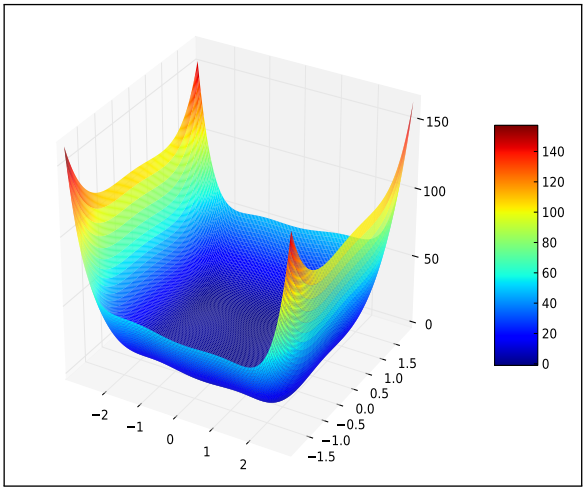

Figure 6.8: Six-Hump Camel Back Function

Shubert Function

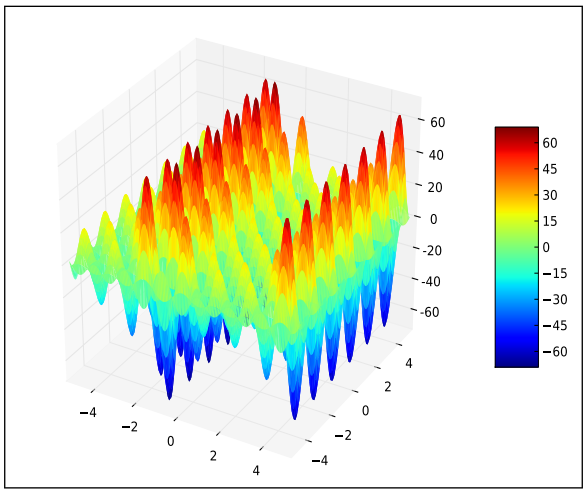

Figure 6.9: Shubert Function

Himmelblau Function

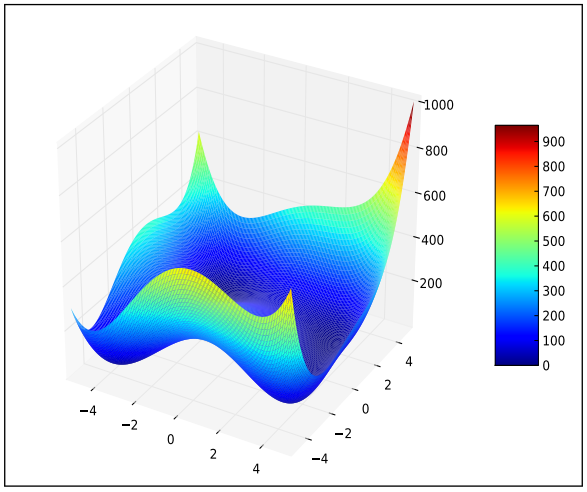

Figure 6.10: Himmelblau Function

Rosenbrock Valley Function

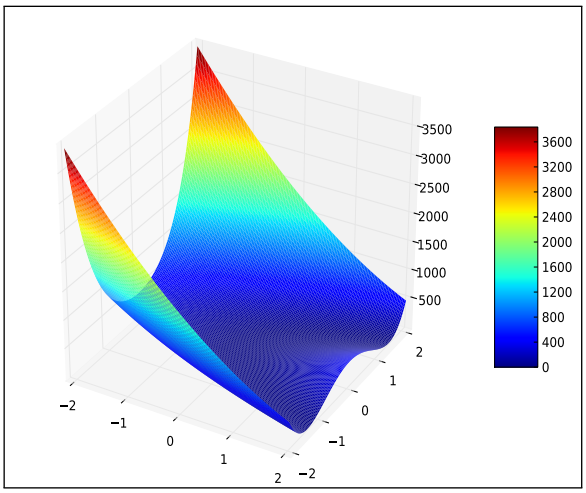

Figure 6.11: Rosenbrock Valley Function

Dropwave Function

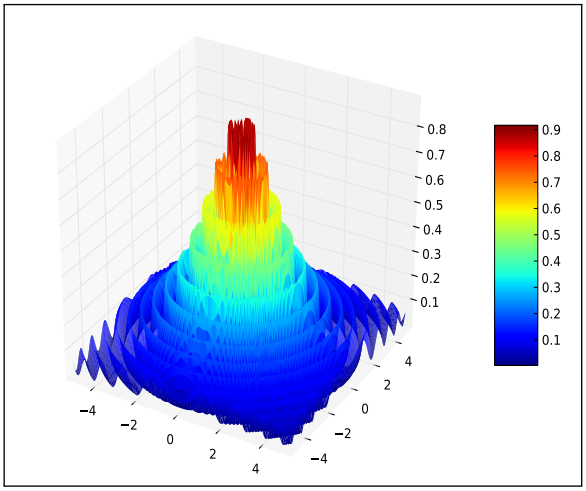

Figure 6.12: Dropwave Function

Easom Function

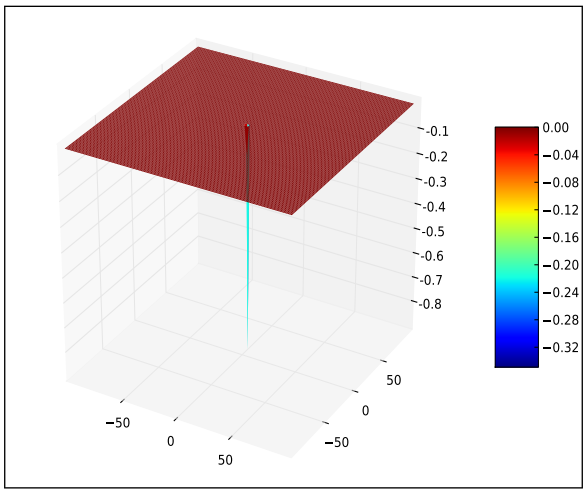

Figure 6.13: Easom Function

Branin Function

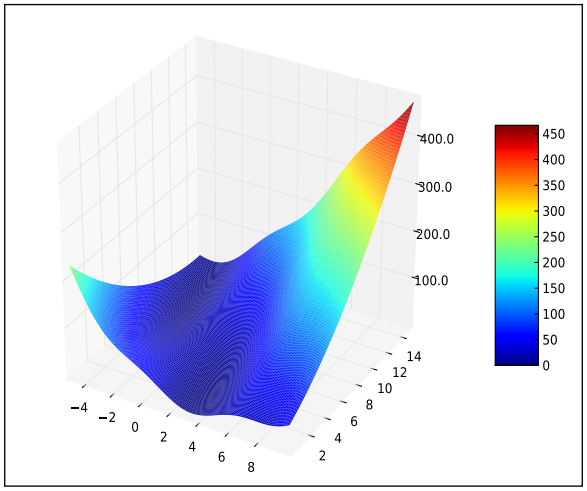

Figure 6.14: Branin Function

Michalewicz Function

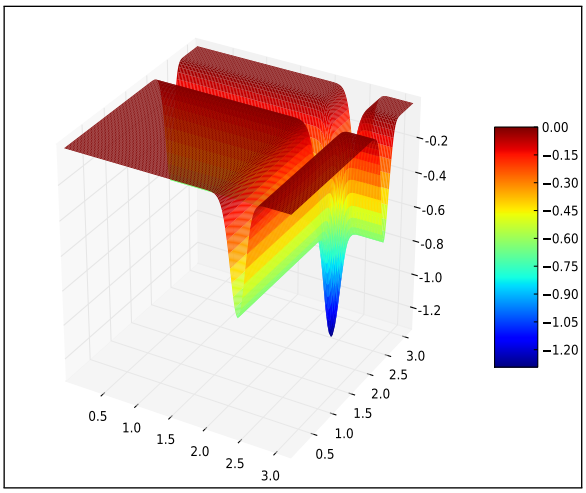

Figure 6.15: Michalewicz Function

Goldstein Function

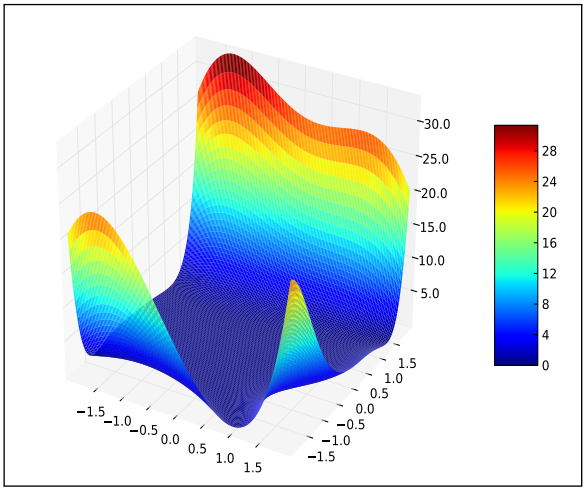

Figure 6.16: THe Goldstein Function

6.3 Results

Chapter 7

Applying the PSO to the FAP

7.1 Introduction

Particle Swarm Optimization (PSO) as discussed in the overview (see page 81) is an algorithm that is largely based on the flying behaviour exhibited by a flock of flying birds. Which is why the core of the algorithm is based upon vector math, with new positions and velocities being calculated after each iteration of the algorithm. Thus, each particle position, is represented by a D-dimensional vector and is then simulated flying through the D-dimensional space using the velocity equation (see **??**).

The performance of the Global PSO is benchmarked and outline in the previous chapter. Most of the problems to which PSO has been applied to, to date have been problems where the position of particles have a constant D-dimensional space, which is to formally state *the dimensionality of a particle position in its entirety, is constant.*

This constant dimensionality introduces a intriguing problem if you want to apply the PSO to an inherent multi-dimension problem like the Frequency Assignment Problem (FAP). In this chapter we will provide a discussion on how, we the authors, went about in applying the PSO to the FAP.

We will start of by first explaining what we deemed as a position for a particle in the Frequency Planning domain. This definition of the particle position is important because it plays a central part in how we "fly" our

particles through the Frequency planning domain. After the position definition, we will define how each position will be evaluated and formally define the fitness function our swarm will use.

Arguably the most important part of the swarm, is how we calculate the velocity of a particle and moving it to a new position in the search space. A discussion on the velocity function we have developed will be discussed in the section after the fitness function.

