Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem

Adil Baykasoğlu[1], Lale Özbakır[2] and Pınar Tapkan[2]
[1]University of Gaziantep, Department of Industrial Engineering
[2]Erciyes University, Department of Industrial Engineering
Turkey

1. Introduction

There is a trend in the scientific community to model and solve complex optimization problems by employing natural metaphors. This is mainly due to inefficiency of classical optimization algorithms in solving larger scale combinatorial and/or highly non-linear problems. The situation is not much different if integer and/or discrete decision variables are required in most of the linear optimization models as well. One of the main characteristics of the classical optimization algorithms is their inflexibility to adapt the solution algorithm to a given problem. Generally a given problem is modelled in such a way that a classical algorithm like simplex algorithm can handle it. This generally requires making several assumptions which might not be easy to validate in many situations. In order to overcome these limitations more flexible and adaptable general purpose algorithms are needed. It should be easy to tailor these algorithms to model a given problem as close as to reality. Based on this motivation many nature inspired algorithms were developed in the literature like genetic algorithms, simulated annealing and tabu search. It has also been shown that these algorithms can provide far better solutions in comparison to classical algorithms. A branch of nature inspired algorithms which are known as swarm intelligence is focused on insect behaviour in order to develop some meta-heuristics which can mimic insect's problem solution abilities. Ant colony optimization, particle swarm optimization, wasp nets etc. are some of the well known algorithms that mimic insect behaviour in problem modelling and solution. Artificial Bee Colony (ABC) is a relatively new member of swarm intelligence. ABC tries to model natural behaviour of real honey bees in food foraging. Honey bees use several mechanisms like waggle dance to optimally locate food sources and to search new ones. This makes them a good candidate for developing new intelligent search algorithms. In this chapter an extensive review of work on artificial bee algorithms is given. Afterwards, development of an ABC algorithm for solving generalized assignment problem which is known as NP-hard problem is presented in detail along with some comparisons.

It is a well known fact that classical optimization techniques impose several limitations on solving mathematical programming and operational research models. This is mainly due to inherent solution mechanisms of these techniques. Solution strategies of classical optimization algorithms are generally depended on the type of objective and constraint

functions (linear, non-linear etc.) and the type of variables used in the problem modelling (integer, real etc.). Their efficiency is also very much dependent on the size of the solution space, number of variables and constraints used in the problem modelling, and the structure of the solution space (convex, non-convex, etc.). They also do not offer general solution strategies that can be applied to problem formulations where, different type of variables, objective and constraint functions are used. For example, simplex algorithm can be used to solve models with linear objective and constraint functions; geometric programming can be used to solve non-linear models with a posynomial or signomial objective function etc. (Baykasoğlu, 2001). However, most of the optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation. Therefore, classic optimization procedures are generally not adequate or easy to use for their solution.

Researchers have spent a great deal of effort in order to adapt many optimization problems to the classic optimization procedures. It is generally not easy to formulate a real life problem that suits a specific solution procedure. In order to achieve this, it is necessary to make some modifications and/or assumptions on the original problem parameters (rounding variables, softening constraints etc.). This certainly affects the solution quality. A new set of problem and model independent nature inspired heuristic optimization algorithms were proposed by researchers to overcome drawbacks of the classical optimization procedures. These techniques are efficient and flexible. They can be modified and/or adapted to suit specific problem requirements (see Figure 1). Research on these techniques is still continuing all around the globe.
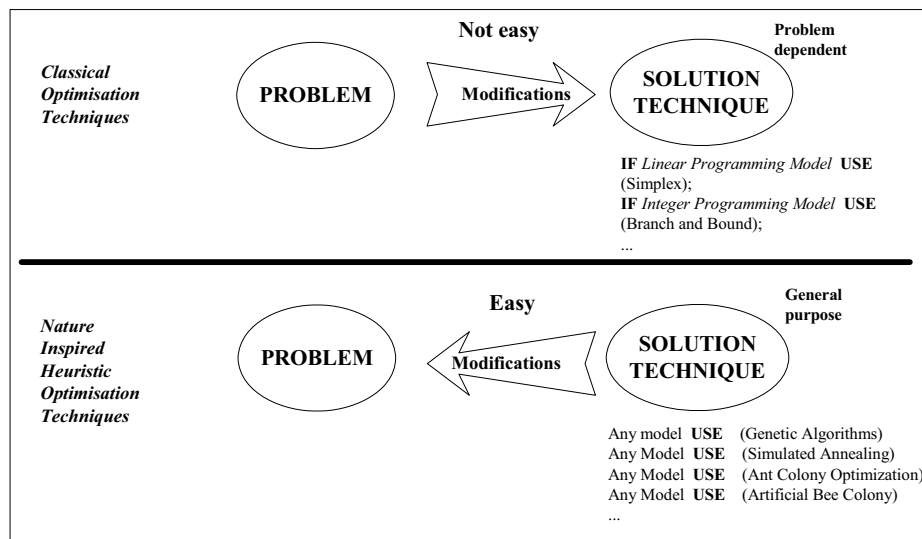

Figure 1. A pictorial comparison of classical and modern heuristic optimisation strategies (Adapted from Baykasoğlu, 2001)

A branch of nature inspired algorithms which are called as swarm intelligence is focused on insect behaviour in order to develop some meta-heuristics which can mimic insect's problem solution abilities. Interaction between insects contributes to the collective intelligence of the social insect colonies. These communication systems between insects have been adapted to

scientific problems for optimization. One of the examples of such interactive behaviour is the waggle dance of bees during the food procuring. By performing this dance, successful foragers share the information about the direction and distance to patches of flower and the amount of nectar within this flower with their hive mates. So this is a successful mechanism which foragers can recruit other bees in their colony to productive locations to collect various resources. Bee colony can quickly and precisely adjust its searching pattern in time and space according to changing nectar sources.

The information exchange among individual insects is the most important part of the collective knowledge. Communication among bees about the quality of food sources is being achieved in the dancing area by performing waggle dance (Figure 2).
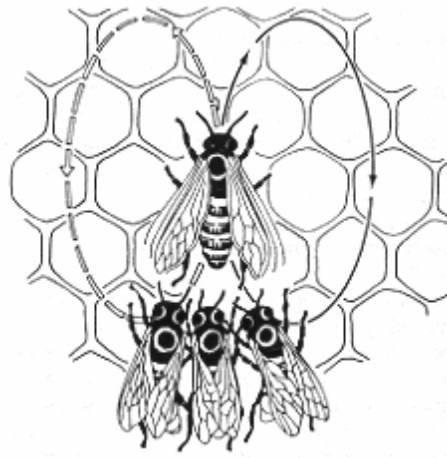

Figure 2. Waggle dance of honey bees

The previous studies on dancing behaviour of bees show that while performing the waggle dance, the direction of bees indicates the direction of the food source in relation to the Sun, the intensity of the waggles indicates how far away it is and the duration of the dance indicates the amount of nectar on related food source. Waggle dancing bees that have been in the hive for an extended time adjust the angles of their dances to accommodate the changing direction of the sun. Therefore bees that follow the waggle run of the dance are still correctly led to the food source even though its angle relative to the sun has changed. So collective intelligence of bees based on the synergistic information exchange during waggle dance.

Observations and studies on honey bee behaviours resulted in a new generation of optimization algorithms. In this chapter a detailed review of bee colony based algorithms is given. Afterwards a bee based algorithm that we name as "artificial bee colony" is explained in detail along with an application to "generalized assignment problem" which is known as a NP-hard problem.

2. Description of the Behaviour of Bees in Nature

Social insect colonies can be considered as dynamical system gathering information from environment and adjusting its behaviour in accordance to it. While gathering information

and adjustment processes, individual insects don't perform all the tasks because of their specializations. Generally, all social insect colonies behave according to their own division labours related to their morphology. Bee system consists of two essential components:

- **Food Sources**

  The value of a food source depends on different parameters such as its proximity to the nest, richness of energy and ease of extracting this energy.

- **Foragers**

  - *Unemployed foragers:* If it is assumed that a bee have no knowledge about the food sources in the search field, bee initializes its search as an unemployed forager. There are two possibilities for an unemployed forager:

    - *Scout Bee (*S in Figure 3): If the bee starts searching spontaneously without any knowledge, it will be a scout bee. The percentage of scout bees varies from 5% to 30% according to the information into the nest. The mean number of scouts averaged over conditions is about 10% (Seeley, 1995).

    - *Recruit* (R in Figure 3): If the unemployed forager attends to a waggle dance done by some other bee, the bee will start searching by using the knowledge from waggle dance.

  - *Employed foragers* (EF in Figure 3): When the recruit bee finds and exploits the food source, it will raise to be an employed forager who memorizes the location of the food source. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive. There are three possible options related to residual amount of nectar for the foraging bee.

    - If the nectar amount decreased to a low level or exhausted, foraging bee abandons the food source and become an unemployed bee.

    - If there are still sufficient amount of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates

    - Or it can go to the dance area to perform waggle dance for informing the nest mates about the same food source. The probability values for these options highly related to the quality of the food source.

  - *Experienced foragers:* These types of forager use their historical memories for the location and quality of food sources.

    - It can be an inspector which controls the recent status of food source already discovered.

    - It can be a reactivated forager by using the information from waggle dance. It tries to explore the same food source discovered by itself if there are some other bees confirm the quality of same food source (RF in FigureS).

    - It can be scout bee to search new patches if the whole food source is exhausted (ES in Figure 3).

    - It can be a recruit bee which is searching a new food source declared in dancing area by another employed bee (ER in Figure 3).
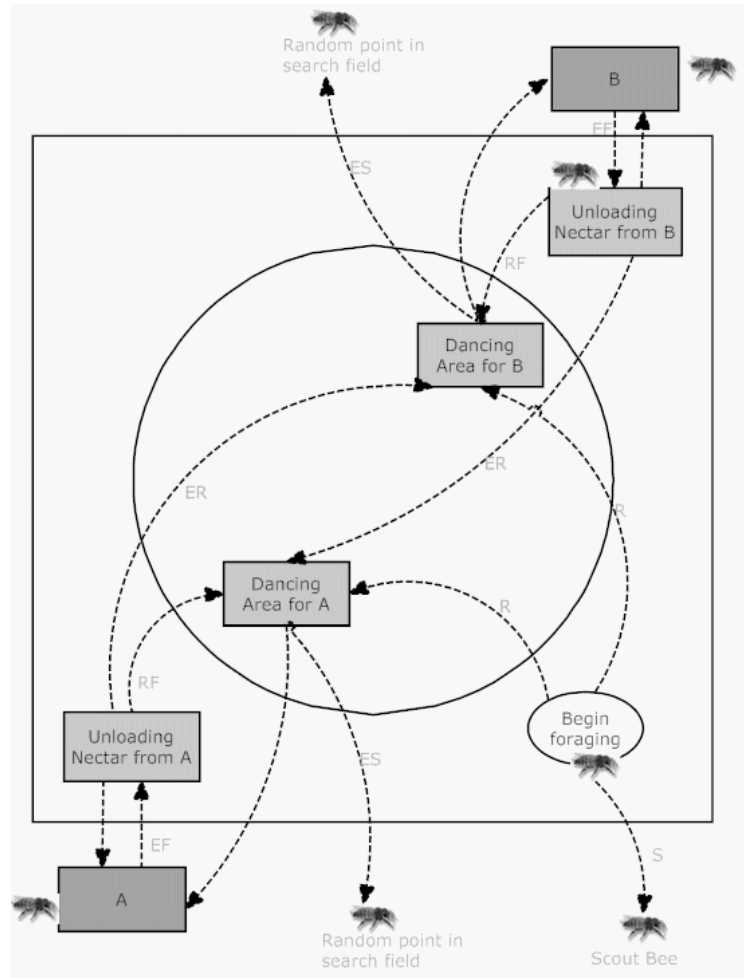
Figure 3. Typical behaviour of honey bee foraging

3. Review and Categorization of Studies on Artificial Bee Systems

The foraging behaviour, learning, memorizing and information sharing characteristics of bees have recently been one of the most interesting research areas in swarm intelligence. Studies on honey bees are in an increasing trend in the literature during the last few years. After a detailed literature survey, the previous algorithms are categorized in this work by concerning the behavioural characteristics of honey bees.