After the velocity function section we will discuss a new mechanism for selecting the global best which allowed us to get better fitness values and therefore direct the swarm more. Finally we will end this section of with a section on the how the swarm utilizes history to produce better results.

7.2 A Position in the Frequency Planning domain

In this section we will describe what a position is in the Frequency Planning domain. We will start of by first describing what a Frequency Plan is as well as provide the general structure to represent such a plan. We will also describe some of the hard and soft constraints and how it molds the plan to be suitable for a network.

A Frequency plan, is almost exactly as the name implies. A plan that outlines frequency usage for a wireless network. The benchmark problems we will be using all pertain to cellular phone networks. For Cellular networks, the frequency plan outlines what frequency must be allocated to what transceiver. With this basic definition, the problem seems relatively trivial to solve if one assumes that one either has a infinite number of frequencies or the amount of frequencies available to our disposal is more than the amount of transceivers in the network.

The reality is, that there are only a finite amount of frequencies available for cellphone transmissions. Hence a regulatory body needs to assign frequencies to cellphone network operators for use in their networks. A regulatory body is needed because, if a network operator just uses any frequency it wants, it is bound to interfere with someone else also utilising the frequency.

The assigned frequencies are also not a huge portion of the entire spectrum. If we look at one of our benchmark problems, Siemens1, the alloted spectrum is from frequency 16 to frequency 90. Which gives the network

operator 74 frequencies to use in its network without considering other constraints.

Besides the Electro-magnetic constraints that are also applicable here, there are regulatory constraints, like for instance frequencies in the spectrum that are by no means allowed to be used. These frequencies are referred to as globally blocked frequencies and are hard constraints. There are also locally blocked frequencies, which apply only in certain regions of the geographical landscape of the network.

As discussed in chapter 2 (page 11) and 3 (see page 23). A Cellphone network is divided into a number of cells, and each cell requires a certain number of transceivers to service its corresponding area. This number of transceivers is based on the expected volume of traffic that a particular cell will experience at peak network usage. Due to this variability in the expected amount of traffic a cell is supposed to handle, the nature of the frequency assignment problem is of a multi dimensional nature. As can be seen