- Foraging behaviours
- Marriage behaviours
- Queen bee concept

The researches, their main contributions and applications are summarized as shown in Table 1.

Type	Honey Bee Literature	Algorithm	Application
Foraging Behaviour	Yonezawa and Kikuchi (1996) Seeley and Buhrman (1999) Schmickl et al. (2005) Lemmens (2006)		*Biological simulation*
	Sato and Hagiwara (1997)	Bee System	*Genetic Algorithm Improvement*
	Karaboga (2005)	Artificial Bee Colony Algorithm (ABC)	*Continuous Optimization*
	Yang (2005)	Virtual Bee Algorithm (VBA)	*Continuous Optimization*
	Basturk and Karaboga (2006)	ABC	*Continuous Optimization*
	Pham et al. (2006a)	Bees Algorithm (BA)	*Continuous Optimization*
	Lucic and Teodorovic (2001)	Bee System (BS)	*Travelling Salesman Problem(TSP)*
	Lucic (2002)	BS	*TSP and Stochastic Vehicle Routing Problem*
	Lucic and Teodorovic (2002)	BS	*TSP*
	Lucic and Teodorovic (2003a)	BS	*TSP*
	Lucic and Teodorovic (2003b)	BS + Fuzzy Logic	*Stochastic Vehicle Routing*
	Teodorovic and Dell'Orco (2005)	Bee Colony Optimization (BCO) + Fuzzy Bee System (FBS)	*Ride-Matching Problem*
	Nakrani and Tovey (2003)	A Honey Bee Algorithm	*Dynamic Allocation of Internet Service*
	Wedde et al. (2004)	BeeHive	*Telecommunication Network Routing*
	Bianco (2004)		*Large Scale Precise Navigation*
	Chong et al. (2006)		*Job Shop Scheduling*
	Drias et al. (2005)	Bees Swarm	*Max-W-Sat Problem*
	Pham et al. (2006b)	BA	*LVQ-Neural Network*
	Pham et al. (2006c)	BA	*MLP- Neural Network*
	Pham et al. (2006d)	BA	*Neural Network*
	Quijano and Passino (2007)		*Dynamic Resource*
	Markovic et al. (2007)	BCO Based	*Max-Routing and Wavelength Assignment*

Marriage Behaviour	Abbass (2001a,b,c)	Marriage in Honey-Bees Optimization (MBO)	*3-Sat Problem*
	Teo and Abbass (2001, 2003)	Modified MBO	*3-Sat Problem*
	Bozorg Haddad and Af shar (2004)	MBO	*Water Resources Management Problems*
	Bozorg Haddad et al. (2006)	Honey-Bees Mating Optimization -HBMO	*Nonlinear constrained and unconstrained optimization*
	Chang (2006)	MBO Based	*Stochastic Dynamic Programming*
	Afshar et al. (2007)	Improved HBMO	*Continuous Optimization*
	Fathian et al. (2007)	HBMO Based	*Data Mining -Clustering*
	Koudil et al. (2007)	MBO Based	*Integrated Partitioning/Scheduling*
	Benatchba et al. (2005)	MBO Based	*Data Mining*
Queen Bee	Sung (2003)	Queen-Bee Evolution Algorithm(QBE)	*Genetic Algorithm Improvement*
	Qin et al. (2004)	QBE Based	*Economic Power Dispatch*
	Kara (2004)	Bee Crossover	*Genetic Algorithm Improvement*
	Azeem and Saad (2004)	Modified QBE	*Genetic Algorithm Improvement*

Table 1. Categorization of literature and applications

In this section, contributions of these researches are explained in detailed to clarify the background of honey bees based optimization algorithms.

Yonezawa and Kikuchi (1996) examine the foraging behaviour of honey bees and construct an algorithm to indicate the importance of group intelligence principals. The algorithm is simulated with one and three foraging bees and the computational simulation results showed that three foraging bees are faster than the system with one foraging bee at decision making process. They also indicate that the honey bees have an adaptive foraging behaviour at complex environment.

Seeley and Buhrman (1999) investigated the nest site selection behaviour of honey bee colonies. The nest site selection process starts with several hundred scout bees that search for potential nest sites. The scouts then return to the cluster, report their findings by means of waggle dances, and decide the new nest site. The type of waggle dance depends on the quality of the site being advertised. The authors repeated the observations of Lindauer in 1955 by taking advantage of modern video-recording and bee-labelling techniques on three honey bee colonies. Many of the results confirmed with the previous study and some of the results provided new and important insights. They pointed out that a colony's strategy of decision making is a weighted additive strategy which is the most accurate but most

information demanding one. This strategy evaluates each alternative according to the relative attributes, gives weights to each attribute according to its importance, sums the weighted attributes for each alternative, and finally chooses the alternative whose total valuation is the highest. Similarly, the bee colony considers a dozen or more alternative nest sites, evaluates each alternative nest site with respect to at least six distinct attributes with different weightings e.g. cavity volume, entrance height, entrance area, entrance direction etc. Consequently, the bee colony uses this strategy by distributing among many bees both the task of evaluating the alternative sites and the task of identifying the best of these sites.

Schmickl et al. (2005) evaluate the robustness of bees' foraging behaviour by using a multi-agent simulation platform. They investigate how the time-pattern of environmental fluctuations affects the foraging strategy and the efficiency of the foraging. They conclude that the collective foraging strategy of a honeybee colony is robust and adaptive, and that its emergent features allow the colony to find optimal solutions.

Lemmens (2006) investigated whether pheromone-based navigational algorithms (inspired by biological ant colony behaviour) are outperformed by non-pheromone-based navigational algorithms (inspired by biological bee colony behaviour) in the task of foraging. The results of the experiments showed that (i) pheromone-based navigational algorithms use less time per iteration step in small-sized worlds, (ii) non-pheromone-based algorithms are significantly faster when finding and collecting food and use fewer time steps to complete the task, and (iii) with growing world sizes, the non-pheromone-based algorithm eventually outperforms pheromone-based algorithms on a time per time step measure. In spite of all these profits it is mentioned that non-pheromone-based algorithms are less adaptive than pheromone-based algorithms.

Sato and Hagiwara (1997) proposed an improved genetic algorithm based on foraging behaviour of honey bees. In a honey bee colony, each bee looks for the feed individually. When a bee finds feed, then it notifies the information to the other many bees by dance and they engage in a job to carry the feed. When they finish the work, each bee tries to find new one individually again. Similarly in the proposed algorithm, named Bee System, global search is done first, and some chromosomes with pretty high fitness (superior chromosomes) are obtained using the simple genetic algorithm. Second, many chromosomes obtain the information of superior chromosomes by the concentrated crossover and they search intensively around there using multiple populations. In the conventional crossover each pair is made randomly, while in the concentrated crossover all of the chromosomes make pair with superior chromosome. Lastly, pseudo-simplex method is contributed to enhance the local search ability of the Bee System. If the solution found by one cycle is not satisfactory, the global search is repeated. As it is known genetic algorithms have good global search ability, however they lack the local search ability. On the other hand with Bee System probability of falling into a local optimum is low because of the combination of local and global search since the aim of the algorithm is to improve the local search ability of genetic algorithm without degrading the global search ability. In the experimental studies Bee System is compared with the conventional genetic algorithm and it is found that Bee System shows better performance than the conventional genetic algorithm especially for highly complex multivariate functions.

Karaboga (2005) analyzes the foraging behaviour of honey bee swarm and proposes a new algorithm simulating this behaviour for solving multi-dimensional and multi-modal optimization problems, called Artificial Bee Colony (ABC). The main steps of the algorithm are: 1) send the employed bees onto the food sources and determine their nectar amounts; 2) calculate

the probability value of the sources with which they are preferred by the onlooker bees; 3) stop the exploitation process of the sources abandoned by the bees; 4) send the scouts into the search area for discovering new food sources, randomly; 5) memorize the best food source found so far. In the algorithm, an artificial bee colony consists of three groups of bees: employed bees, onlookers and scouts. Employed bees are associated with a particular food source which they are currently exploiting. They carry the information about this particular source and share this information with a certain probability by waggle dance. Unemployed bees seek a food source to exploit. There are two types of unemployed bees: scouts and onlookers. Scouts search the environment for new food sources without any guidance. Occasionally, the scouts can accidentally discover rich, entirely unknown food sources. On the other hand onlookers observe the waggle dance and so are placed on the food sources by using a probability based selection process. As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases, too. In the ABC algorithm the first half of the colony consists of the employed bees and the second half includes the onlookers. For every food source, there is only one employed bee. Another issue that is considered in the algorithm is that the employed bee whose food source has been exhausted by the bees becomes a scout. In other words, if a solution representing a food source is not improved by a predetermined number of trials, then the food source is abandoned by its employed bee and the employed bee is converted to a scout. The algorithm is tested on three well known test functions. From the simulation results, it is concluded that the proposed algorithm can be used for solving uni-modal and multi-modal numerical optimization problems.

Yang (2005) presents a virtual bee algorithm (VBA) which is effective on function optimization problems. The main steps of the algorithm are: 1) create an initial population of virtual bees where each bee is associated with a memory; 2) encode the optimization function into virtual food; 3) define the criterion for communicating food location with others; 4) march all the virtual bees randomly to new positions for virtual food searching, find food and communicate with neighbouring bees by virtual waggle dance; 5) evaluate the encoded intensity /locations of bees; 6) decode the results to obtain the solution to the problem. However the proposed algorithm is similar with genetic algorithm, it is much more efficient due to the parallelism of the multiple independent bees. To realize this statement, the VBA algorithm is tested on two functions with two parameters, one is single-peaked and the other is multi-peaked. The results show that the new algorithm is much efficient than genetic algorithm.

Basturk and Karaboga (2006) presented another ABC algorithm and expanded the experimental results of Karaboga (2005). The performance of the algorithm is tested on five multi-dimensional benchmark functions and the results were compared with genetic algorithms. It is pointed out that ABC algorithm outperforms genetic algorithm for functions having multi-modality and uni-modality.

Pham et al. (2006a) proposed an optimization algorithm inspired by the natural foraging behaviour of honey bees, called Bees Algorithm. The proposed algorithm is also applicable to both combinatorial and functional optimization problems. In real life, foraging process begins by scout bees being sent to search for promising flower patches. When they return to the hive, unload their nectar and go to the dance floor to perform a dance known as the waggle dance which is essential for colony communication. After waggle dancing, the dancer goes back to the flower patch with follower bees that were waiting inside the hive. More follower bees are sent to more promising patches. This allows the colony to gather food quickly and efficiently.

Similarly Bees Algorithm starts with scout bees being placed randomly on the search space. The main steps of the algorithm are: 1) initialize population with random solutions; 2) evaluate fitness of the population; 3) determine a certain number of fittest bees and select their sites for neighbourhood search; 4) recruit a certain number of bees for selected sites, evaluate their fitness; 5) select the fittest bee from each site to form the new population; 6) assign remaining bees to search randomly and evaluate their fitness. The Bees Algorithm is applied to two standard functional optimization problems with two and six dimensions, respectively. The results showed that the Bees Algorithm is able to find solutions very close to the optimum. The algorithm is also applied to eight benchmark functions and the results were compared with deterministic simplex method, stochastic simulated annealing optimization procedure, genetic algorithm and ant colony system. Bees Algorithm generally outperformed other techniques in terms of speed of optimization and accuracy of results. On the other hand Bees Algorithm has too many tuneable parameters.