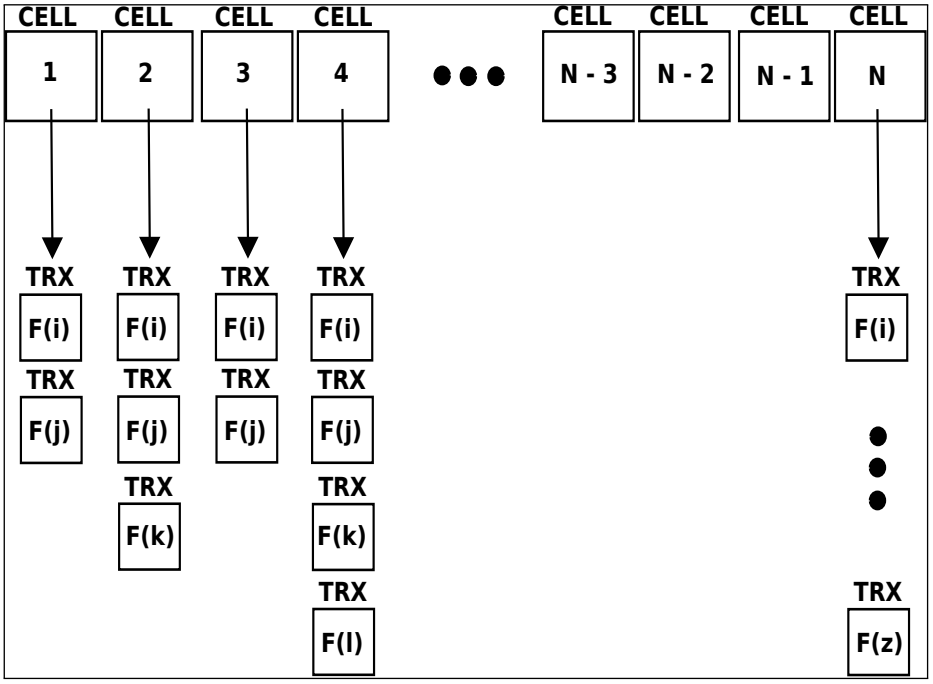

Figure 7.1: The Structure of a Frequency Plan

in figure 7.1 a cellular network can have any amount (N in the figure) of cells to attain the desired coverage over the geographical landscape. In our benchmarks problems the cellular networks have a number of cells ranging

from 500 to 1000+. The most important part of the plan, is the actual TRX's within each cell. In the figure we can clearly see, how the amount of TRX's vary from one cell to the next. $F(i)$ is a frequency at position i from the available usable spectrum.

Note based on the structure of the plan depicted in figure 7.1 there is no concept of which cell interferes with which other cell and if their is indeed interference, how much is inferred as a result. All this information isn't part of the plan. Instead this information, for purpose of this dissertation is supplied by the benchmark.

This information is referred to as the interference matrix. Within this interference matrix each entry references two cells entries Cell A and Cell B. The entry then lists the amount of interference that occurs when Cell B interferes with Cell A[1].

A Frequency Plan is a possible solution to the Frequency Assignment Problem. Therefore in the PSO, we the authors have developed, each Particle's position in the solution space is represented by a frequency plan. As mentioned earlier, moving particles through the frequency plan solution space introduces an interesting problem due to the multi-dimensionality of a plan. We will describe how particles are moved from one position to another through the solution space in section 7.4

In this section we have given a description of what a our PSO will use a position in the frequency planning domain. We outlined the general structure of what a frequency plan is as well as defined how the interference values are retrieved when two cells interfere. In the next section we will define the fitness function that our PSO's uses.

7.3 The Fitness Function

In this section we will provide a discussion on the fitness function we utilise in all the variants of our swarms to rate each individual particle position of the swarm.

As discussed in the previous section, the set of benchmark problems we apply our particle swarms to define the amount of interference incurred when two cells assigned frequencies interfere. This interference information is

[1]Interference occurs based on the electromagnetic constraints as defined in chapter 3

referred to as the interference matrix. Each pair of cells has two interference values defined, The first value referred to as Co-channel interference is when the frequency of one TRX is the equal to a TRX in the other cell. The second value, called adjacent channel interference is when the frequency of a TRX in one cell differs by 1 with another TRX from the other cell.

Evaluation of a frequency plan to determine its fitness value is a simple process. The evaluation procedure goes through each pair of cells defined in the interference matrix where it looks up both cells in the frequency plan. The second cell is said to interfere with the first cell. Therefore each TRX in the first cell is checked with all the TRX's of the other cell. Depending on whether how the frequencies differ from each other, the fitness procedure adds either the co-channel or the adjacent channel interference to a summing variable. This procedure is mathematically defined in chapter 3 see page 36 for the formal equation.

With regard to our benchmarks, not all interference values are added to the summing variable, since each of the benchmarks define a Minimum Tolerable interference variable. Which means that if a given interference value is either equal or less than this defined value it is acceptable and won't have a impact on the overall plan.

In this section we outlined the basic fitness function our PSO will use to rate the feasibility of particles after each itertation. In the next section we will define how the particle move from one iteration to the next in the solution space.

7.4 Velocity Function for Frequency Planning

The Velocity function is arguably the core of the PSO algorithm. It is the procedure by which particles in the swarm move from one point to another point in the solution space.

The velocity function doesn't blindly move a particle from one point to another but instead, it takes the particle history into account as well as the best particle in the swarm. Therefore, the velocity function is the core means by which the swarm explores the solution space. A more thorough explanation is provided on page 84.

In this section we will provide a discussion on the process we the authors went through to develop a velocity function that is suitable for particles to

move from one Frequency plan to another. We will start of explaining our first and worst method. With each method we define we will give an outline of the problems associated with it. We will end of this section with our primary method which has obtained the best results.

7.4.1 Movement in the Frequency Planning domain

The standard velocity equation works on the basis of vector math. Each particle has a velocity and position which is represented by a standard mathematical vector. The standard equation basically just alters the direction the particle is moving to move to a more promising position in the solution space.

Vector math has standard basic operations defined for adding,subtracting and multiplication. Hence, applying the PSO to problems that are either mathematical functions or problems that map well to the vector domain is a straight forward trivial process. With regard to the Frequency planning domain an important question needs to be answered. How to move one multi dimension frequency plan to another ?

We the authors answered the question by thinking of the frequency plan and its dimensionality in a different manner. The most important realization is that, we don't have to develop a procedure that moves a whole plan to another taking into the account the dimensionality etc. Rather, we opted to disregard the plan as a whole and just apply the velocity function at a much smaller scale. Thus, the velocity function is applied at the lowest level of a frequency plan.

The basic idea about our velocity function is for the movement of the swarm to be at a much finer granularity. Therefore, when a particle needs to move towards a global best particle, the velocity procedure goes into the intricate details of the particle wanting to move and the global best particle. Hence, the procedure goes into each cell defined in the frequency plan represented by the standard particle as well as the global best particle.

For each cell in the frequency plan, the standard PSO equation with inertia is applied to the TRX value. Thus each TRX value is moved in the general direction (on the frequency domain) towards the TRX value defined in the same cell of the global best frequency plan.

The velocity function we the authors have developed is therefore simplified from being a procedure that needs to move around in a multidimensional space, to move values in a 1 dimensional space. Therefore, our velocity function retains the simplicity of the original PSO algorithm as well as the general concept it is based upon.