Luck and Teodorovic (2001) published the first study on Bee System based on the PhD thesis of Luck for 6 Travelling Salesman Problem (TSP) test problems. Luck (2002) aimed to explore the possible applications of collective bee intelligence in solving complex traffic and transportation engineering problems. In this context, (TSP) and stochastic vehicle routing problem (SVRP) were studied. TSP is an NP-hard problem that aims to find the minimum distance circuit passing through each node only once. The algorithm starts with locating the hive in one of the nodes on the graph that the bees are collecting nectar i.e. the graph in which the travelling salesman route should be discovered. The artificial bees collect the nectar during a certain prescribed time interval and the position of the hive is randomly changed. The bees start to collect the nectar from the new location and again the location of the hive is randomly changed. The iteration in the searching process represents one change of the hive's position and the iteration ends when one or more feasible solution is created. The artificial bees live in an environment characterized by discrete time and consequently each iteration is composed of a certain number of stages. During any stage, bees choose nodes to be visited in a random manner. By this probability function it is provided that the greater the distance between two nodes, the lower the probability that a bee chooses this link. The influence of the distance is lower at the beginning of the search process. The greater the number of iterations, the higher the influence of the distance. On the other hand the greater the total number of bees that visited by certain link in the past, the higher the probability is of choosing that link in the future. This represents the interaction between individual bees in the colony. During one stage the bee visits a certain number of nodes, create a partial travelling salesman tour, and return to the hive. In the hive the bee participates in a decision making process. The bee decides whether to recruit the nest mates by dancing before returning to the food source, to continue to forage at the food source without recruiting the nest mates, or to abandon the food source. The second alternative has very low probability since bees are social insects and communicate each other. The probability that a bee uses the same partial tour (or abandons it) depends on the length of the partial tour. The longer the tour that the bee has discovered, the smaller is the probability that the bee will fly again along the same tour. It is noted that the nectar quantity along a certain link is inversely proportional to the link length. At the beginning of any stage if a bee does not use the same partial travelling salesman tour, the bee goes to the dancing area and follows another bee according to a probability function. This function depends on the total length of the partial route and the number of bees that are advertising this route.

Additionally, before relocating the hive, 2-opt and 3-opt heuristic algorithms are applied to improve the solution obtained by the bees in the current iteration. On the other hand in nature, not all bees start foraging simultaneously and in the algorithm it is assumed that at the beginning of each iteration all bees are in the hive and the number of foraging bees in every subsequent stage is increased. The performance of the algorithm was tested on 10 benchmark problems. Experimental results showed that in all instances with less than 100 nodes, the Bee System produced the optimal solution and the times required to find the best solutions by the Bee System were low. At the second part of this thesis Bee System was integrated with fuzzy logic and this approach was applied to Vehicle Routing Problems (VRP). The procedure and the results were presented at Luck and Teodorovic (2003b). Luck and Teodorovic (2002, 2003a) published their second and third study on Bee System based on Luck's (2002) 8 and 10 TSP test problems.

Luck and Teodorovic (2003b) combined Bee System algorithm, which was first proposed by Luck and Teodorovic (2001), and fuzzy logic approach to obtain good solutions for stochastic VRP. The proposed approach contains two steps: 1) solve VRP as a TSP by using Bee System and obtain frequently an infeasible solution to the original problem; 2) decide when to finish one vehicle's route and when to start with the next vehicle's route by using the solution created at the previous step and fuzzy rule base generated by Wang-Mendel's algorithm. Stochastic VRP is to find a set of routes that would minimize transportation cost where the locations of the depot, nodes to be served and vehicle capacity are known, and demand at the nodes only approximated. Due to the uncertainty of demand at the nodes, a vehicle might not be able to service a node once it arrives there due to insufficient capacity. It is assumed in such situations that the vehicle returns to the depot, empties what it has picked up thus far, returns to the node where it had a failure, and continues service along the rest of the planned route. Consequently, demand at nodes is treated as a random variable and actual demand value is known only after the visit to the node. The developed model was tested on 10 TSP examples. In order to convert the original TSP problems into the corresponding VRPs, the first node was treated as a depot. The results were compared with the best solution obtained by the heuristic algorithm based on Bee System. The results were found to be very close to the best solution assuming that the future node demand pattern was known.

Teodorovic and Dell'Orco (2005) proposed Bee Colony Optimization (BCO) meta-heuristic which was the generalization of the Bee System presented by Luck (2002). The BCO was capable to solve deterministic combinatorial problems, as well as combinatorial problems characterized by uncertainty. The primary goal of their paper was to explore the possible applications of collective bee intelligence in solving combinatorial problems characterized by uncertainty. In this respect Fuzzy Bee System (FBS) was introduced where the agents use approximate reasoning and rules of fuzzy logic in their communication and acting. The performance of FBS algorithm was tested on ride-matching problem which aims to constitute routing and scheduling of the vehicles and passengers by minimizing the total distance travelled by all participants, minimizing the total delay, or equalizing vehicle utilization. There were no theoretical results that could support proposed approach but preliminary results were very promising.

Nakrani and Tovey (2003) proposed a honey bee algorithm for dynamic allocation of internet services. In the proposed algorithm, servers and HTTP request queues in an Internet server colony were modelled as foraging bees and flower patches respectively. The

algorithm was compared with an omniscient algorithm that computes an optimal allocation policy, a greedy algorithm that uses past history to compute allocation policy, and an optimal-static algorithm that computes omnisciently the best among all possible static allocation policies. The experimental results showed that the algorithm performs better than static or greedy algorithms. On the other hand it was outperformed by greedy algorithm for some low variability access patterns.

Wedde et al. (2004) introduced a fault-tolerant, adaptive and robust routing protocol inspired from dance language and foraging behaviour of honey bees for routing in telecommunication network, called BeeHive. In order to evaluate the performance of the algorithm, it was tested on Japanese Internet Backbone and compared with AntNet, DGA and OSPF. The results showed that BeeHive achieves a similar or better performance as compared to the other algorithms.

Bianco (2004) presented a mapping paradigm for large scale precise navigation that takes inspiration from the bees' large scale navigation behaviour. Bees performed very long navigations when they feed, travelling for many kilometres but, at the same time, getting an excellent precision when they return to their small hives. Test results demonstrated that such capabilities were sufficient to get rather good precision.

Chong et al. (2006) presented a novel approach that uses the honey bees foraging model, inspired by Nakrani and Tovey (2004), to solve the job shop scheduling problem. Job shop scheduling is concerned with finding a sequential allocation of competing resources that optimizes a particular objective function. Each machine can process only one job and each job can be processed by only one machine at a time. The performance of the algorithm was tested on 82 job shop problem instances and compared with ant colony and tabu search algorithms. The experimental results conducted that tabu search outperforms other two heuristics according to solution quality and execution time. On the other hand bee algorithm performed slightly better than ant algorithm and the execution time for both heuristics was approximately equal.

Drias et al. (2005) introduced a new intelligent approach named Bees Swarm Optimization (BSO), which is inspired from the behaviour of real bees especially harvesting the nectar of the easiest sources of access while always privileging the richest. The proposed algorithm was adapted to the maximum weighted satisfiability problem (MAX-W-SAT) problem which was NP-Complete. MAX-W-SAT problem asks for the maximum weight which can be satisfied by any assignment, given a set of weighted clauses. The performance of the algorithm was compared with GRASP, SSAT and AGO and it was concluded that BSO outperformed the other evolutionary algorithms.

Pham et al. (2006b) presented the use of the Bees Algorithm, proposed by Pham et al. (2006a) to train the Learning Vector Quantization (LVQ) neural network for control chart pattern recognition. The training of a LVQ network can be regarded as the minimization of an error function. The error function defines the total difference between the actual output and the desired output of the network over a set of training patterns. In terms of the Bees Algorithm, each bee represents a LVQ network with a particular set of reference vectors. The aim of the algorithm was to find the bee with the set of reference vectors producing the smallest value of the error function. Despite the high dimensionality of the problem, the algorithm still succeeded to train more accurate classifiers than that produced by the standard LVQ training algorithm.

Pham et al. (2006c) presented the use of the Bees Algorithm, proposed by Pham et al. (2006a) to train the Multi-layered Perceptron (MLP) neural network for control chart pattern recognition. The training of a MLP network can be regarded as the minimization of an error function. The error function defines the total difference between the actual output and the desired output of the network over a set of training patterns. In terms of the Bees Algorithm, each bee represents a MLP network with a particular set of weight vectors. The aim of the algorithm was to find the bee with the set of weight vectors producing the smallest value of the error function. Despite the high dimensionality of the problem, the algorithm succeeded to train more accurate classifiers than back propagation algorithm.

Pham et al. (2006d) presented an application of the Bees Algorithm, proposed by Pham et al. (2006a) to the optimization of neural networks for the identification of defects in wood veneer sheets. The Bees Algorithm was used instead of a back propagation algorithm to optimize the weights of the neural network. The optimization using the Bees Algorithm involves the bees searching for the optimal values of the weights assigned to the connections between the neurons within the network where each bee represents a neural network with a particular set of weights. The aim of the Bees Algorithm was to find the bee producing the smallest value of the error function. The experimental results show that the Bees Algorithm was able to achieve an accuracy that was comparable to the back propagation method. However, the Bees Algorithm proved to be considerably faster.

Quijano and Passino (2007) developed an algorithm, based on the foraging behaviour of honey bees, to solve resource allocation problem. The primary sources for constructing components of the proposed model were: dance strength determination, dance threshold, unloading area, dance floor and recruitment rates, explorer allocation and its relation to recruitment. They also proposed an engineering application on dynamic resource allocation for multi-zone temperature control, to highlight the main features of the dynamical operation of the honey bee social foraging algorithm.

Markovic et al. (2007) used the BCO algorithm, which was introduced by Teodorovic and Dell'orco (2005) to solve Max-Routing and Wavelength Assignment (Max-RWA) problem in all-optical networks. The Max-RWA problem is to maximize the number of established lightpaths in a given optical network for a given traffic demand matrix and the given number of wavelengths. The proposed BCO-RWA algorithm was tested on European Optical Network and the results were compared with the results obtained by the LP relaxation approach and the tabu meta-heuristic algorithm. The BCO-RWA algorithm always outperformed the results of the compared algorithms and was able to produce very good solutions in a reasonable computation time.

Abbass (2001a) presented the first novel search algorithm inspired by the marriage process in honey bees. A honey bee colony consists of the queen(s), drones, worker(s), and broods. In this study the colony was assumed to have a single queen and a single worker. In real life a mating flight starts with a dance performed by the queens and the drones follow the queens to mate with them. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony. Each time a queen lays fertilized eggs, she retrieves at random a mixture of the sperms accumulated in the spermatheca to fertilize the egg. Similarly at the MBO algorithm the mating flight can be visualized as a set of transitions in a state space where the queen moves between the different states in the space and mate with the drone encountered at each state probabilistically. The probability of mating is high when either the queen is still in the start of her mating flight and therefore

her speed is high, or when the fitness of the drone is as good as the queen's. The algorithm starts with initializing the queen's genotype at random. After that a heuristic is used to improve the queen's genotype realized by workers. Afterwards, a set of mating flights is undertaken. In each mating flight, the queen's energy and speed are initialized randomly. The queen then moves between different states (solutions) in the space according to her speed and mates with the drone. If a drone is successfully mated with the queen (the drone passes the probabilistic decision rule), its sperm is added to the queen's spermatheca (list of partial solutions) and the queen's speed and energy are reduced. After the queen finishes her mating flight, she returns to the nest, selects a sperm randomly, performs crossover and mutation. The worker is then used to improve the resultant brood and the number of workers represents the number of heuristics encoded in the program. Afterwards, the queen is replaced with the fittest brood if the latter is better than the former. The remaining broods are then killed and a new mating flight starts. The MBO algorithm has three user-defined parameters: the queen's spermatheca size representing the maximum number of matings in a single mating flight, the number of broods that will be born by the queen, and the amount of time devoted to brood care signifying the depth of local search. A general constraint satisfaction problem (CSP) is the problem of finding an assignment to a set of variables that satisfies a set of constraints over the domains of those variables. The prepositional satisfiability problems (SAT) is a special case of CSP where the domain of each variable is either true or false. Also 3-SAT is a special case of SAT where each constraint contains three variables. The MBO algorithm was applied to a hundred different 3-SAT problems and the experimental results conducted that the algorithm was very successful. The heuristics that workers use was Greedy SAT (GSAT) and random walk. At the experimental studies GSAT, random walk, MBO with GSAT and MBO with random walk were compared and MBO-GSAT performed the best among the other three.

Abbass (2001b) presented a variation of the MBO algorithm which was first proposed by Abbass (2001a) where the colony contains a single queen with multiple workers. For the workers six different heuristics were used: GSAT, random walk, random flip, random new, 1-point and 2-point crossover. The algorithm was tested on a group of one-hundred hard 3-SAT problems. The best results were occurred with the smallest colony size and average spermatheca size. On the other hand, the fittest worker was GSAT, which was followed by random walk. It was also showed that MBO performed better than GSAT alone although GSAT was the heuristic with the highest fitness in MBO.

Abbass (2001c) analyzed the marriage behaviour of honey bees again as the continuation of the work (Abbass, 2001a). The difference between these studies was the number of queens and workers. Abbass (2001c) considered the honey bee colony with more than one queen in addition to a group of workers, where at the colony of Abbass (2001a) there was only one queen and one worker. In the paper MBO algorithm was applied to fifty different 3-SAT problems containing 50 variables and 215 constraints. The experimental results concluded that the largest spermatheca size, an average colony size, and the smallest number of queens gave the best performance. On the other hand the algorithm was compared with WalkSAT, one of the state-of-the-art algorithms for SAT, and MBO algorithm outperformed WalkSAT.

Teo and Abbass (2001) presented a modification of MBO algorithm which can be considered as an extension of Abbass (2001a) and Abbass (2001c). The purpose of this modification was to use a more conventional annealing approach during the trajectory acceptance decision to guide the search process towards a more optimal solution space. New trajectories were only

be accepted as a potential drone for mating if it was a more optimal trajectory that was if the trajectory was fitter than the queen's genotype. Otherwise, if it was a trajectory that takes the search to a less optimal solution space, then it is only accepted probabilistically subject to the new annealing function. In other words, rather than accepting all the trajectories created during a queen's flight as in the original MBO algorithm, a new trajectory is accepted only if it is a move to a fitter solution space. Otherwise, the algorithm will accept a transition to a less optimal solution space probabilistically according to a function of the queen's fitness. On the other hand, five different heuristics were used for improving broods by workers: GSAT, random walk, probabilistic greedy, one point crossover, and WalkSAT. As in Abbass (2001a) Teo and Abbass again considered the honey bee colony with only one queen. Experimental studies were conducted in three manner: testing each of five different heuristics working alone without MBO, testing the performance of each heuristic with the original MBO and modified MBO, and lastly testing the proposed algorithm against the original MBO using the five different heuristics operating in combination as a committee of heuristics. For the test problems, ten different 3-SAT problems were generated each comprising of 1075 constraints and 250 variables. The heuristic performance's resulted with the following order for the first group of experiments: WalkSAT, GSAT, random walk, probabilistic greedy and one point crossover. At the second group of experiments both the original and proposed annealing functions used during the mating flight process were similarly efficient with all heuristics. However, the effectiveness of MBO with WalkSAT in finding solutions was improved slightly by the new annealing function as the proposed version of MBO found more solutions than the original version. Lastly at the third group of experiments both annealing strategies were again similarly efficient.

Teo and Abbass (2003) proposed another modification of MBO algorithm based on Teo and Abbass (2001). In both Abbass (2001a) and Teo and Abbass (2001), the annealing function used the queen's fitness as the basis for accepting/rejecting a transition in the drone's space, either during the spawning or mating stage. In a conventional simulated annealing approach, the previous state was used as the basis for the transition. Moreover, from a biological point of view, the drone's creation is independent of the queen as they usually come from another colony, although they might be related. Therefore, it is more natural to accept a transition based on the drone's own fitness. As a result the objective of their paper was to test a purely conventional annealing approach as the basis for determining the pool of drones. The performance of the modified algorithm was tested on ten different 3-SAT problems and compared with the previous versions of MBO. All heuristics were failed to find even a single solution when working alone whereas their performances were improved significantly when combined with MBO. On the other hand the proposed version of MBO dominated the previous studies and able to find solutions for problems where the previous versions cannot.

Bozorg Haddad and Afshar (2004) benefited from MBO algorithm based on the study of Abbass (2001c) and performed an application to water resources management problems. The algorithm was modelled to find good solutions for optimum management of a single reservoir. The results compared very well with similar heuristic methods as well as global optimal results.

Bozorg Haddad et al. (2006) proposed Honey-Bees Mating Optimization (HBMO) algorithm, based on Abbass (2001a, 2001c), to solve highly non-linear constrained and unconstrained real valued mathematical models. The performance of the HBMO was tested on several

constrained and unconstrained mathematical optimization functions and compared with the results obtained by genetic algorithm. Results from the genetic algorithm and HBMO algorithm converge well with minor improvement in the HBMO solution. Moreover, to illustrate the model application and performance, the HBMO algorithm was also used for developing an optimum operation policy for a single reservoir. The HBMO again generated a significantly better solution.

Chang (2006) gave the first demonstration of the capability of the MBO approach in a theoretical perspective for solving combinatorial optimization problems and stochastic sequential decision making problems. The paper first concerned with MBO algorithm for solving non-stochastic combinatorial optimization problems and proved that MBO has the ability to converge to the global optimum value. MBO was then adapted into an algorithm called "Honey-Bees Policy Iteration" (HBPI) for solving infinite horizon discounted cost stochastic dynamic programming (SDP) problems, also known as markov decision processes (MDPs) and HBPI algorithm was also proved converging to the optimal value. Chang (2006) points out that MBO can be considered as a hybrid scheme of simulated annealing and genetic algorithm. Simulated annealing corresponds to the queen's mating flight to obtain the potential drone sperms in her spermatheca and genetic algorithm corresponds to broods generation and improvements step with some differences.

Afshar et al. (2007) presented an improved version of the HBMO algorithm for continuous optimization problems and its application to a nonlinear-constrained continuous single reservoir problem. By the comparison with global optimum values obtained from LINGO 8.0 NLP solver, it was observed that the convergence of the algorithm to the optimum was very rapid.

Fathian et al. (2007) presented an application of HBMO algorithm for clustering which is one of the attractive data mining techniques that is in use in many fields. To evaluate the performance of the algorithm in clustering, it was tested on several real datasets and compared with several typical stochastic algorithms including the ACO algorithm, the simulated annealing approach, the genetic algorithms, and the tabu search approach. The results illustrated that the proposed HBMO approach can be considered as a viable and an efficient heuristic to find optimal or near optimal solutions to clustering problems since the results were very encouraging in terms of the quality of solutions found, the average number of function evaluations and the processing time required.

Koudil et al. (2007) adapted MBO algorithm which was first presented by Abbass (2001) to solve integrated partitioning/scheduling problem in codesign. The proposed approach was tested on a benchmark problem and the results were compared with genetic algorithm. The test results showed that MBO achieves good results in terms of solution quality, and it gives better results than genetic algorithm in terms of execution times.

Benatchba et al. (2005) used the MBO algorithm which was first presented by Abbass (2001a, 2001b, 2001c) to solve a data mining problem expressed as a Max-Sat problem. For MBO, four different heuristics were used for improving broods by workers: a local search algorithm LS, GSAT, HSAT, and GWSAT. The training set used as benchmark was extracted from a medical one, aiming at analyzing the most revealing symptoms of the presence or not of a laparotomy of the principal bile duct. The best result obtained with MBO was the solution with 96% satisfaction by using GSAT as a worker.

Sung (2003) proposed queen-bee evolution to enhance the capability of genetic algorithms. In the queen-bee evolution algorithm the queen-bee crossbreeds with the other bees selected

as parents by a different selection algorithm instead of known selection algorithms such as roulette wheel selection. This procedure increases the exploitation of genetic algorithms but on the other hand increases the probability of falling into premature convergence. To decrease this probability some individuals were strongly mutated instead of mutating all individuals with small mutation probability as in the normal evolution. The proposed algorithm was tested with one combinational and two typical function optimization problems. Experimental results demonstrated that the proposed algorithm enabled genetic algorithms to quickly approach to the global optimum.

Qin et al. (2004) applied queen bee evolution which was proposed by Sung (2003) into economic power dispatch problem (EPD). EPD problem is to minimize the overall cost rate and meet the load demand of a power system simultaneously and formulated as a nonlinear constrained complex optimization problem. The numerical results demonstrated that the proposed algorithm was faster and more robust than the conventional genetic algorithm.

Kara (2004) proposed a new crossover type, which is called Bee Crossover to improve the genetic algorithm's performance. The bee queen has the sexual intercourses with other male bees, and similarly a specified chromosome can be considered as bee queen for the first parent of crossover and the other parent is one of the remaining chromosomes in the colony. The author proposed three different crossover types. At the first type, the chromosome with the best fitness value is fixed parent and all the remaining chromosomes are crossed over with this fixed parent at least once in each generation. At the second type, the chromosome with the worst fitness value is a fixed parent and the remaining procedure is the same with the first type. At the third type, population is sorted with respect to the fitness values and the fixed parent in the first generation is determined by the first chromosome in this list. In the second generation, the fixed parent is the second chromosome in the list and so on. The performance of these crossover types were compared with uniform crossover. The results showed that in the most of time, honey bee crossovers obtained results in less number of iterations and the worst results were obtained by uniforms crossover. On the other hand, uniform crossover lost the diversity of population in a small range of time while honey bee crossovers lost the population diversity in the larger ranges of time.

Azeem and Saad (2004) proposed a modified queen bee evolution which was first presented by Sung (2003). In the proposed algorithm, if any solution has the fitness very close or above of the fitness of the queen bee, this solution is identified to a new pool as a queen bee where the original algorithm is limited to a single pool. Another difference between the original and proposed algorithm is on the crossover operator. The original algorithm utilizes uniform crossover where each gene is crossed with some probability. On the other hand proposed algorithm uses weighted uniform crossover where weights are assigned to each gene according to the similarity of the test patterns in the population. With this type of crossover, genetic algorithm will search more new state spaces. The algorithm was tested for tuning of scaling factor for the Fuzzy Knowledge Base Controller (FKBC) on two complex non-linear examples. Experiments showed that FKBC yielded superior results than conventional control algorithms in the complex situations where the system model or parameters were difficult to obtain. Moreover, the results were compared with roulettes wheel parent selection and obtained results were encouraging.

In the following sections of this work the first application of a nature inspired bee based algorithm (that we name as Artificial Bee Colony, ABC) to generalized assignment problem is presented.

4. Generalized Assignment Problem

The Generalized Assignment Problem (GAP) aims that assigning a set of tasks to a set of agents with minimum cost. Each agent represents a single resource with limited capacity. Each task must be assigned to only one agent and it requires a certain amount of the resource of the agent.