7.4.2 Keeping frequencies bounded

We the authors, have defined the basic concept that we will use in our swarm to allow the individual particles to move around in the frequency domain. But this alone is not enough, since the swarm currently has no concept of the constraints that exist in the domain. These constraints include what frequencies are allowed to be used and which must be avoided. Therefore, the velocity function needs to be altered to make the swarm in some sense aware, and hence keep the particle positions bounded within the allowable search space.

The boundary check we implemented was fairly trivial. The check only applied when one of the following conditions were met after the calculated velocity was applied to the current position:

- If a TRX frequency is above the maximum allowable frequency (higher bound) given to the network.
- If a TRX frequency is below the minimum allowable frequency (lower bound) given to the network.

A mod operation is applied to the value to bring it within the allowable range. If for instance, the maximum allowable frequency is 50, and the TRX value (after velocity) is 56. The 56 value gets modded with 50 to produce a value of 6. This modded value is then added to the minimum allowable frequency. In essence, the value is wrapped around to always be within acceptable range.

The not so trivial case is when the frequency value is lower that the minimum frequency given to the network. This is because modding the frequency value has no effect. For example, if the lowest allowable frequency is 20 and the TRX value after movement is 15. Modding the TRX value of 15 with 20 has no effect. Therefore we have opted for the following methods to solve this problem:

1. First subtract the lower value from the minimum allowable frequency. Then add the result to the minimum allowable frequency. The resultant value is checked again whether it oversteps the bounds of the maximum allowable frequency and bounded accordingly.

2. Add the lower value to the minimum allowable frequency. The resultant value is checked whether it oversteps the bounds of the maximum allowable frequency and bounded accordingly.

3. Repeatedly subtract the lower value from the maximum allowable frequency until the resultant frequency is within the acceptable frequency range.

An important notion to consider is that based on the velocity equation it is entirely in the realm of possibility that a TRX value after movement might contain a negative value. Our boundary check solves this problem by first taking the absolute value of negative TRX value. The boundary check then treats the now positive TRX value, as a normal value that needs to be bounded.

7.4.3 Using indices instead of frequencies

In the first iteration of our velocity equation the swarm worked with raw frequency values. But upon closer inspection and the frequency range the swarm was using to move around we noticed that the swarm wasn't prohibited from using Globally Blocked Channels or Locally Blocked Channels. Hence, the swarm increasingly moved towards allocating these prohibited values to TRX's since the fitness function doesn't penalize the use of these frequency values. This is due in part because these values are under no circumstances allowed to be used and thus the fitness function isn't designed to check for these values.

To solve this problem, we the authors proposed two solutions:

1. Modify the fitness function to penalize a frequency plan if it uses any Globally Blocked Channels or Locally Blocked Channels.

2. Instead of letting the swarm work with raw frequency values, rather let the swarm work with array indices. These array indices indicate positions in a array that has been pre-filled with only *valid* frequencies.

Thus the swarm then moves around in a range from 0 to F, where F is the size of the array.

With the first solution, the fitness function will have to be modified to levy a penalty if a prohibited frequency value is used. The first proposed solution was disregarded because it introduces complexity which can be completely avoided with the second proposed solution.

Where as with the second solution the fitness function will not have to be modified and the boundary check is simplified since there is no need to check for a lower bound anymore. The boundary check only now has to check for negative index values and if the higher bound is violated which is now, the size of the array.

7.5 Building a Global Best

Selection of the global best particle by the swarm is a very important procedure. After the swarm has determined which particle has achieved the best position, the swarm enters the velocity function phase.

As discussed in the velocity function section and in the Particle Velocity section on page 84 each particle position is then modified to move in the general direction of the global best and personal best position. Therefore the global best acts as a beacon for the rest of the swarm in the solution space to indicate where good solutions seem to be for the rest of the swarm.

Initially the method we the authors used for selecting the global best in our PSO for the FAP didn't not differ at all from the traditional global PSO algorithm. The Swarm would loop through all the particle and apply the fitness function to determine the fitness of the particles position. The algorithm would then iterate over the swarm to determine which particle has the lowest fitness or in Frequency Planning terms or in Frequency planning terms, which plan has the lowest interference overall. The particle with the lowest fitness would then be the global best for that iteration.

Selecting the global best by evaluating the position as a whole seems to be a natural fit. But if we were to analyse a frequency plan in more detail; specifically how a single value in a resident TRX of a particular cell can affect the interference generated by the whole cell. One can come to the conclusion that it is entirely in the realm of possibility that, one bad value

in a TRX can overshadow a potential good value of a neighboring TRX similarly the total interference of a cell can overshadow a cell that had very low interference.

Therefore, in the traditional method of selecting the global best, a particle is actually selected because it contains less overshadowing TRX's. Hence potential good TRX values get lost.

We the authors went about to rectify this problem by exploiting the information the fitness exposes to us much more thoroughly. The information exposed by the fitness function allows us to see what effects certain values have on the interference of the cell. Therefore, to make better use of this information we developed two methods, each one being more fine grained than the other.

1. Besides the particle knowing its fitness, we changed the structure of a cell to allow a cell to know the interference it resident TRX's generate.

2. Besides the particle also storing the total fitness, each TRX of a particular cell also stores the interference it has generated.

With both these methods, the global best selection scheme needs to be changed to allow the swarm to take advantage of this newly exposed information. The scheme we the authors implemented to take advantage of the information does not look at a particle position fitness as a whole. Instead, the procedure builds a global best position with this new information.

The global best building scheme works slightly different for each method. For method 1, the scheme loops through the entire swarm and selects the cell with the lowest interference. It then copy this cell and places it in the global best position at the same position it was found at. For method 2 the scheme follows the same procedure with the only difference being that the scheme now copies a TRX value and places it at the same position in the global best position.

When the swarm executed using this new scheme initially it didn't produce good results. This is largely due to the interference information for a cell in method 1 and for a TRX in method 2 gets reset to 0 after each iteration. Which seems to be correct, but in essence what is occurring is that after each iteration the swarm is effectively discarding all information gathered in that iteration.

To enable to this information to direct the swarm a bit more, we changed the algorithm to not reset the interference values per TRX and per Cell to 0. Instead, the interference values for an iteration is now added to the previous iteration interference values stored by the cell and TRX. This also has the net effect, that bad decisions made by the swarm for a particular particle get progressively worse as the swarm progresses.

We the authors, experimented with reseting the gathered information after a certain number of iterations. But it had no significant impact on the overall solution being produced. In some cases, the swarm moved the much worse solution and wasn't able to improve best particle to the previous best particle levels before the information reset.