There are many application domains of GAP such as computer and communication networks, location problems, vehicle routing, group technology, scheduling etc. Extended review of this problem and its possible applications is presented in Martello and Toth (1981, 1990), Cattrysse (1990) and Cattrysse et al. (1994). Several exact algorithms for GAP have been proposed by Ross and Soland (1975), Fisher et al. (1986), Martello and Toth (1990) and recently Savelsberg (1997) and Nauss (2003). Also several heuristics have been proposed to solve GAP. Martello and Toth (1981, 1990) proposed a combination of local search and greedy method. Osman (1995) developed new Simulated Annealing and Tabu Search algorithms to investigate their performance on GAP. Chu and Beasley (1997) presented a Genetic Algorithm for GAP that tries to improve feasibility and optimality simultaneously. Different variable depth search algorithms (Racer and Amini (1994), Yagiura et al. (1998, 1999)), Ejection Chain based Tabu Search algorithms (Laguna et al. (1995), Diaz and Fernandez (2001), Yagiura et al. (2004)), Path Relinking approaches (Alfandari et al. (2001, 2002, 2004), Yagiura et al. (2001, 2002, 2006)) , Ant Colony Optimization (Randall (2004)), Max-Min Ant System Heuristic based on greedy randomized adaptive heuristic (Lourencp and Serra (2002)) can be mentioned as the other meta-heuristic approaches proposed for GAP in recent years.

The aim of this study is to present an artificial bee colony algorithm to solve GAP. Our main interest on this problem came from its NP-hard structure that was proved by Fisher et al. (1986). Moreover, Mortello and Toth (1990) presented the NP-completeness of proving that a solution is a feasible solution. GAP can be formulated as an integer programming model as follows;

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} x_{ij}$$

subject to

$$\sum_{i=1}^{n} a_{ij} x_{ij} \leq b_j \qquad \forall j, \quad 1 \leq j \leq m$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \forall i, \quad 1 \leq i \leq n$$

$$x_{ij} \in \{0,1\} \qquad 1 \leq i \leq n \quad \forall i, \quad 1 \leq j \leq m \quad \forall j$$

I is set of tasks (*i*=1,..,*n*); *J* is set of agents (*j*= 1,..,*m*); b_j = resource capacity of agent *j*; a_{ij} = resource needed if task *i* is assigned to agent *j* ; c_{ij} = cost of task *i* if assigned to agent *j*
x_{ij} = decision variable (x_{ij}=1, if task *i* is assigned to agent *j*; 0, otherwise)
The first constraint set is related to the resource capacity of agents. The second constraint set ensure that each task is assigned to only one agent.

5. Artificial Bee Colony Algorithm for GAP

In this section the general ABC framework and the principal algorithms for the initial solution and neighbour solutions generation for GAP are presented. The general steps of the proposed ABC algorithm are presented in Table 2.

1. *Initialize parameters*
2. *Construct initial Employed Bee Colony solutions by using*
   Greedy Randomized Adaptive Search Heuristic (GRAH)
3. *Evaluate fitness value for each bee*
4. *I=0*
5. *Repeat*
6. *N=0*
7. *repeat*
   a. *Apply Shift neighbourhood*
   b. *Apply DoubleShift neighbourhood*
   c. *Calculate probabilities related to fitness values*
   d. *Assign Onlooker Bees to Employed Bees according to probabilities*
   e. *For all Onlooker Bees*
      i. *Ejection -Chain Neighbourhood*
   f. *Find best Onlooker, replace with respective Employed Bee*
      iffit(Best Onlooker)<fit(Employed)
   g. *Find best Feasible Onlooker, replace with Best solution,*
      if fit (BestFeas Onlooker)<fit(Best)
   h. *N=N+1*
8. *Until (N=Employed Bee)*
9. *I=I+1*
10. *Until (I=MaxIteration)*

Table 2. ABC algorithm for GAP

Each step of the general ABC algorithm is detailed in Table 3.

0. *Parameter Initialization*
   n = Number of employed bees
   m = Number of onlooker bees $(m>n)$
   Iteration : Maximum iteration number
   α_j : initial value of penalty parameter for j^{th} agent
   EC-Length : Length of ejection chain neighbourhood
1. *Initialize employed bees with GRAH algorithm*
   σ^i : i^{th} employed bee in the population
2. *Evaluate employed bees*
   Fitness Function (for minimization)

$$\sum_{j=1}^{m}\sum_{i=1}^{n}c_{ij}x_{ij} + \alpha\sum_{j=1}^{m}\max\left\{0, \sum_{i=1}^{n}b_{ij}x_{ij} - a_j\right\}$$

3. *Repeat*
   Cycle = 1
   1. *Number of Scout bees = 0,1*n*

2. *For each Employed Bee*
     a. *Apply SHIFT Neighbourhood*
         i. *If fit(ShiftNeighbour)<fit(EmployedBee) then*
             1. *Employed Bee = Shift Neighbour*
     b. *Apply DOUBLESHIFT Neighbourhood*
         i. *If fit(DoubleShiftNeighbour)<fit(EmployedBee) then*
             1. *Employed Bee = DoubleShift Neighbour*
     c. *Determine probabilities by using fitness function*

$$p_i = \frac{\sum \left(\frac{1}{fit_i} \right)^{-1}}{fit_i} \quad \textit{(for minimization)}$$

     d. *Calculate the number of onlooker bees which will be sent to food sources of employed bees, according to previously determined probabilities*
     e. *N_i= Number of onlooker bees sent to i^{th} sites = p_i*m*
     f. *O_{ij}: j^{th} onlooker bee of ith solution (j=1,…,N_i)*
         $\{O_{i1}, O_{i2},…,O_{iNi}\}$=EjectionChain($\sigma^i$)
     g. *Calculate fitness values for each onlooker bee*
         If the best fitness value of onlooker bees is better than the fitness value of employed bee, employed bee solution is replaced with this onlooker solution.

         If (min (fit(O_{ij}))<fit(σ^i) then σ^i = O_{ij}

3. *Best Solution*

         If fit($Best_{Cycle-1}$)> Min(Fit(σ^i))$_{i=1..n}$ then $Best_{Cycle}$= σ^i
         Else $Best_{Cycle}$= $Best_{Cycle-1}$

Until (i=n)
4. *Scout bees*
     a. *Initialize scout bees with GRAH algorithm*
     b. *The worst employed bees as many as the number of scout bees in the population are respectively compared with the scout solutions. If the scout solution is better than employed solution, employed solution is replaced with scout solution. Else employed solution is transferred to the next cycle without any change.*
5. *Cycle = Cycle+1*
Until (cycle =Iteration)

Table 3. Detailed ABC algorithm for GAP

Initial bee colony is constructed by using GRAH algorithm (Lourenço and Serra, 2001). The greedy heuristic constructs a solution as follows:

- At each step, a next task to be assigned is selected.
- The agent (the selected task is going to be assigned to) is determined.
- Repeat these two steps until all tasks have been assigned to an agent.

In GRAH procedure the choice is probabilistic bias to a probability function. This function is updated at each iteration in a reinforcement way by using the features of good solutions. The main execution steps of the GRAH algorithm is summarized as shown in Table 4.

1. *Let* $S_j = \emptyset$ $\forall j = 1,...,m$ *(S_j is the set of task assigned to agent j)*

2. *Construct a list of agents for each task, L_i, initially Li={1,…,m} $\forall i$.*

3. *Consider any order of the tasks, i=1.*

4. *While (not all tasks have been assigned) repeat*
   4.1 *Choose randomly an agent j^* from L_i following the probability function that depends on the resource of agent j and the resource need by task i:*

   $$p_{ij} = \frac{a/b_{ij}}{\sum_{l \in L_i} a_l/b_{il}}, \quad j \in L_i$$

   The agent with minimal cost has greater probability to be chosen.

   4.2 *Assign task i to agent j^* : $S_{j^*} = S_{j^*} \cup \{i\}$. Let i=i+1 and if*

   $$\sum_{i \in S_{j^*}} b_{ij^*} > a_{j^*}$$ *remove j^* from any list. Repeat step 4 (Note that the capacity constraint can be violated).*

5. *Let* $\sigma(i) = j$ *if* $i \in S_j$

Table 4. The GRAH Algorithm

6. Neighbourhood Structures

Shift Neighbourhood: This type of neighbour is obtained from original solution by changing the agent assignment of one task. The algorithm steps are summarized in Table 5. An example implementation of this algorithm is portrayed in Figure 4.

Double Shift Neighbourhood: This neighbourhood structure is the special case of the long chain neighbourhood. Since the two shift moves are performed in double shift, this is the (EC-Length=2) state of the long chain. Double shift neighbourhood contains the swap neighbourhood, which is the interchange of agents of two different tasks assigned to, within its scope. In the long chain neighbourhood, task for each shift move is selected from B list. In double shift neighbourhood, new shift move is determined by using the set of all tasks. Because, there is no restriction to achieve a new shift move. A simplified demonstration of the neighbourhood is shown in Figure 5.

Shift (σ)

1. *Let $S = \{i \mid i \in \{1...n\}\}$, k=1, ShiftNeighbour= σ*

2. *If $S = \emptyset$ then stop; otherwise i_k is ejected from σ_k . $S = S - \{i_k\}$*

3. *Let j^* be the agent j that minimizes*

   $$c_{ikj} + \alpha_j \max\{0, \left(\sum_{i \in l, \sigma(i)=j} a_{ij}\right) + a_{ikj} - b_j\} \quad \text{among all agents } j \in J \setminus \{\sigma(i_k)\} .$$

4. *Assign i_k to j^*, output $\sigma^{'}$, calculate Fitness($\sigma^{'}$)*

5. *If Fitness($\sigma^{'}$)<Fitness (ShiftNeighbor) then ShiftNeighbour = $\sigma^{'}$*

6. *k:= k+1, return to Step 2*

7. *Output ShiftNeighbour*
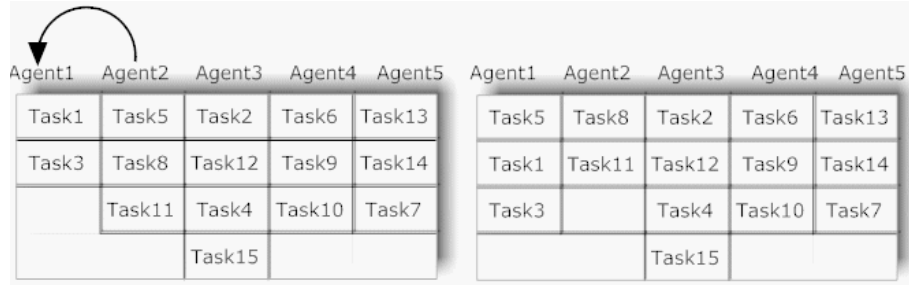
Table 5. Shift neighbourhood
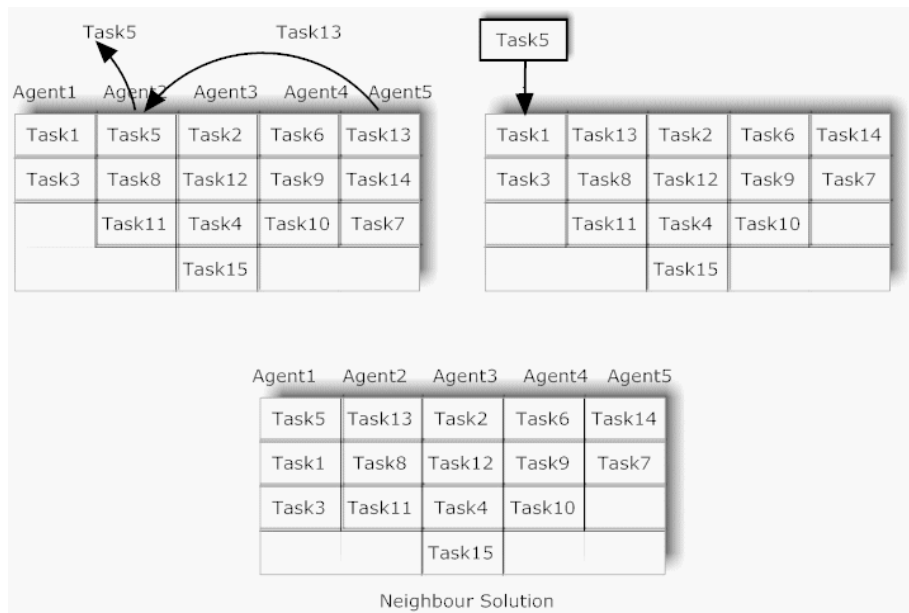
Figure 4. Shift neighbourhood structure

Figure 5. Double shift neighbourhood structure

Long Chain Neighbourhood: A neighbour is obtained by performing the multiple shift moves whose length is specified as chain length. A simple explanation of the neighbourhood structure and main steps of the algorithm are presented here, but detailed information can be obtained from Yagiura et al. (2004). Assume that task i_0 is ejected from agent $\sigma(i_0)$ as a free task where $\sigma(i_0)$ denotes the agent that task i_0 is assigned to. The amount of resource of $\sigma(i_0)$ is increased by this ejection move. Avail is defined as the resulting amount of resource as shown in the following equation.

$$avail(i) = \begin{cases} a_{i,\sigma(i)} - p_{\sigma(i)}(\sigma) & if \quad a_{i,\sigma(i)} > p_{\sigma(i)}(\sigma) \\ a_{i,\sigma(i)} & otherwise \end{cases}$$

Assuming that task i_1 is the task whose shift into $\sigma(i_0)$ is most profitable among the tasks satisfying $a_{i1,\sigma(i_0)} \leq avail(i_0)$ Task i_1 is shifted into agent $\sigma(i_0)$. This is called as reference

structure. After this first ejection move, the free task i_0 will be tried to assign into some other agents according to its effect on fitness function as shown in Table 6 (Step 4). This is called as the trial move. The next ejection move is applied to the previous reference structure, not to the solutions generated by the trial moves. Same steps are repeated until the stopping criterion is satisfied. The general mechanism of the long chain neighbourhood is presented in Table 6 and portrayed in Figure 6.

Long Chain (σ)

1. *Let $S := \emptyset$.*

2. *If $S = I'$, stop; otherwise randomly choose a $i_0 \in I \setminus S'$,*

   Let $S := S \cup \{i_0\}$, and $\sigma' := \sigma$. (Job i_0 is ejected from $\sigma(i_0)$.)

3. *Let j^* be the agent j that minimizes $c_{i_0 j} + \alpha_j \max\{0, \left(\sum_{i \in l, \sigma(i)=j} a_{ij} \right) + a_{i_{01} j} - b_j\}$*

   among all agents
   $j \in J / \{\sigma(i_0)\}$, and let $l := 0$.

4. *If $B(i_l) \setminus \{i_k | k \le l\} = \emptyset$, return to Step 2; otherwise let $l := l+1$ and proceed to Step 5.*

5. *Randomly choose $i_l \in B(i_{l-1}) \setminus \{i_k | k \le l-1\}$ and let $\sigma'(i_l) := \sigma(i_{l-1})$ (an ejection move of job i_l).*
   Then execute the following Steps (a) and (b) (two trial moves).

   (a) *Let $\sigma'(i_0) := \sigma(i_l)$ (i_0 is inserted into $\sigma(i_l)$), and output σ' .*

   (b) *Let $\sigma'(i_0) := j^*$ (i_0 is inserted into i^*), and output σ' .*

6. *Return to Step 4.*

Table 6. Long chain neighbourhood structure

$$p_j(\sigma) = \max\left\{ 0, \left(\sum_{i \in l, \sigma(i)=j} a_{ij} \right) - b_j \right\}$$

$$score(i, j) = -c_{ij} + c_{i,\sigma(i)}$$

$$I' = \left\{ k \in I \middle| \exists h \in I \ s.t. \ a_{h,\sigma(k)} \le avail(k) \ and \ \sigma(h) \ne \sigma(k) \right\}$$

$$bestscore(i) = \min\left\{ score(k, \sigma(i)) \middle| k \in l \quad \sigma(k) \ne \sigma(i) \ and \ a_{k,\sigma(i)} \le avail(i) \right\}$$

$$B(i) = \left\{ k \in I' \middle| score(k, \sigma(i)) = bestscore(i), \quad \sigma(k) \ne \sigma(i) \ and \ a_{k,\sigma(i)} \le avail(i) \right\}$$

As shown in Figure 6, Task 5 is selected as the free task and removed from Agent 2. After avail(Task5) is updated, Task 13 is determined for the shift move which has the best score among the other tasks satisfying avail(Task5). Task 13 removed from Agent 5 and assigned to Agent 2. This is the reference structure for neighbourhood. In the next step, a trial move

to assign the free task Task 5 to an agent is determined according to the assignment effect on fitness function value. Assuming that Agent 1 is determined, Task 5 is assigned to that agent to complete the trial move. The result of a trial move is the complete neighbour for the original solution. This neighbour is obtained by applying *l*=2 (two shift moves) which is also called as double shift neighbourhood. If the length of ejection chain > 2, then the same steps are repeated on the previous reference structure, which is in the case of the free task is not assigned to an agent. In Figure 6, avail(Task!3) is updated and Task 2 is determined for the next shift move. After Task 2 is assigned to Agent 5, a new trial move is performed to assign ejection task. Assuming that the most profitable agent is Agent 2, Task 5 is assigned to Agent 2 to obtain a complete neighbour solution. This is the long chain with length 3 (three shift moves).   Same steps are repeated to complete the previously determined length of ejection.
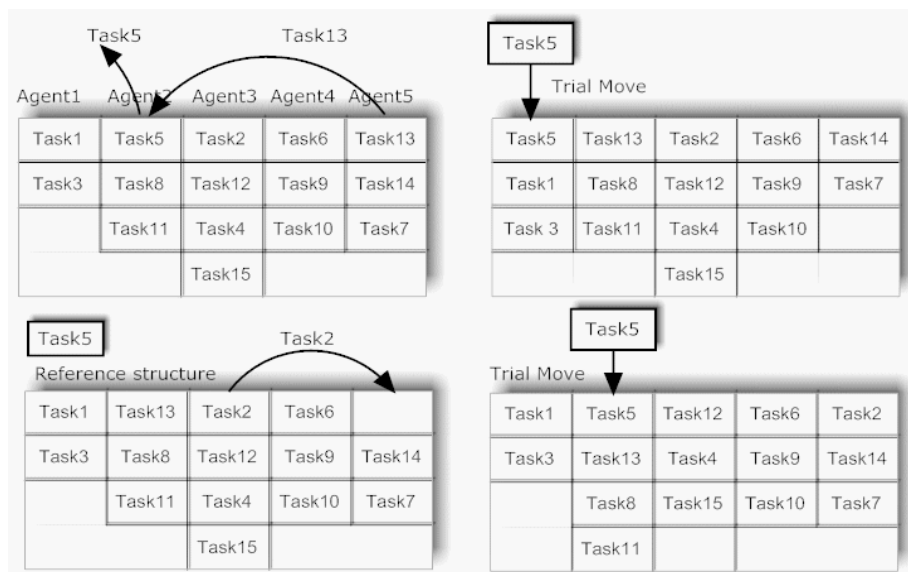

Figure 6. Long chain neighbourhood structure (*l*=3)

7. Computational Study

The proposed ABC algorithm is coded in C# and tested in a set of problems ranging from 5 agents-15 tasks to 10 agents-60 tasks. These test problems are publicly available from the www.OR-Library.com. The set of test problems can be divided into two groups: Gapl-Gap6/easy and Gap7-Gapl2/difficult. Each problem set consists of 5 different problems with the same size, so there are 12*5 =60 problems to solve. These set of problems are of maximization form of GAP and optimal values are known.

In this section, a simple GAP example is designed to explain the execution of one iteration of the proposed ABC algorithm. An example consisting of 3 agent and 6 task assignment problem is implemented to solve as a minimization problem. A bee solution is represented as an array of tasks which contains the assignment of agents.

There are 3 Employed Bees and 5 Onlooker Bees in the example. Initial solutions of bee colony are generated by using the GRAH algorithm. A shift neighbourhood structure is

applied to each Employed Bee. For Bee 1, after shift neighbourhood, a better solution is obtained by changing the assignment of Task 1 from Agent 2 to Agent 1 and original Bee 1 solution is changed to neighbour solution as shown in Figure 7. In the second step, double shift neighbourhood is applied to new Bee 1. Since there is no solution better than Bee 1, employed bee solution is not changed by this neighbourhood. Shift and Double Shift steps are repeated for Bee 2 and Bee 3. After these steps, a transient bee colony is constituted to determine the probabilities. These probabilities are calculated by using the equation in Table 3 (3.2.c) to determine the number of onlooker bees assigned to each employed bee. As shown in Figure 7, the worst bee (Bee 3) retains the minimum number of onlooker bee. For each employed bee, ejection chain neighbourhood is applied and the quantity of neighbours generated is determined according to the number of onlooker assigned to employed bee. The fitness value of onlooker bees are compared with the original employed bee fitness and the best onlooker is selected as the winner. Updated bee colony for the next iteration is shown in Figure 7. In addition to this updating stage, the best feasible solution among the bee colony is compared to the best solution found so far. If the employed bee is better than the best, the best solution is updated.

Experimental Setup for GAP Problems

Parameters of proposed algorithm are defined as follows;

- Number of employed bees (n)
- Number of onlooker bees (m>n)
- Number of scout bees (0.1*n)
- Maximum iteration number (Iteration)
- Initial value of penalty coefficient (a_j)
- Length of ejection chain neighbourhood (EC-Length)

Penalty function is used to calculate the fitness function. While constructing initial solutions by using the GRAH algorithm and generating neighbours by using shift, double shift and ejection chain algorithms, proposed approach allows producing infeasible solutions. Consequently, there is an additional term in the objective function determined by penalizing the infeasible solutions with a_j coefficient ($a_j > 0$). Fitness function is computed by using the following equation.

$$fit(\sigma_i) = \sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij} x_{ij} + \left(\sum_{j=1}^{m} \alpha_j \max\left\{ 0, \sum_{i=1}^{n} b_{ij} x_{ij} - a_j \right\} \right)$$

The first term in the equation denotes the total cost of assignment of tasks to agents. The second term is defined as an additional penalty function for minimization, a_j represents the cost of using one unit of overloaded capacity of jth agent. Initial values of a_j's are determined as user defined parameter. If a solution is not feasible the second term will be positive and therefore the search will be directed to feasible solution. If the capacity is not exceeded, this term will be 0 to ensure not penalized. The parameter a_j can be increased during the run to penalize infeasible solutions and drive the search to feasible ones which means the adaptive control of penalty costs.