With this small change to the structure of cells and TRX's. The swarm produced much better frequency plans that had lower total interference that all previous generated plans. In the next chapter we will present these results.

In this section we discussed a new method for selecting the global best particle from a swarm of particles. We also outlined the structural changes we had to make to enable to algorithm to use these new structural changes to generate lower frequency plans. Finally we discussed some of the problems encountered and how we the authors went about solving it. In the next section we will discuss how we incorporated the notion of a particle or rather a cell, keeping history of previous TRX's values.

7.6 Keeping History

In the traditional PSO history is retained by using the particle personal best position to direct the next movement of the particle. Other methods such as inertia also allows history to direct the movement of the particle. With regard to our PSO on the FAP, the algorithm we have develop also uses these concepts. But these concepts are not able to effectively exploit the history of a particle since they have no concept of what combinations of frequency values have been previously used in a cell.

In the algorithm we developed, we borrowed the concept of Tabu Lists from the Tabu Search Algorithm. Using Tabu lists a particle will be able to better exploit the solution space it currently finds itself in. In our algorithm,

we incorporated Tabu Lists by adding to each cell a list which keeps track of resident TRX values in the cell for 20 iterations.

Care must be taken to select a max size of the Tabu List since one wants to keep enough history so that the search space can be adequately exploited. The Max Tabu List size must be less than the amount of available frequencies. Finally the max Tabu List cannot be too large, since the amount of checks the algorithm has to do to see if a value is Tabu is a very expensive operation. The operation is expensive, since for each potential value the Tabu List needs to be iterated through to see if the value is Tabu.

Only after the newly calculated velocity has been applied to the current position of a particular particle is the position checked for *collisions*. We say a collision occurs once a value is found to be in the Tabu List. To resolve a collision, we simply randomly select a new valid value which is just a index to a valid frequency value. This randomly selected value is then also checked if it collides with another value in the Tabu List.

Generation of randomly values continues until a counter variable reaches a value of 50 or any other predefined value. The collision resolving procedure then just simply accepts the last generated value as valid. The reason for the counter variable is because since no track is kept on previously generated values, the collision resolving procedure might indefinitely keep generating values all of them colliding with values presently in the Tabu List.

By incorporating Tabu Lists and the collision resolving procedure, the efficiency of the algorithm reduced dramatically. To increase efficiency of the operations in the algorithm, we paralyzed the collision resolving procedure, the velocity function and the sanitation procedures [2]. Therefore, multiple cells are moved simultaneously to towards other corresponding cells.

The paralization of these procedures increased efficiency of the algorithm. We the authors also noticed a slight side effect because of these operations. The randomness of our random number generator decreased. This effect was noticed because we outputed the counter variable in the collision resolver procedure. When the value was being outputted our algorithm produced much better results.

The reason for this is because outputting the variable inherently introduces a delay and therefore, the random numbers generators in other threads

[2]The sanitation procedure consist of all the constraint checks

have different seed values. Hence, with a delay in each parallel thread the numbers generated by the random number generator is more distinct. Without the delay, the because the of the parallelization some threads might start of with similar seed values because the current time is used a seed value for our random number generator [3].

Keeping the delay in mind and the effect is has on the final result, we introduced a delay in our collision resolving procedure. The reason the particular procedure was selected was because it was where the effect of delay was first noticed. After performing tests with delays of 5 milliseconds (ms), 10 ms, 15, 20 we settled on 20 ms since it was gave just enough time for a reasonably distinction between seed values used by other parallel threads.

In this section we discussed how we used Tabu Lists from Tabu Search to keep history. We discussed the alterations we had to make to the algorithm to incorporate the new feature as well as outlined the necessary procedures needed to effectively exploit the Tabu Lists. Finally we discussed the effects the Tabu Lists had on the performance of the algorithm and how we the authors went about solving it.

7.7 Summary

[3] This is the default behaviour of the .Net 4.0 random number generator

Bibliography

[1] Karen Aardal, Cor Hurkens, Jan Karel Lenstra, and Sergey Tiourine. Algorithm for radio link frequency assignment: The calma project. *Operations Research*, 50(6):968–980, Nov - Dec 2002.

[2] Karen I. Aardal, Stan P. M. van Hoesel, Arie M. C. A. Koster, Carlo Mannino, and Antonio Sassano. Models and solution techniques for frequency assignment problems. *4OR: A Quarterly Journal of Operations Research*, 1(4):261–317, December 2004.

[3] E. Alba, F. Luna, A. J. Nebro, and J. M. Troya. Parallel heterogeneous genetic algorithms for continuous optimization. *Parallel Comput.*, 30(5-6):699–719, 2004.

[4] Mahmoud H. Alrefaei and Sigrn Andradttir. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science*, 45:748–764, 1999.

[5] S. Areibi and A. Vannelli. Circuit partitioning using a tabu search approach. In *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, pages 1643 –1645, 3-6 1993.

[6] A. Augugliaro, L. Dusonchet, and E. R. Sanseverino. An evolutionary parallel tabu search approach for distribution systems reinforcement planning. *Advanced Engineering Informatics*, 16(3):205 – 215, 2002.

[7] I. Badarudin, A.B.M. Sultan, M.N. Sulaiman, A. Mamat, and M.T.M. Mohamed. Metaheuristic approaches for optimizing agricultural land areas. In *Data Mining and Optimization, 2009. DMO '09. 2nd Conference on*, pages 28 –31, 27-28 2009.

[8] T.R. Benala, S.D. Jampala, S.H. Villa, and B. Konathala. A novel approach to image edge enhancement using artificial bee colony optimization algorithm for hybridized smoothening filters. In *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 1071 –1076, 9-11 2009.

[9] Christian Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4):353 – 373, 2005.

[10] Bruce M. Blumberg. *Exploring Artificial Intelligence in the New Millenium*, chapter D-Learning: what learning in dogs tells us about building characters that learn what they ought to learn, pages 37–67. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[11] Ralf Borndrfer, Andreas Eisenbltter, Martin Grtschel, and Alexander Martin. The orientation model for frequency assignment problems. Technical report, Konrad-Zuse-Zentrum Berlin, 1998.

[12] Jeffrey E. Boyd, Gerald Hushlak, and Christian J. Jacob. Swarmart: interactive art from swarm intelligence. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 628–635, New York, NY, USA, 2004. ACM.

[13] L. Cavique, C. Rego, and I. Themido. Subgraph ejection chains and tabu search for the crew scheduling problem. *The Journal of the Operational Research Society*, 50:608–616, 1999.

[14] A. Chawla, S. Mukherjee, and B. Karthikeyan. Characterization of human passive muscles for impact loads using genetic algorithm and inverse finite element methods. *Biomechanics and Modeling in Mechanobiology*, 8(1):67–76, 2009.

[15] Jeremy K. Chen, Gustavo de Veciana, and Theodore S. Rappaport. Site-specific knowledge and interference measurement for improving frequency allocatoins in wireless networks. *IEEE Transactions on Vehicular Technology*, 58:2366–2376, 2009.

[16] Chin Soon Chong, Appa Iyer Sivakumar, Malcolm Yoke Hean Low, and Kheng Leng Gay. A bee colony optimization algorithm to job shop scheduling. In *WSC '06: Proceedings of the 38th conference on*

Winter simulation, pages 1954–1961. Winter Simulation Conference, 2006.

[17] Agnieszka Debudaj-Grabysz and Zbigniew J. Czech. Theoretical and practical issues of parallel simulated annealing. In *PPAM'07: Proceedings of the 7th international conference on Parallel processing and applied mathematics*, pages 189–198, Berlin, Heidelberg, 2008. Springer-Verlag.

[18] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243 – 278, 2005.

[19] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243 – 278, 2005.

[20] Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Gener. Comput. Syst.*, 16(9):851–871, 2000.

[21] Audrey Dupont, Andrea Carneiro Linhares, Christian Artigues, Dominique Feillet, Philippe, and Michel Vasquez. The dynamic frequency assignment problem. *European Journal of Operational Research*, 195:75–88, 2009.

[22] Andreas Eisenblätter. Assigning frequencies in gsm networks. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 2001.

[23] Andreas Eisenblätter. *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds*. PhD thesis, Technische Universität Berlin, Berlin, Germany, 2001.

[24] H.M. Elkamchouchi, H.M. Elragal, and M.A. Makar. Channel assignment for cellular radio using particle swarm optimization. In *Radio Science Conference, 2006. NRSC 2006. Proceedings of the Twenty Third National*, volume 0, pages 1 –9, 14-16 2006.

[25] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[26] Andries P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley Publishing, 2007.

[27] N. Fescioglu-Unver and M.M. Kokar. Application of self controlling software approach to reactive tabu search. In *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, pages 297 –305, 20-24 2008.

[28] L. M. Gambardella, D Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *The Journal of the Operational Research Society*, 50(2):167–176, Feb 1999.

[29] Gautam Garai and B. B. Chaudhuri. A distributed hierarchical genetic algorithm for efficient optimization and pattern matching. *Pattern Recogn.*, 40(1):212–228, 2007.

[30] Antonio Gómez-Iglesias, Miguel A. Vega-Rodríguez, Francisco Castejón, Miguel Cárdenas-Montes, and Enrique Morales-Ramos. Artificial bee colony inspired algorithm applied to fusion research in a grid computing environment. In *PDP '10: Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pages 508–512, Washington, DC, USA, 2010. IEEE Computer Society.

[31] Mohan Gopalakrishnan, Ke Ding, Jean-Marie Bourjolly, and Srimathy Mohan. A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Manage. Sci.*, 47(6):851–863, 2001.

[32] J.S Graham, R. Montemanni, J. N J. Moon, and D. H. Smith. Frequency assignment. multiple interference and binary constraints. *Wireless Networking*, 14:449–464, 2008.

[33] Timo Hämäläinen, Harri Klapuri, Jukka Saarinen, Pekka Ojala, and Kimmo Kaski. Accelerating genetic algorithm computation in tree shaped parallel computer. *J. Syst. Archit.*, 42(1):19–36, 1996.

[34] Julia Handl and Bernd Meyer. Ant-based and swarm-based clustering. *Swarm Intelligence*, 1(2):95–113, December 2007.

[35] Abdel-rahman Hedar and Masao Fukushima. Tabu search directed by direct search methods for nonlinear global optimization. *European Journal of Operational Research*, 170:329–349, 2006.

[36] Alan Herz, David Schindl, and Nicolas Zufferey. Lower bounding and tabu search procedures for the frequency assignment problem with polarization constraints. *4OR: A Quarterly Journal of Operations Research*, 3:139–161, 2005.

[37] L. Houssin, C. Artigues, and E. Corbel. Frequency allocation problem in a sdma satellite communication system. In *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, pages 1599 –1604, 6-9 2009.

[38] http://fap.zib.de/. Fap web, 2010.

[39] Shun-Fa Hwang and Rong-Song He. Improving real-parameter genetic algorithm with simulated annealing for engineering problems. *Adv. Eng. Softw.*, 37(6):406–418, 2006.

[40] Lhassane Idoumghar and Ren Schott. Two distributed algorithms for the frequency assignment problem in the field of radio broadcasting. *IEEE Transactions on Broadcasting*, 55:223–229, 2009.

[41] D.M. Jaeggi, G.T. Parks, T. Kipouros, and P.J. Clarkson. The development of a multi-objective tabu search algorithm for continuous optimisation problems. *European Journal of Operational Research*, 185(3):1192 – 1212, 2008.

[42] A.A. Javadi, R. Farmani, and T.P. Tan. A hybrid intelligent genetic algorithm. *Advanced Engineering Informatics*, 19(4):255 – 262, 2005. Computing in Civil Engineering.

[43] I. Jovanoski, I. Chorbev, D. Mihajlov, and I. Dimitrovski. Tabu search parameterization and implementation in a constraint programming library. In *EUROCON, 2007. The International Conference on Computer as a Tool*, pages 459 –464, 9-12 2007.

[44] Wei jun Xia and Zhi ming Wu. A hybrid particle swarm optimization approach for the job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 29(3):360–366, June 2006.

[45] Vijay Kalivarapu, Jung-Leng Foo, and Eliot Winer. Improving solution characteristics of particle swarm optimization using digital

pheromones. *Structural and Multidisciplinary Optimization*, 37:415 – 427, 2009.

[46] Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108 – 132, 2009.

[47] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J. of Global Optimization*, 39(3):459–471, 2007.

[48] Il kwon Jeong and Ju jang Lee. Adaptive simulated annealing genetic algorithm for system identification. *Engineering Applications of Artificial Intelligence*, 9(5):523 – 532, 1996.

[49] Sergio Ledesma, Miguel Torres, Donato Hernndez, Gabriel Avia, and Guadalupe Garca. Temperature cycling on simulated annealing for neural network learning. In *MICAI 2007: Advances in Artificial Intelligence*, pages 161–171, 2007.

[50] K. Lenin and M.R. Mohan. Attractive and repulsive particle swarm optimization for reactive power optimization. *Journal of Engineering and Applied Sciences*, 1(4):288–292, 2006.

[51] Yuming Liang and Lihong Xu. Mobile robot global path planning using hybrid modified simulated annealing optimization algorithm. In *GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pages 309–314, New York, NY, USA, 2009. ACM.