Initial values of a_j's are designed as user defined ($a_j > 0$). Updating stage is adapted from Yagiura et al. (2004) by using the following equations. After the generation of onlooker neighbours of each employed bee a_j values are updated.
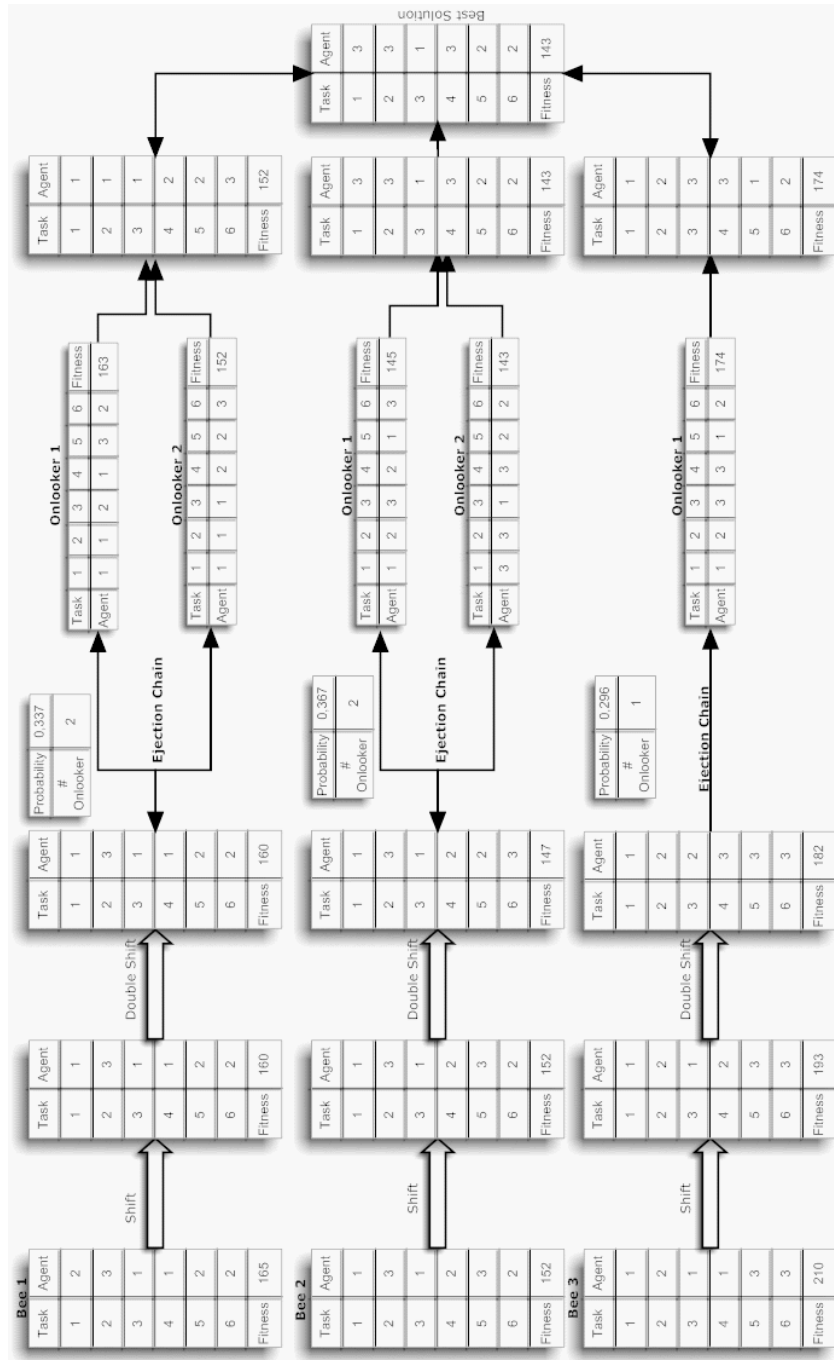
Figure 7. A sample execution of neighbourhood structure

1. If there is no feasible solution found in onlooker neighbours, a_j are increased for all $j \in J$ by

$$\alpha_j = \begin{cases} \alpha_j(1 + \Delta q_j^{inc}(\sigma)), & \alpha_j > 0 \\ \Delta q_j^{inc}(\sigma) \min_{h \in J} \{b_h \alpha_h | b_h \alpha_h > 0\} / b_j, & othervise \end{cases}$$

Where

$$\Delta = \begin{cases} \dfrac{stepsizeinc}{\max_{j \in J} |q_j^{inc}(\sigma)|}, & if \quad \max_{j \in J} |q_j^{inc}(\sigma)| > 0 \\ 0, & othervise \end{cases}$$

2. (Otherwise) If at least one feasible solution found within onlooker neighbours, all a_j are decreased by using the same equations except that $q^{dec}(\sigma)$ instead of $q^{inc}(\sigma)$ and *stepsizedec* instead of *stepsizeinc*.

$$q_j^{inc}(\sigma) = p_j(\sigma) / b_j$$

$$q_j^{dec}(\sigma) = \begin{cases} -1, & if \quad p_j(\sigma) = 0 \\ 0, & othervise \end{cases}$$

Parameter Setting

Parameters of algorithm are treated as two different sets. As mentioned before Gapl to Gap6 problem sets are specified as easy while Gap7 to Gapl2 as difficult. Accordingly, two different parameter sets are determined as shown in Table 7.

Parameter	Gapl-Gap6	Gap7-Gapl2
# of Iteration	100	250
# of Employed Bee	50	50
# of Onlooker Bee	100	500
# of Scout Bee	5	5
α	1	1
EC-Length	5	10

Table 7. Parameter setting

Five runs for each problem are evaluated. Different algorithms that solved Gapl-Gapl2 in the literature are determined for comparison. The values in Table 8 represent the mean deviation from the optimal value for each problem set. Proposed algorithm found the optimal solutions in all five runs for all problem sets with previously defined parameters. As compared to other 12 algorithms the proposed algorithm is unambiguously the best performer.

	ABC	*MTH*	*FJVBB*	FSA	*MTBB*	SPH	*LT1FA*	RSSA	*TS6*	TS1	*GAk*	GA,	ASH+LS +TS
Gapl	0.00	5.43	0.00	0.00	0.00	0.08	1.74	0.00	0.00	0.00	0.00	0.00	-
Gap2	0.00	5.02	0.00	0.19	0.00	0.11	0.89	0.00	0.24	0.10	0.00	0.01	-
Gap3	0.00	2.14	0.00	0.00	0.00	0.09	1.26	0.00	0.03	0.00	0.00	0.01	-
Gap4	0.00	2.35	0.83	0.06	0.18	0.04	0.72	0.00	0.03	0.03	0.00	0.03	-
Gap5	0.00	2.63	0.07	0.11	0.00	0.35	1.42	0.00	0.04	0.00	0.00	0.10	-
Gap6	0.00	1.67	0.58	0.85	0.52	0.15	0.82	0.05	0.00	0.03	0.01	0.08	-
Gap7	0.00	2.02	1.58	0.99	1.32	0.00	1.22	0.02	0.02	0.00	0.00	0.08	0.00
Gap8	0.00	2.45	2.48	0.41	1.32	0.23	1.13	0.10	0.14	0.09	0.05	0.33	0.042
Gap9	0.00	2.18	0.61	1.46	1.06	0.12	1.48	0.08	0.06	0.06	0.00	0.17	0.00
Gap10	0.00	1.75	1.29	1.72	1.15	0.25	1.19	0.14	0.15	0.08	0.04	0.27	0.013
Gap11	0.00	1.78	1.32	1.10	2.01	0.00	1.17	0.05	0.02	0.02	0.00	0.20	0.00
Gap12	0.00	1.37	1.37	1.68	1.55	0.10	0.81	0.11	0.07	0.04	0.01	0.17	0.00

ABC: The Proposed Algorithm, MTH: Martello and Toth (1981) constructive heuristic, FJVBB: Fisher et al. (1986) branch and bound procedure with and upper CPU limit, FSA: Cattrysse (1990) fixing simulated annealing algorithm, MTBB: Martello and Toth (1991) branch and bound procedure with an upper CPU limit, SPH: Cattrysse et al. (1994) set partitioning heuristic, LT1FA: Osman (1995) long term descent, 1-interchange mechanism and first admissible, RSSA: Osman (1995) hybrid SA/TS with different seed values, TS6: Osman (1995) long term TS, BA selection, Rl tabu restrictions and Al aspiration criterion, TS1: Osman (1995) long term TS, FA selection, Rl tabu restrictions and Al aspiration criterion, GAi>: Chu and Beasley (1997) genetic algorithm with heuristic operator, GAa: Chu and Beasley (1997) genetic algorithm without heuristic algorithm

Table 8. Comparison of results

8. Conclusion

In this study a relatively new member of swarm intelligence family that is named as "artificial bee colony" is explained in detail. Actually, different names were used in the literature for the algorithms inspired from natural honey bees. Here we prefer to use the name "artificial bee colony" to reflect population characteristic of the algorithm. A very detailed literature review along with a categorization is presented in this study. All accessible previous work on bee based optimization algorithms is tried to be reviewed. Most of the work in the literature is carried out in last two years and researchers mainly concentrated on continuous optimization and TSP problems. Previous work has presented that bee inspired algorithms have a very promising potential for modelling and solving complex optimization problems. But there is still a long way to go in order to fully utilise the potential of bee inspired algorithms. Such an attempt is also made in this study to present performance of a bee inspired algorithm, "artificial bee colony" on a NP-hard problem which is known as generalised assignment problem. The proposed bee algorithm is found very effective in solving small to medium sized generalized assignment problems. Actually, the proposed algorithm easily found all optimal solutions where the compared 12 algorithms were not able to find for most of the cases. Our research is still under progress and we are hoping to find effective solutions for large size and tightly constrained generalised assignment problems. These problems are over complex, therefore their solution can be considered as a very good indicator for the potential of the nature inspired algorithms including "artificial bee colony".

9. Acknowledgements

10. References

Abbass H.A., A Single Queen Single Worker Honey-Bees Approach to 3-SAT, *GECCO2001 The Genetic and Evolutionary Computation Conference*, San Francisco, USA, 2001a.

Abbass H.A., A Monogenous MBO Approach to Satisfiability, *CIMCA'2001 International Conference on Computational Intelligence for Modeling, Control and Automation*, Las Vegas, NV, USA, 2001b.

Abbass H.A., MBO: Marriage in Honey Bees Optimization A Haplometrosis Polygynous Swarming Approach, *CEC2001 Proceedings of the Congress on Evolutionary Computation*, Seoul, Korea, 207-214,2001c.

Afshar A., Bozorg Haddad O., Marino M.A., Adams B.J., Honey-Bee Mating Optimization (HBMO) Algorithm for Optimal Reservoir Operation, *Journal of the Franklin Institute*, In press.

Alfandari L., Plateau A., Tolla P., A Two-Phase Path Relinking Algorithm for the Generalized Assignment Problem, In *Proceedings of the Fourth Metaheuristics International Conference*, Porto, Portugal, 175-179,2001.

Alfandari L., Plateau A., Tolla P., A Two-Phase Path Relinking Algorithm for the Generalized Assignment Problem, *Technical Report* No: 378, CEDRIC, CNAM, 2002.

Alfandari L., Plateau A., Tolla P., A Path Relinking Algorithm for the Generalized Assignment Problem, In M.G.C. Resende, J.P. de Sousa (Eds.), *Metaheuristics: Computer Decision-Making*, Kluwer Academic Publishers, Boston, 1-17,2004

Azeem M.F., Saad A.M., Modified Queen Bee Evolution Based Genetic Algorithm for Tuning of Scaling Factors of Fuzzy Knowledge Base Controller, *IEEE INDICON 2004 Proceedings of the India Annual Conference*, 299-303,2004.

Basturk B., Karaboga D., An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization, *IEEE Swarm Intelligence Symposium 2006*, Indianapolis, Indiana, USA, 2006.

Baykasoğlu A., Goal Programming using the Multiple Objective Tabu Search, *Journal of Operational Research Society*, 52(12), 1359-1369,2001.

Benatchba K., Admane L., Koudil M., Using Bees to Solve a Data Mining Problem Expressed as a Max-Sat One, In *Proceedings of IWINAC'2005, International Work Conference on the Interplay between Natural and Artificial Computation*, Canary Islands, Spain, 212-220,2005.

Bianco G.M., Getting Inspired from Bees to Perform Large Scale Visual Precise Navigation, *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 619-624,2004.

Bozorg Haddad O., Afshar A., MBO Algorithm, A New Heuristic Approach in Hydrosystems Design and Operation, *1st International Conference on Managing Rivers in the 21st Century*, 499-504,2004.

Bozorg Haddad O., Afshar A., Mariano M.A., Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization, *Water Resources Management*, 20, 661-680,2006.