[52] Nguyen Tung Linh and Nguyen Quynh Anh. Application artificial bee colony algorithm (abc) for reconfiguring distribution network. In *Computer Modeling and Simulation, 2010. ICCMS '10. Second International Conference on*, volume 1, pages 102 –106, 22-24 2010.

[53] Hongbo Liu, Ajith Abraham, and Maurice Clerc. Chaotic dynamic characteristics in swarm intelligence. *Applied Soft Computing*, 7(3):1019 – 1026, 2007.

[54] Wen Liu, Haixiang Shi, and Lipo Wang. Minimizing interference in satellite communications using chaotic neural networks. *Natural Com-*

putation, 2007. ICNC 2007. Third International Conference on, 2:441 –444, aug. 2007.

[55] Francisco Luna, Christian Blum, Enrique Alba, and Antonio J. Nebro. Aco vs eas for solving a real-world frequency assignment problem in gsm networks. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 94–101, New York, NY, USA, 2007. ACM.

[56] H. Mahmoudzadeh and K. Eshghi. A metaheuristic approach to the graceful labeling problem of graphs. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 84 –91, 1-5 2007.

[57] S. Mallela and L.K. Grover. Clustering based simulated annealing for standard cell placement. In *Design Automation Conference, 1988. Proceedings., 25th ACM/IEEE*, pages 312 –317, 12-15 1988.

[58] Vittoria Maniezzo and Roberto Montemanni. An exact algorithm for the min-interference frequency assignment problem. Technical report, Department of Computer Science, University of Bologna, 2000.

[59] Y. Marinakis, M. Marinaki, and N. Matsatsinis. A hybrid discrete artificial bee colony - grasp algorithm for clustering. In *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, pages 548 –553, 6-9 2009.

[60] Asha Mehrotra. *GSM System Engineering*. Arteh House, Inc, 1997.

[61] David Meignan, Jean-Charles Créput, and Abderrafiaa Koukam. A cooperative and self-adaptive metaheuristic for the facility location problem. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 317–324, New York, NY, USA, 2009. ACM.

[62] George Jiri Mejtsky. The improved sweep metaheuristic for simulation optimization and application to job shop scheduling. In *WSC '08: Proceedings of the 40th Conference on Winter Simulation*, pages 731–739. Winter Simulation Conference, 2008.

[63] P.R.S. Mendonca and L.P. Caloba. New simulated annealing algorithms. In *Circuits and Systems, 1997. ISCAS '97., Proceedings of*

1997 IEEE International Symposium on, volume 3, pages 1668 –1671 vol.3, 9-12 1997.

[64] Marie-Bernadette Pautet Michel Mouly. *The GSM System for Mobile Communications*. Cell & Sys, 1992.

[65] Qi Ming-yao, Miao Li-xin, Zhang Le, and Xu Hua-yu. A new tabu search heuristic algorithm for the vehicle routing problem with time windows. In *Management Science and Engineering, 2008. ICMSE 2008. 15th Annual Conference Proceedings., International Conference on*, pages 1648 –1653, 10-12 2008.

[66] Christopher K. Monson and Kevin D. Seppi. Adaptive diversity in pso. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 59–66, New York, NY, USA, 2006. ACM.

[67] Roberto Montemanni. *Upper and Lower bounds for the fixed spectrum frequency assignment problem*. PhD thesis, School of Tecnology, University of Glamorgan, 2001.

[68] Roberto Montemanni and Derek H. Smith. Heuristic manipulation, tabu search and frequency assignment. *Comput. Oper. Res.*, 37(3):543–551, 2008.

[69] Angel E. Mu noz Zavala, Arturo Hernández Aguirre, and Enrique R. Villa Diharce. Constrained optimization via particle evolutionary swarm optimization algorithm (peso). In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 209–216, New York, NY, USA, 2005. ACM.

[70] Garry Mullet. *Wireless telecommunications systems and networks*. Thomsan Delmar Learning, 2006.

[71] Amit Nagar, Sunderesh S. Heragu, and Jorge Haddock. A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem. *Annals of Operations Research*, 63(3):397–414, June 1996.

[72] B. Natrajan and B.E. Rosen. Image enhancement using very fast simulated reannealing. In *Image Analysis and Interpretation, 1996.,*

Proceedings of the IEEE Southwest Symposium on, pages 230 –235, 8-9 1996.

[73] Lin Ni and Hong-Ying Zheng. An unsupervised intrusion detection method combined clustering with chaos simulated annealing. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 6, pages 3217 –3222, 19-22 2007.

[74] Koji Nonobe and Toshihide Ibaraki. A tabu search approach to the constraint satisfaction problem as a general problem solver. *European Journal of Operational Research*, 106(2-3):599 – 623, 1998.

[75] E. Nowicki and S. Zdrzalka. Single machine scheduling with major and minor setup times - a tabu search approach. *The Journal of the Operational Research Society*, 47:1054–1064, 1996.

[76] Michael O’Neill and Anthony Brabazon. Self-organising swarm (soswarm). *Soft Comput.*, 12(11):1073–1080, 2008.

[77] W. Paszkowicz. Properties of a genetic algorithm equipped with a dynamic penalty function. *Computational Materials Science*, 45(1):77 – 83, 2009. Selected papers from the E-MRS 2007 Fall Meeting Symposium G: Genetic Algorithms in Materials Science and Engineering - GAMS-2007.

[78] Thomas La Porta Patrick Traynor, Patrick McDaniel. *Security for Telecommunications Networks*, volume 40 of *Advances in Information Security*. Springer US, 2008.

[79] Pedro Pereira, Fernando Silva, and Nuno A. Fonseca. Biored - a genetic algorithm for pattern detection in biosequences. In *2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008)*, pages 156–165, 2009.

[80] Jean-Yves Potvin. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 63:337–370, 1996.

[81] Nilesh B. Prajapati, Rupal R. Agravat, and Mosin I. Hasan. Comparative study of various cooling schedules for location area planning in cellular networks using simulated annealing. *Networks & Communications, International Conference on*, 0:146–150, 2009.

[82] Abraham P. Punnen and Y. P. Aneja. A tabu search algorithm for the resource-constrained assignment problem. *The Journal of the Operational Research Society*, 46(2):214–220, Feb 1995.

[83] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *Evolutionary Computation, IEEE Transactions on*, 8(3):240 – 255, june 2004.

[84] Andrea Reese. Random number generators in genetic algorithms for unconstrained and constrained optimization. *Nonlinear Analysis: Theory, Methods & Applications*, 71(12):e679 – e692, 2009.

[85] D. J. Reid. Genetic algorithms in constrained optimization. *Mathematical and Computer Modelling*, 23(5):87 – 111, 1996.

[86] Angelos N. Rouskas, Michael G. Kazantzakis, and Miltiades E. Anagnostou. Minimizing of frequency assignment span in cellular networks. *IEEE Transactions on Vehicular Technology*, 48:873–882, 1999.

[87] P. Demestichas E. Tzifa S. Kotrotsos, G. Kotsakis and V. Demesticha. Forumlation and computationally efficient algirithms for an interference-orientated version of the frequency assignment problem. *Wireless Personal Communications*, 18:289–317, 2001.

[88] Mohsen Saemi and Morteza Ahmadi. Integration of genetic algorithm and a coactive neuro-fuzzy inference system for permeability prediction from well logs data. *Transport in Porous Media*, 71(3):273–288, February 2008.

[89] H.G. Sandalidis, P.P. Stavroulakis, and J. Rodriguez-Tellez. An efficient evolutionary algorithm for channel resource management in cellular mobile systems. *Evolutionary Computation, IEEE Transactions on*, 2(4):125 –137, nov 1998.

[90] Matthew Settles and Terence Soule. Breeding swarms: a ga/pso hybrid. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 161–168, New York, NY, USA, 2005. ACM.

[91] Patrick Siarry, Alain Pétrowski, and Mourad Bessaou. A multipopulation genetic algorithm aimed at multimodal optimization. *Adv. Eng. Softw.*, 33(4):207–213, 2002.

[92] Alok Singh. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, 9(2):625 – 631, 2009.

[93] J. R. Slagle, A. Bose, P. Busalacchi, and C. Wee. Enhanced simulated annealing for automatic reconfiguration of multiprocessors in space. In *IEA/AIE '89: Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pages 401–408, New York, NY, USA, 1989. ACM.

[94] K.G. Srinivasa, K.R. Venugopal, and L.M. Patnaik. A self-adaptive migration model genetic algorithm for data mining applications. *Information Sciences*, 177(20):4295 – 4313, 2007.

[95] Marc St-Hilaire, Steven Chamberland, and Samuel Pierre. A tabu search algorithm for the global planning problem of third generation mobile networks. *Comput. Electr. Eng.*, 34(6):470–487, 2008.

[96] Anand Prabhu Subramanian, Himanshu Gupta, Samir R. Das, and Jing Cao. Minimum interference channel assignment multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, 7(7):1459–1473, December 2008.

[97] B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *The Journal of the Operational Research Society*, 57(10):1143–1160, 2006.

[98] Ashish Sureka and Peter R. Wurman. Applying metaheuristic techniques to search the space of bidding strategies in combinatorial auctions. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 2097–2103, New York, NY, USA, 2005. ACM.

[99] Ashish Sureka and Peter R. Wurman. Using tabu best-response search to find pure strategy nash equilibria in normal form games. In *AAMAS '05: Proceedings of the fourth international joint conference on*

Autonomous agents and multiagent systems, pages 1023–1029, New York, NY, USA, 2005. ACM.

[100] Dusan Teodorovic, Panta Lucic, Goran Markovic, and Mauro Dell' Orco. Bee colony optimization: Principles and applications. In *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, pages 151 –156, 25-27 2006.

[101] T.O. Ting, M.V.C. Rao, and C.K. Loo. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. *Power Systems, IEEE Transactions on*, 21(1):411 – 418, feb. 2006.

[102] Raj Gaurang Tiwari, Mohd. Husain, Sandeep Gupta, and Arun Pratap Srivastava. A new ant colony optimization meta-heuristic algorithm to tackle large optimization problem. In *COMPUTE '10: Proceedings of the Third Annual ACM Bangalore Conference*, pages 1–4, New York, NY, USA, 2010. ACM.

[103] Vedat Toğan and Ayşe T. Daloğlu. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Comput. Struct.*, 86(11-12):1204–1218, 2008.

[104] Nguyen Thanh Trung and Duong Tuan Anh. Comparing three improved variants of simulated annealing for optimizing dorm room assignments. In *Computing and Communication Technologies, 2009. RIVF '09. International Conference on*, pages 1 –5, 13-17 2009.

[105] Hsien-Yu Tseng and Chang-Ching Lin. A simulated annealing approach for curve fitting in automated manufacturing systems. *Journal of Manufacturing Technology Management*, 18:202 – 216, 2007.

[106] Roger L. Wainwright. A family of genetic algorithm packages on a workstation for solving combinatorial optimization problems. *SIGICE Bull.*, 19(3):30–36, 1994.

[107] Z. G. Wang, Y. S. Wong, and M. Rahman. Development of a parallel optimization method based on genetic simulated annealing algorithm. *Parallel Comput.*, 31(8+9):839–857, 2005.

[108] N. A. Wassan. A reactive tabu search for the vehicle routing problem. *The Journal of the Operation Research Society*, 57(1):111–116, Jan 2006.

[109] Xian-Huan Wen, Tina Yu, and Seong Lee. Coupling sequential-self calibration and genetic algorithms to integrate production data in geostatistical reservoir modeling. In *Geostatistics Banff 2004*, volume 14 of *Quantitative Geology and Geostatistics*, pages 691–701. Springer Netherlands, 2005.

[110] Dennis Weyland. Simulated annealing, its parameter settings and the longest common subsequence problem. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 803–810, New York, NY, USA, 2008. ACM.

[111] Andreas Windisch, Stefan Wappler, and Joachim Wegener. Applying particle swarm optimization to software testing. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1121–1128, New York, NY, USA, 2007. ACM.

[112] Lihua Wu and Yuyun Wang. An introduction to simulated annealing algorithms for the computation of economic equilibrium. *Computational Economics*, 12:151–169, 1998.

[113] Yiliang Xu, Meng Hiot Lim, and Yew-Soon Ong. Automatic configuration of metaheuristic algorithms for complex combinatorial optimization problems. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2380 –2387, 1-6 2008.

[114] Jingan Yang and Yanbin Zhuang. An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem. *Appl. Soft Comput.*, 10(2):653–660, 2010.

[115] Dominic C O'Brien Yangyang Zhang. Fixed channel assignment in cellular radio networks using particle swarm optimization. In *Proceedings of the Internation Symposium of Industrial Electronics*, June 2005.

[116] Quan Yuan, Feng Qian, and Wenli Du. A hybrid genetic algorithm with the baldwin effect. *Information Sciences*, 180(5):640 – 652, 2010.

[117] L. Zhang, S. Guo, Y. Zhu, and A. Lim. A tabu search algorithm for the safe transportation of hazardous materials. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 940–946, New York, NY, USA, 2005. ACM.