Cattrysse D., Set Partitioning Approaches to Combinatorial Optimization Problems, *Ph.D. Thesis*, Katholieke Universiteit Leuven, Centrum Industrieel Beleid, Belgium, 1990.

Cattrysse D., Salomon M., Van Wassenhove L.N., A Set Partitioning Heuristic for the Generalized Assignment Problem, *European Journal of Operational Research*, 72,167-174.1994.

Chang H.S., Converging Marriage in Honey-Bees Optimization and Application to Stochastic Dynamic Programming, *Journal of Global Optimization*, 35,423-441,2006.

Chong C.S., Low M.Y.H., Sivakumar A.I., Gay K.L., A Bee Colony Optimization Algorithm to Job Shop Scheduling, *Proceedings of the 37th Winter Simulation*, Monterey, California, 1954-1961,2006.

Chu P.C., Beasley J.E., A Genetic Algorithm for the Generalized Assignment Problem, *Computers Operations Research*, 24,17-23,1997.

Diaz J.A., Fernandez E., A Tabu Search Heuristic for the Generalized Assignment Problem, *European Journal Operational Research*, 132,22-38,2001.

Drias H., Sadeg S., Yahi S., Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem, *IWAAN International Work Conference on Artificial and Natural Neural Networks*, Barcelona, Spain, 318-325,2005.

Fathian M., Amiri B., Maroosi A., *Application of Honey-Bee Mating Optimization Algorithm on Clustering*, In press.

Fisher M.L., Jaikumar R., Van Wassenhove L.N., A Multiplier Adjustment Method for the Generalized Assignment Problem, *Management Science*, 32,1095-1103,1986.

Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization, *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, 2005.

Kara A., Imitation of Bee Reproduction as a Crossover Operator in Genetic Algorithms, *PRICAI 2004*, LNAI 3157, C. Zhang, H.W. Guesgen, W.K. Yeap (Eds.), Springer-Verlag, Berlin Heidelberg, 1015-1016,2004.

Koudil M., Benatchba K., Tarabet A., Sahraoui E.B., Using Artificial Bees to Solve Partitioning and Scheduling Problems in Codesign, *Applied Mathematics and Computation*, 186(2), 1710-1722,2007.

Laguna M., Kelly J.P., Gonzalez-Velarde J.L., Glover P., Tabu Search for the Multilevel Generalized Assignment Problem, *European Journal of Operational Research*, 82,176-189.1995.

Lemmens N.P.P.M., To Bee or Not to Bee: A Comparative Study in Swarm Intelligence, *Master Thesis*, Maastricht University, Maastricht ICT Competence Centre, Institute for Knowledge and Agent Technology, Maastricht, Netherlands, 2006.

Lourengo H.R., Serra D., Adaptive Search Heuristics for the Generalized Assignment Problem, *Mathware and Soft Computing*, 9,209-234,2002.

Lucic P., Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing, *PhD Thesis*, Civil Engineering, Faculty of the Virginia Polytechnic Institute and State University, 2002.

Lucic P., Teodorovic D., Bee system: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence, *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis*, Sao Miguel, Azores Islands, 441-445,2001.

Luckic P., Teodorovic D., Transportation Modeling: An Artificial Life Approach, *ICTAI'02 14th IEEE International Conference on Tools with Artificial Intelligence*, 216-223,2002.

Lucic P., Teodorovic D., Computing with Bees: Attacking Complex Transportation Engineering Problems, *International Journal on Artificial Intelligence Tools*, 12(3), 375-394,2003a.

Lucic P., Teodorovic D., Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach, in *Fuzzy Sets in Optimization*, Editor J.L. Verdegay, Springer-Verlag, Berlin Heidelbelg, 67-82,2003b.

Markovic G.Z., Teodorovic D., Acimovic-Raspopovic V.S., Routing and Wavelength Assignment in All-Optical Networks Based on the Bee Colony Optimization, *AI Communications - The European Journal on Artificial Intelligence*, (in press).

Martello S., Toth P., An Algorithm for the Generalized Assignment Problems, *Operational Research*, ed. J.P. Brans. North-Holland, 589-603,1981.

Martello S., Toth P., *Knapsack Problems: Algorithms and Computer Implementations*, Wiley, New York, 1990.

Nakrani S., Tovey C, On Honey Bees and Dynamic Allocation in an Internet Server Colony, *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects*, Atlanta, Georgia, USA, 2003.

Nauss R.M., Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach, *INFORMS Journal of Computing*, 15,249-266,2003.

Osman I.H., Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches, *OR Spektrum*, 17,211-225,1995.

Pham D.T., Kog E., Ghanbarzadeh A., Otri S., Rahim S., Zaidi M., The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, *IPROMS 2006 Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems*, Oxford, Elsevier, 2006a.

Pham D.T., Otri S., Ghanbarzadeh A., Kog E., Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition, *ICTTA'06 Information and Communication Technologies*, 1624-1629, 2006b.

Pham D.T., Koc E., Ghanbarzadeh A., Otri S., Optimisation of the Weights of Multi-Layered Perceptions Using the Bees Algorithm, *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*, Sakarya, Turkey, 38-46,2006c.

Pham D.T., Soroka A.J., Ghanbarzadeh A., Kog E., Otri S., Packianather M., Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm, *IEEE International Conference on Industrial Informatics*, 8,1346-1351,2006d.

Qin L.D., Jiang Q.Y., Zou Z.Y., Cao Y.J., A Queen-Bee Evolution Based on Genetic Algorithm for Economic Power Dispatch, *UPEC 2004 39th International Universities Power Engineering Conference*, Bristol, UK, 453-456,2004.

Quijano N., Passino K.M., Honey Bee Social Foraging Algorithms for Resource Allocation Theory and Application, *American Control Conference*, New York City, USA, 2007.

Racer M., Amini M.M., A Robust Heuristic for the Generalized Assignment Problem, *Annals of Oper. Res.*, 50,487-503,1994.

Randall M., Heuristics for Ant Colony Optimisation Using the Generalised Assignment Problem, In *Proceedings of the IEEE Congress on Evolutionary Computation*, Portland, Oregon, 2,1916-1923,2004.

Ross G.T., Soland P.M., A Branch and Bound Based Algorithm for the Generalized Assignment Problem, *Mathematical Programming*, 8,91-103,1975.

Sato T., Hagiwara M., Bee System: Finding Solution by a Concentrated Search, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 4(C), 3954-3959,1997.

Savelsbergh M., A Branch-and-Price Algorithm for the Generalized Assignment Problem, *Operations Research*, 45,831-841,1997.

Schmickl T., Thenius R., Crailsheim K., Simulating Swarm Intelligence in Honey Bees: Foraging in Differently Fluctuating Environments, *GECCO'05*, Washington, DC, USA, 273-274,2005.

Seeley T.D., *The Wisdom of the Hive*, Harvard University Press, Cambridge, MA, 1995.

Seeley T.D., Buhrman S.C., Group Decision Making in Swarms of Honey Bees, *Behav. Ecol. Sociobiol.*, 45,19-31,1999.

Sung H.J., Queen-Bee Evolution for Genetic Algorithms, *Electronic Letters*, 39(6), 575-576, 2003.

Teo J., Abbass H.A., An Annealing Approach to the Mating-Flight Trajectories in the Marriage in Honey Bees Optimization Algorithm, *Technical Report CS04/01*, School of omputer Science, University of New South Wales at ADFA, 2001.

Teo J., Abbass H.A., A True Annealing Approach to the Marriage in Honey-Bees Optimization Algorithm, *International Journal of Computational Intelligence and Applications*, 3(2), 199-211,2003.

Teodorovic D., Dell'Orco M., Bee Colony Optimization - A Cooperative Learning Approach to Complex Transportation Problems, *Advanced OR and AI Methods in Transportation*, 51-60,2005.

Wedde H.F., Farooq M., Zhang Y., BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior, *Ant Colony, Optimization and Swarm Intelligence*, Eds. M. Dorigo, Lecture Notes in Computer Science 3172, Springer Berlin, 83-94,2004.

Yagiura M., Yamaguchi T., Ibaraki T., A Variable-Depth Search Algorithm with Branching Search for the Generalized Assignment Problem, *Optimization Methods and Software*, 10,419-441,1998.

Yagiura M., Yamaguchi T., Ibaraki T., A Variable-Depth Search Algorithm for the Generalized Assignment Problem, Vob S, Martello S., Osman I.H., Roucairol C, (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Boston, 459-471,1999.

Yagiura M., Ibaraki T., Glover P., An Effective Metaheuristic Algorithm for the Generalized Assignment Problem, In *2001 IEEE International Conference on Systems, Man and Cybernetics*, pg 242, Tucson, Arizona, 2001.

Yagiura M., Ibaraki T., Glover P., A Path Relinking Approach for the Generalized Assignment Problem, In *Proceedings of the International Symposium on Scheduling*, 105-108,2002.

Yagiura M., Ibaraki T., Glover P., An Ejection Chain Approach for the Generalized Assignment Problem, *Informs Journal of Computing*, 16(2), 131-151,2004.

Yagiura M., Ibaraki T., Glover E, A Path Relinking Approach with Ejection Chains for the Generalized Assignment Problem, *European Journal of Operational Research*, 169,548-569,2006.

Yang X.S., Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, *IWINAC 2005, LNCS 3562*, Yang, J. M. and J.R. Alvarez (Eds.), Springer-Verlag, Berlin Heidelberg, 317-323,2005.

Yonezawa Y., Kikuchi T., Ecological Algorithm for Optimal Ordering Used by Collective Honey Bee Behavior, *7th International Symposium on Micro Machine and Human Science*, 249-256,1996.
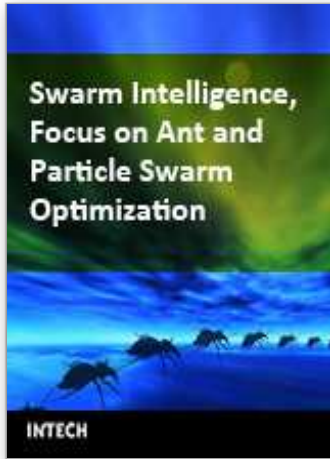
Swarm Intelligence, Focus on Ant and Particle Swarm Optimization

Edited by FelixT.S.Chan and Manoj KumarTiwari

In the era globalisation the emerging technologies are governing engineering industries to a multifaceted state. The escalating complexity has demanded researchers to find the possible ways of easing the solution of the problems. This has motivated the researchers to grasp ideas from the nature and implant it in the engineering sciences. This way of thinking led to emergence of many biologically inspired algorithms that have proven to be efficient in handling the computationally complex problems with competence such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. Motivated by the capability of the biologically inspired algorithms the present book on "Swarm Intelligence: Focus on Ant and Particle Swarm Optimization" aims to present recent developments and applications concerning optimization with swarm intelligence techniques. The papers selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. In addition to the introduction of new concepts of swarm intelligence, this book also presented some selected representative case studies covering power plant maintenance scheduling; geotechnical engineering; design and machining tolerances; layout problems; manufacturing process plan; job-shop scheduling; structural design; environmental dispatching problems; wireless communication; water distribution systems; multi-plant supply chain; fault diagnosis of airplane engines; and process scheduling. I believe these 27 chapters presented in this book adequately reflect these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Adil Baykasoğlu, Lale Özbakır and Pınar Tapkan (2007). Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem, Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, FelixT.S.Chan and Manoj KumarTiwari (Ed.), ISBN: 978-3-902613-09-7, InTech, Available from: http://www.intechopen.com/books/swarm_intelligence_focus_on_ant_and_particle_swarm_optimization/artificial_bee_colony_algorithm_and_its_application_to_generalized_assignment_problem

INTECH
open science | open minds